



INSCRIPCIONES

CENTRO DE EDUCACION CONTINUA DE LA
DIVISION DE ESTUDIOS SUPERIORES DE
LA FACULTAD DE INGENIERIA, U. N. A. M.

Palacio de Minería Calle de Tacuba No. 5
México 1, D. F.

Horario de oficinas:

Lunes a viernes de 9 a 18 h.

Cuota de inscripción \$ 3,500.00

La cuota de inscripción incluye:

- una carpeta con las notas de los profesores
- bibliografía sobre el tema
- servicio de cafetería
- comidas

Requisitos

- Pagar la cuota de inscripción o traer oficio de la empresa o institución que ampare su inscripción, a más tardar una semana antes del inicio del curso
- Llenar la solicitud de inscripción

Para mayores informes hablar a los teléfonos

521-40-20 521-73-35 512-31-23

CONSTANCIA DE ASISTENCIA

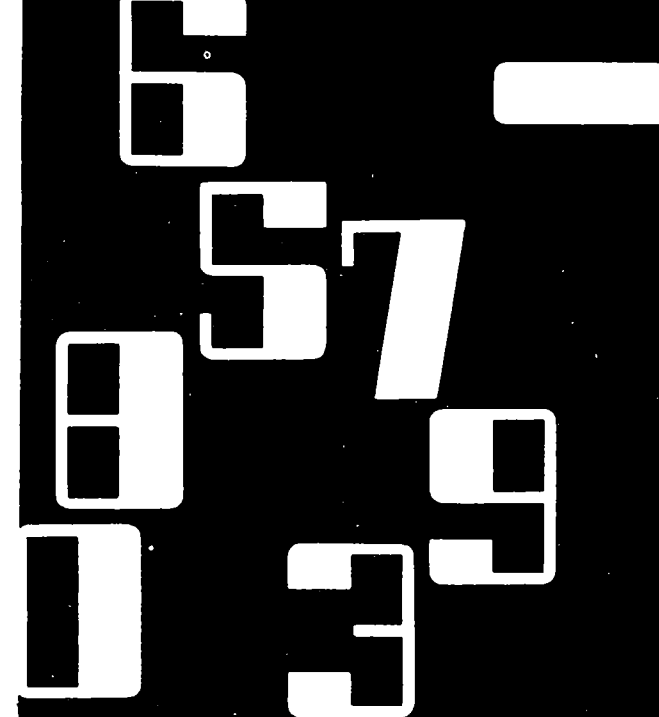
Las autoridades de la Facultad de Ingeniería de la U.N.A.M., otorgarán una constancia de asistencia a los participantes que concurren regularmente y que realicen los trabajos que se les asignen durante el curso.

CIRCULA LIBRE DE PORTE
POR VIA DE SUPERFICIE
Y DENTRO DEL TERRITORIO NAL.
ART. 17 LEY ORGANICA DE LA U N A M



centro de educación continua
división de estudios superiores
facultad de ingeniería, u n a m

Palacio de Minería
Calle de Tacuba No. 5
México 1, D.F.



78

métodos numéricos y aplicaciones con la computadora digital

Duración: 45 h

Fechas: del 7 al 29 de abril

Horario: viernes de 17 a 21 h; sábados de 9
a 13 y de 14 a 17 h

Coordinador: M. en C. Verónica Czitrom

En colaboración con la Asociación de Ingenieros
Universitarios Mecánicos Electricistas, A.C.

centro de educación continua
división de estudios superiores
facultad de ingeniería, u n a m



PRESENTACION DEL CURSO

La revolución característica del siglo XX ha sido el computador. Para resolver modelos de problemas de ingeniería, administración, economía y planeación, es necesario conocer las técnicas de solución numérica utilizando la más importante herramienta de cálculo, el computador digital.

OBJETIVO

En este curso se darán al profesionista las técnicas modernas para la solución numérica de complejos modelos de situaciones del ejercicio profesional.

A QUIEN SE DIRIGE

Este curso está dirigido a profesionales de las ramas de ingeniería, administración, planeación y economía que necesita emplear el computador digital para resolver problemas numéricos de su apreció profesional.

PRERREQUISITO

Es deseable tener un conocimiento básico de programación.

TEMARIO

1. **Repaso de Fortran**
 - 1.1 Introducción
 - 1.2 Lenguaje Fortran
2. **Algebra Matricial**
 - 2.1 Suma
 - 2.2 Multiplicación
 - 2.3 Operaciones elementales de matrices
 - 2.4 Inversión
 - 2.5 Eigenvalores y eigenvectores
3. **Sistemas de Ecuaciones Lineales**
 - 3.1 Método de Gauss-Jordan
 - 3.2 Método de Gauss-Seidel
4. **Raíces de Funciones Trascendentes y Polinomios**
 - 4.1 Funciones trascendentes
 - 4.2 Funciones polinomiales
5. **Interpolación**
 - 5.1 Polinomios de Lagrange
 - 5.2 Polinomios de Hermíticos
6. **Integración y Diferenciación Numérica**
 - 6.1 Método trapezoidal
 - 6.2 Método de Simpson
 - 6.3 Diferenciación numérica
7. **Solución de Ecuaciones Diferenciales**
 - 7.1 Método de Runge Kutta
 - 7.2 Método de Milne
 - 7.3 Método de Diferencias Finitas
 - 7.4 Sistemas de Ecuaciones Diferenciales
8. **Optimización**
 - 8.1 Programación Lineal
 - 8.2 Programación Dinámica

PROFESORES

M. EN C. VERONICA CZITROM
DR. VICTOR GEREZ GREISER
M. EN C. MARCIAL PORTILLA ROBERTSON



centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



A LOS ASISTENTES A LOS CURSOS DEL CENTRO DE EDUCACION
CONTINUA

Las autoridades de la Facultad de Ingeniería, por conducto del Jefe del Centro de Educación Continua, otorgan una constancia de asistencia a quienes cumplan con los requisitos establecidos para cada curso. Las personas que deseen que aparezca su título profesional precediendo a su nombre en la constancia, deberán entregar copia del mismo o de su cédula a más tardar el SEGUNDO DIA de clases, en las oficinas del Centro con la señorita encargada de inscripciones.

El control de asistencia se llevará a cabo a través de la persona encargada de entregar las notas del curso. Las inasistencias serán computadas por las autoridades del Centro, con el fin de entregarle constancia solamente a los alumnos que tengan un mínimo del 80% de asistencia.

Se recomienda a los asistentes participar activamente con sus ideas y experiencias, pues los cursos que ofrece el Centro están planeados para que los profesores expongan una tesis, pero sobre todo, para que coordinen las opiniones de todos los interesados constituyendo verdaderos seminarios.

Es muy importante que todos los asistentes llenen y entreguen su hoja de inscripción al inicio del curso. Las personas comisionadas por alguna institución deberán pasar a inscribirse en las oficinas del Centro en la misma forma que los demás asistentes, entregando el oficio respectivo.

Con objeto de mejorar los servicios que el Centro de Educación Continua ofrece, al final del curso se hará una evaluación a través de un cuestionario diseñado para emitir juicios anónimos por parte de los asistentes.

METODOS NUMERICOS Y APLICACIONES CON LA COMPUTADORA
(7 al 29 de abril, 1978)

Fecha	Horario	Tema	Profesor
7 de abril	17 a 21 h	REPASO DE FORTRAN	M. en C. Carlos Ramos Larios
8 de abril	9 a 13 h	REPASO DE FORTRAN	Ing. Ricardo Ciria M.
8 de abril	14 a 17 h	ALGEBRA MATRICIAL	Dr. Víctor Gerez Greiser
14 de abril	17 a 21 h	ALGEBRA MATRICIAL	M. en C. Verónica Czitrom
15 de abril	9 a 17 h	SISTEMAS DE ECUACIONES LINEALES	Dr. Víctor Gerez Greiser
21 de abril	17 a 21 h	RAICES DE FUNCIONES TRASCENDENTES Y POLINOMIOS INTERPOLACION	Ing. Horacio Sandoval
22 de abril	9 a 17 h	INTEGRACION Y DIFERENCIACION NUMERICA	M. en C. Verónica Czitrom
28 de abril	17 a 21 h	SOLUCION DE ECUACIONES DIFERENCIALES	Ing. Horacio Sandoval
29 de abril	9 a 13 h	OPTIMIZACION	M. en C. Verónica Czitrom
29 de abril	14 a 17 h	OPTIMIZACION	Dr. Víctor Gerez Greiser
		CLAUSURA.	

DIRECTORIO DE PROFESORES

METODOS NUMERICOS Y APLICACIONES CON LA COMPUTADORA DIGITAL

ING. RICARDO CIRIA MARCE
DEPARTAMENTO DE PROCESAMIENTO
CENTRO DE CALCULO
FACULTAD DE INGENIERIA, UNAM
CIUDAD UNIVERSITARIA
MEXICO 20, D.F.
TEL: 550.52.15 ext. 4150

M. EN C. VERONICA CZITROM
PROFESORA DE MATEMATICAS
D. E. S. F. I., UNAM
CIUDAD UNIVERSITARIA
MEXICO 20, D.F.
TEL: 548.09.50
595.53.58(dom.)

DR. VICTOR GEREZ GREISER
PROFESOR TITULAR
INGENIERIA MECANICA Y ELECTRICA
FACULTAD DE INGENIERIA, UNAM
CIUDAD UNIVERSITARIA
MEXICO 20, D.F.
TEL: 550.52.15 ext.3750 ó 3746
525.64.52

ING. CARLOS RAMOS LARIOS
CENTRO DE CALCULO
FACULTAD DE INGENIERIA, UNAM
CIUDAD UNIVERSITARIA
MEXICO 20, D.F.
TEL: 550.52.15 ext. 4150

ING. JOSE HORACIO SANDOVAL RODRIGUEZ
INVESTIGADOR
INSTITUTO DE INGENIERIA, UNAM
CIUDAD UNIVERSITARIA
MEXICO 20, D.F.
TEL: 548.65.60 ext. 208



centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



METODOS NUMERICOS Y APLICACIONES CON LA COMPUTADORA
DIGITAL.

TEMA I : REPASO DE FORTRAN.

ABRIL, 1978.

FORTRAN

INTRODUCCION.- Desde el inicio de las computadoras digitales, uno de los campos de más aplicación de estos dispositivos ha sido el científico, donde existen necesidades de cálculos y operaciones muy complicadas o largas, siendo la computadora un auxiliar poderoso en la solución de estos problemas.

Inicialmente, la programación de las computadoras se hacía a nivel de lenguaje de máquina, esto es, instrucciones numéricas que obligan a la máquina a ejecutar una operación sencilla (sumar, restar, etc.). Esta manera de programar a la computadora exigía al usuario un profundo conocimiento de las características del equipo usado, tanto de hardware como de operación.

Por la dificultad que entrañaba el usar una computadora para un uso científico, se pensó en simplificar la programación mediante un lenguaje más parecido al lenguaje matemático, por lo tanto más sencillo de aprender, y que pudiera ser compatible con distintas marcas de computadoras. Uno de los primeros lenguajes de alto nivel (o sea que para ser ejecutados tienen que ser traducidos mediante un compilador) fue el Fortran (Fórmula Translation) que es un lenguaje para uso eminentemente científico, con instrucciones muy fáciles de entender y con compiladores en casi todas las computadoras del mercado.

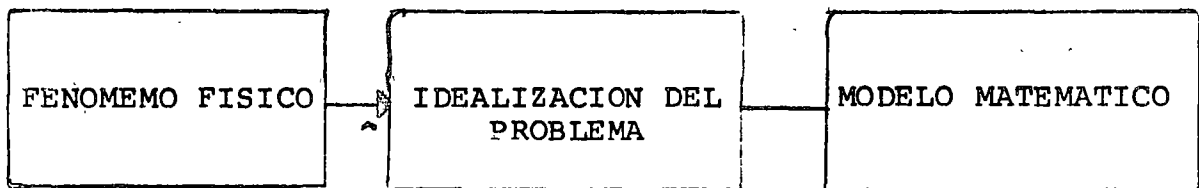
Estas notas estarán orientadas a él "cómo resolver el problema utilizando la computadora". Pues durante el resto del curso la computadora será una herramienta para el dise

ño de mecanismos.

La estructura del problema tiene 4 pasos a seguir, los cuales son:

- 1.- La formulación precisa del problema
- 2.- Modelo matemático
- 3.- Análisis matemático
- 4.- Solución del problema con computadora

Resulta importante poder pasar del fenómeno físico al modelo matemático, esta relación se muestra en la siguiente figura:

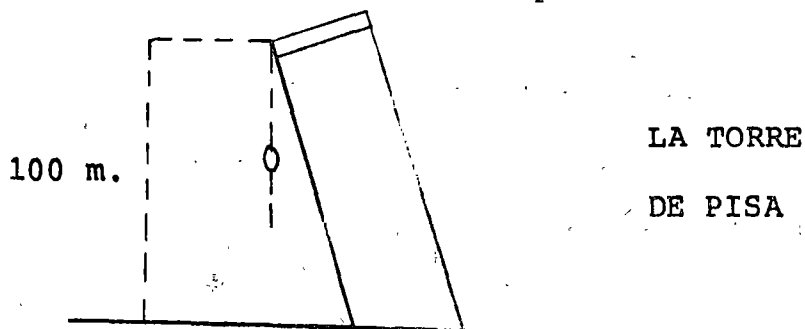


Veamos como sucede esto en la solución de un problema científico. Por ejemplo veamos el caso de la Ley de Movimiento de Newton, la cual dice que la fuerza es igual a la masa por la aceleración:

$$f = ma$$

Supongamos que esta relación es exacta, sin tomar en cuenta la teoría de la relatividad.

Este típico problema se estudió en un curso elemental de Física conocido como el problema de la piedra que cae!



Queremos preguntarnos cuánto tarda la piedra en caer.

Si analizamos la situación vemos que la distancia d que viaja en un tiempo t con aceleración constante es:

$$d = \frac{a t^2}{2}$$

$$\text{si } a = 9.8 \text{ m/seg.}^2$$

$$t = \frac{9.8 \times (100)^{1/2}}{2}$$

Bien ahora preguntémosnos qué tan realista fue nuestra respuesta. Notemos que NO consideramos efectos tales como:

- 1.- La variación de la dirección de la gravedad
- 2.- Variación de la gravedad dependiente de la altura sobre el nivel del mar.
- 3.- Resistencia del aire (sobre la piedra)
 - a) Forma de la piedra
 - b) Velocidad de la piedra
 - c) Densidad del aire (varía con la altitud y la temperatura).
 - d) Densidad de la piedra.
- 4.- Atracción gravitacional entre sol y luna
- 5.- Vientos y corrientes de aire, etc.

Es posible incluir todos estos factores en nuestro modelo matemático, analizar estas ecuaciones y mejorar

el tiempo de caída real (pregunta original). Creo yo que hemos llevado el problema a un extremo, como conclusión podemos decir que el modelo debe ser lo suficientemente exacto para obtener de este resultados útiles, -- sin caer en el extremo anterior, donde el precio que -- hay que pagar en el análisis matemático y esfuerzos de computación, tal vez no sea lo que queremos.

Una vez que se ha hecho el diseño matemático - del modelo del problema, es necesario determinar un "algoritmo" para resolverlo, ésto es, una serie finita de pasos que nos lleve a la solución del problema.

Para el problema anterior, el algoritmo sería - el siguiente:

- 1.- Leer el valor de la altura (d)
- 2.- Hacer $a = 9.8 \text{ m/seg.}^2$
- 3.- Hacer $t = \frac{9.8 \times d}{1/2}$

2

- 4.- Escribir el valor de t como respuesta.

Una de las técnicas más populares para describir algoritmos es por medio de diagramas de flujo, los cuales se explicarán a continuación.

DIAGRAMAS DE FLUJO

Los diagramas de flujo son representaciones gráficas

ficas de los programas. Cada decisión y operación a desarrollar será colocada en una caja, la forma de la caja nos indicará el tipo de instrucción a desarrollar. Se utilizarán flechas para interconectar estas cajas, las flechas nos indicarán la secuencia de las operaciones, usualmente debemos empezar por la parte superior y bajar siguiendo las flechas (top down).

El análisis del problema se facilita utilizando diagramas de flujo, pues no son ambiguos y tienen una estructura similar a la del problema. Desafortunadamente, no existe hoy en día una convención estándar para estas representaciones, a lo largo de estas notas se usará la notación IBM, que resulta ser la más general aunque no es universal.

Las siguientes reglas serán usadas.

- 1.- Cada proposición será colocada en una caja.
(Es válido colocar varias proposiciones en la misma caja).
- 2.- La secuencia de las operaciones se indica con segmentos de línea dirigidos (flechas) entre las cajas.
- 3.- Se utilizan diferentes tipos de cajas según sea la proposición.

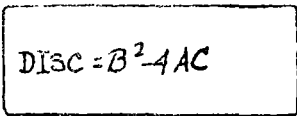
DESCRIPCION DE LOS SIMBOLOS



Indica inicio del programa o de un subprograma.

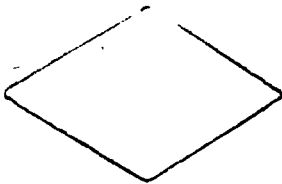


Operación a ejecutar por ejemplo de la figura 1.



Indica que se debe de hacer la operación:

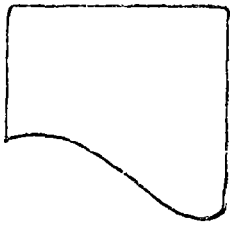
$B^2 - 4 AC$ y su valor asignar lo a DISC.



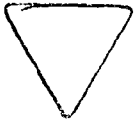
Proposición IF (# pag 7) compa ración, compara lo que está dentro de la caja dependiendo de esta comparación se podrán seguir 3 caminos. La comparación se indica con: si queremos comparar I con N, lo indicaremos (dentro de la caja) por I : N.



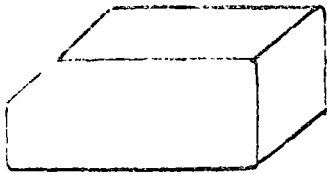
Indica que hay que leer tarjetas perforadas con datos o proposiciones.



Indica que queremos imprimir en la impresora algún mensaje o resultado..



Este símbolo es un conector



Paquete de tarjetas (usualmente un programa).



Operación DO (la cual se explicara en la siguiente sección.



Fin o alto del programa.



Cinta magnética o cinta de papel perforado.

Primer problema de clase.

Este problema trata de determinar la precipitación pluvial total y su promedio en el Distrito Federal durante un lapso, digamos un año.

Los datos con los que se cuenta son las lecturas diarias efectuadas en milímetros, (notemos que no es posible tener números negativos) o sea la cantidad de lluvia.

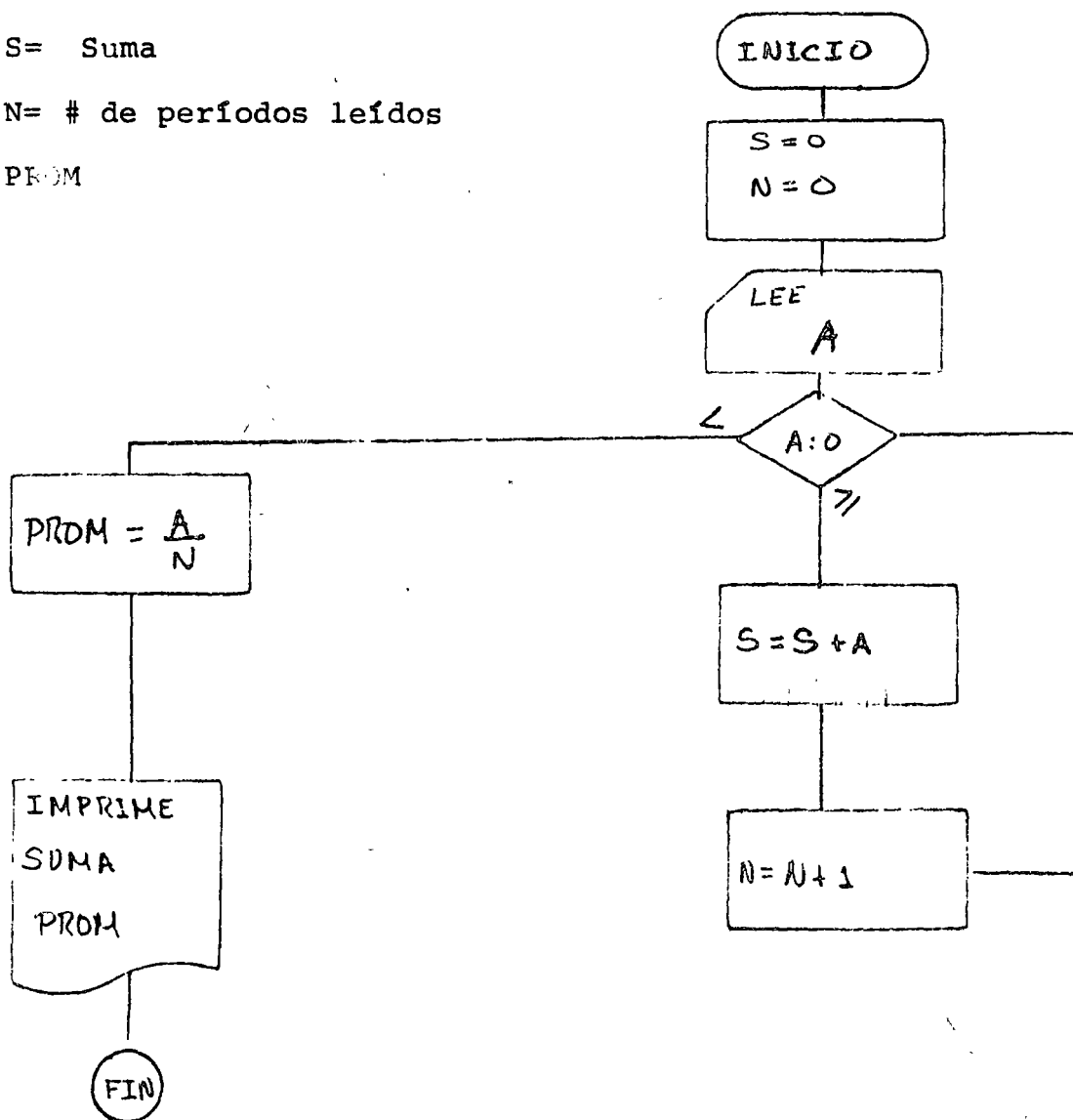
Procederemos a resolver el problema de la siguiente forma, (utilizando primero un diagrama de flujo).

A= cantidad de lluvia (agua)

S= Suma

N= # de periodos leídos

PROM



A continuación procederemos a explicar el diagrama de flujo.

La primera caja de inicio nos indica el comienzo de nuestro programa. La siguiente caja asigna el valor cero a las variables A y N donde A es el valor (cantidad de lluvia) leído, y N va a ser el número de lecturas tomadas. A continuación procedemos a leer el primer dato, notemos que no es posible tener, valor negativo de lluvia, por lo que valiéndonos de esta propiedad en la siguiente caja preguntamos si el valor leído es negativo, si es negativo calculamos el promedio de lluvia en el Distrito Federal, e imprimimos la suma y el promedio, para poder hacer esta operación en la última tarjeta de datos se pondrá un número negativo. Si el valor leído no es cero, se procede a sumar, lo cual se efectúa haciendo la asignación de la suma del valor anterior de suma más el valor leído, cosa similar ocurre en el "contador" del número de datos leídos. Notemos que este ciclo se repite hasta que leamos un número negativo, al suceder esto, el programa imprime el resultado y termina.

10

CODIFICACION FORTRAN DEL PROGRAMA

XXX TARJETAS DE CONTROL

C INICIO DE LAS VARIABLES

SUMA = 0

N = 0

1 READ (2, 2) A

2 FORMAT (15)

IF (A) 3, 3, 4

4 SUMA = SUMA + A

N = N + 1

GO TO 1

3 PROM = SUMA/N

WRITE (3,5) PROM, SUMA

5 FORMAT (10X, F10.4, 5X, I5)

CALL EXIT

END.

DESCRIPCION.

Como todo lenguaje (ya sea de programación o natural) el Fortran tiene un alfabeto, ésto es, una serie de símbolos que sirven para formar expresiones e instrucciones. El alfabeto de Fortran para B - 6500 consta de:

Letras: A, B, C, D, . . . , X, Y, Z.

Dígitos: 0, 1, 2, 3, . . . , 9 .

CARACTERES ESPECIALES.

+	MAS
-	MENOS
*	ASTERISCO (MULTIPLICA)
/	DIAGONAL (SLASH) DIVIDE
=	ASIGNA (NO CONFUNDIR CON IGUALDAD)
,	COMA (USADA COMO SEPARADOR)
(ABRE PARENTESIS
)	CIERRA PARENTESIS
	ESPACIO EN BLANCO
"	COMILLAS (UTILIZADA EN FORMATOS)
**	DOS ASTERISCOS (ELEVA AL CUADRADO)

Todo programa en Fortran contiene instrucciones de los siguientes tipos:

- a) Asignación
- b) Control
- c) Entrada/Salida
- d) Información para el compilador
- e) Funciones y subprogramas

La forma de codificar (escribir) un programa en Fortran es la siguiente.

Cada tarjeta contiene 80 columnas que deben distribuirse de la siguiente manera:

COLUMNAS	USO
1 - 5	Número de proposición (etiqueta)
6	Continuación
7 - 72	Proposición
73 - 80	Identificación o número de secuencia

Un programa completo en Fortran se vería codificado como sigue: (PROGRAMA MOSTRADO EN DIAGRAMA DE FLUJO ANTERIORMENTE).

5	6	7	72	73	80
10		READ 10, A, B, C			
		FORMAT (3 F10.0)			
		IF (A(20, 50, 20			
20		DISC = B **2 - 4*A*C			
		IF (DISC) 40, 25, 25			
25		DISC = SORT (DISC)			
		X1 = (-B + DISC) / (2*A)			
		X2 = (-B - DISC) / (2*A)			
		PRINT 30, X1, X2			
30		FORMAT (1H, 2 F10. 3)			
		GO TO 50			
40		PRINT 45			
45		FORMAT ("DISCRIMINANTE NEGATIVO")			
50		CALL EXIT			
		END			

CONSTANTES.

Una constante en Fortran puede ser de 2 tipos:

- a) Entera (Integer)
- b) Real (Real)

a) Una constante entera es cualquier número sin punto decimal. Ejemplo:

0, 91, - 173, + 327

si un entero se escribe sin signo, se supone positivo. Los valores que pueden tomar las constantes en IBM-1130 son los comprendidos en el rango: ^

- 32767	a	32767
(- (2 ¹⁵ - 1))	a	2 ¹⁵ - 1)

No se permite introducir comas en una constante entera. Ejemplo de constantes enteras ilegales:

3.2	tiene punto decimal
27.	
31459036	demasiado grande
5,496	contiene una coma

b) Una constante real es cualquier número con punto decimal. Ejemplo:

0.
 91.3
 -145.8
 5.E3

Que escribiremos como:

5.0 x 10 ³	5. E03
-5. x 10 ³	-5. E03
4.1 x 10 ⁰	4. E00

La magnitud de una constante real no debe ser mayor que 2^{127} ó menor que 2^{-128}

VARIABLES. Una variable en Fortran es la representación simbólica de una cantidad que puede tomar diferentes valores.

Por ejemplo, en la instrucción

$$A = 5.0 + B$$

A y B son variables, el valor de B está determinado por alguna instrucción previa y puede cambiar. El valor de A está variando para cada nuevo valor de B.

NOMBRES DE VARIABLES.- Un nombre de una variable consiste en una cadena de 1 a 5 caracteres alfanuméricos, excluyendo caracteres especiales, y siendo el primero una letra.

Ejemplo: de nombres de variables permisibles.

DET

AB1

I

LL4E

Ejemplo: de nombres de variables ilegales:

LLL4 Empieza con caracter no alfabetico

ABCDEFGHIJ Demasiado grande

A-B Caracter ilegal

A/B Caracter ilegal

El tipo de cantidad (real o entera) que representa se puede especificar de dos maneras: explícita e implícitamente.

La forma implícita de especificar una variable es como sigue:

- a) si la primera letra del nombre de la variable es: I, J, K, L, M, N, la variable se considera como una variable entera.
- b) Si la primera letra del nombre de la variable NO es: I, J, K, L, M, N, la variable se considera como una variable real.

La forma explícita de especificar una variable es usando una Declaración de tipo, la cual hace que el compilador ignore la especificación implícita, por ejemplo: si por medio de una declaración de tipo, designamos a la variable ITEM como de tipo real, será manejada por el compilador como una variable real, sin importar que su primera letra sea una I.

EXPRESIONES ARITMETICAS

Una expresión aritmética (e.a.) es una sucesión de constantes, variables y símbolos de operaciones aritméticas que siguen las reglas que a continuación se darán.

Los siguientes son los símbolos de operación aritmética:

<u>Símbolo u operador:</u>	<u>Significado:</u>
+	SUMA
-	RESTA
*	MULTIPLICACION
/	DIVISION
**	EXPONENCIACION

Ejemplo:

<u>Expresión algebraica</u>	<u>Equivalente en Fortran</u>
$a + b$	A + B
$a - b$	A _ B
ab	A * B
$\frac{a}{b}$	A/B
a^b	A**B

REGLAS.

- 1.- Dos operadores no deben estar juntos. Deben estar separados por cantidades o paréntesis en la expresión por ejemplo; $A + B$ es inválida, mientras que $B + A$ ó $A + (B)$ son válidas.
- 2.- No se pueden omitir operadores, por ejemplo:

3A NO significa 3*A.

MODOS COMPUTACIONALES.

Los cálculos aritméticos son hechos en dos formas: entera y real, dependiendo del tipo de las cantidades envueltas en el cálculo. Las constantes o variables que forman una expresión aritmética no necesitan ser del mismo tipo.

El modo de la expresión es entero, real o mixto, dependiendo de si las constantes son enteras, reales o están -- mezcladas.

Por ejemplo:

<u>EXPRESION</u>	<u>TIPO DE CANTIDAD</u>	<u>MODO DE LA EXPRESION</u>
3	Constante entera	entera
I + J	Variables enteras	entero
3.0	Constante real	real
A	Variable real	real
5*JOB + ITEM	Variables enteras, Constante entera	entera
A**B	Variables reales	real
A + B/ITEM	Variables reales	mixto (el resultado se guarda como real)

Se pueden usar paréntesis en las expresiones aritméticas como en algebra, para especificar el orden en el cual se van a efectuar las operaciones aritméticas que forman la expresión.

PROPOSICION DE ASIGNACION

La forma general de esta proposición es:

(variable) = (expresión aritmética)

Ejemplo:

A = 5

AB2 = A * (B ** 2)

DISCR = B ** 2 - 4 * A * C

El objeto de esta instrucción es el de asignar a la (variable) el valor de la (expresión aritmética), borrando el valor anterior de dicha variable.

PROPOSICIONES DE ENTRADA/SALIDA

Las proposiciones de entrada/salida permiten al programador introducir (obtener) datos al (del) programa.

La forma general de dichas instrucciones es:

READ (2 , n_f) (lista de variables)

WRITE (3 , n_f) (lista de variables

n₁

Es el número de la unidad (física) de lectura de datos: Lectora de tarjetas perforadas, lectora de cinta perforada, cinta magnética, etc. En IBM-1130 este número es el 2 para lectora de tarjetas.

n_f

Es el número de una proposición de formato. Para que el programa pueda leer datos, debe tener conocimiento de la forma en que se le van a presentar. Esto se hace mediante la proposición FORMAT.

LA FORMA GENERAL DE ESTA POSICION ES

n_f FORMAT (lista de especificaciones)

La lista de Especificaciones le indica al compilador en qué forma están perforados los datos. Básicamente hay dos tipos de especificaciones

- I entera
- F Flotante (cant. reales)
- E Exponencial

El formato I tiene la forma
Iw donde w es el ancho del campo

Este formato se utiliza para leer (o escribir) valores de variables enteros, con un ancho máximo de w dígitos.

Por ejemplo, si queremos leer un valor de la variable L de tarjeta, tendremos que escribir:

```

      READ   ( 2, 10 ) L
10  FORMAT (I4)

```

Las dos proposiciones anteriores hacen que la computadora lea un valor entero de a lo más 4 dígitos, y lo asigne a la variable L.

Para leer (o escribir) valores que corresponden a -- cantidades reales, se usa el formato F el cual tiene la forma general Fw.d, donde w es el número máximo de dígitos y d es el número de dígitos decimales. (d puede ser 0).

Por ejemplo, para leer el valor de la variable R podemos usar:

READ (2,3) R

3 FORMAT (F10.3)

Esto le indica al compilador que va a leer un valor real de 9 dígitos, de los cuales 3 son decimales.

Para la impresión de resultados, el número asignado a la impresora en línea en IBM-1130 es 3.

Existe un formato que permite mejorar la impresión de resultados, este es el formato X, el cual hace que la impresora (y en algunos casos la lectora) se "salte" tantos espacios como lo indique la especificación, por ejemplo, la especificación 4x hará que el programa, en impresión, deje 4 espacios en blanco, libres.

A continuación veremos las conversiones en entrada-salida, de distintos valores bajo diferentes especificaciones.

(Ø significa espacio en blanco)

ENTRADA

<u>Campo de Entrada</u>	<u>Especificación</u>	<u>Valor Interno</u>
567	I3	+ 567
- 329	I6	- 329
- 27	I7	- 27
27	I5	+ 27000
234	I7	234

SALIDA

<u>Valor Interno</u>	<u>Especificación</u>	<u>Campo de Salida</u>
+ 23	I4	+ 23

<u>Valor Interno</u>	<u>Especificación</u>	<u>Campo de Salida</u>
- 79	I 4	- 79
+ 30145	I 5	30145
- 30145	I 5	*****
+ 978	I 1	*
0	I 3	0

ENTRADA

<u>Campo de Entrada</u>	<u>Especificación</u>	<u>Valor Interno</u>
36725931	F8.4	+ 3672.5931
3.672593	F8.4	+ 3.672593
- 367259	F8.4	- 367259
367259	F6.6	+ 0.367259

SALIDA

<u>Valor Interno</u>	<u>Especificación</u>	<u>Campo de Salida</u>
+ 36.7929	F7.3	36.763
+ 36.7934	F9.3	36.793
- 0.0316	F6.3	- 0.032
+ 579.645	F4.2	***
+ 579.645	F6.2	579.65
-579.645	F6.2	*****

En algunos de los casos, es necesario imprimir títulos o encabezados de tal manera de que los resultados del programa sean más legibles y comprensibles. Para este tipo de problemas es común poner el texto a ser impreso entre comillas, dentro del formato que le corresponda.

Por ejemplo, si queremos imprimir los valores de tres variables A, B, C, en una forma legible, podríamos usar el siguiente formato:

```
WRITE ( 3, 10 ) A, B, C,
10 FORMAT ('A = ', F10.4, ' B = ', F10.4, ' C = ' F10.4)
```

Lo cual provoca que la máquina imprima lo siguiente:
(suponiendo que A = 5.6, B = - 4.85, C = 1.492)
A = _ _ _ _ 5.6000 _ B = _ _ _ - 4.8500 C = _ _ _ _ 1. 4920

Es importante hacer notar que la primera posición de impresión (i-e la columna 1) sirve para el control del carro de impresión, por lo que es necesario tener presente este hecho siempre que se vaya a imprimir.

El control de carro posible se reduce a las siguientes:

PRIMERA COLUMNA

- 0 Espaciamiento sencillo antes de imprimir
- 0 Espaciamiento doble antes de imprimir
- 1 Salto a otra página antes de imprimir
- + Sobre - escritura antes de imprimir

Proposiciones de Control

Normalmente, las instrucciones de un programa en Fortran

se van ejecutando en forma secuencial, por lo que es necesario alterar (en algunos problemas) esta forma de ejecución.

Las instrucciones que alteran el flujo de las instrucciones en un programa son las instrucciones de control que son:

```
GO TO
IF Aritmético
IF Lógico

STOP

CONTINUE
```

Proposición GO TO

Esta proposición es llamada de transferencia incondicional y su forma general es:

```
GO TO n donde n es un número
de proposición
```

El efecto de esta proposición es el de transferir el flujo de las instrucciones a aquella que tiene el número de proposición n. Esto se puede visualizar como un "salto" en el orden de ejecución del programa.

Por ejemplo:

En el segmento de programa siguiente, el orden de ejecución es: 1,2,3,5,6,2,3,5,6.7

1	A = 0	4	C = A * B
2	B = A + 3	5	WRITE (3,8), A,B,C
3	GO TO 5	6	GO TO 2
		7	END

Proposición IF

Generalmente, en el transcurso de un programa, es necesario hacer "preguntas" sobre algunos valores de las variables del programa, y tomar "decisiones" según el resultado de la pregunta.

Esta "toma de decisiones se efectúa por medio de la proposición IF, la cual tiene las versiones formadas Aritmética Lógica.

El IF Aritmético tiene la forma

IF (exp. aritmética) n₁, n₂, n₃

Donde n₁, n₂ y n₃ son 3 etiquetas ó números de proposición.

El funcionamiento de este IF es el siguiente:

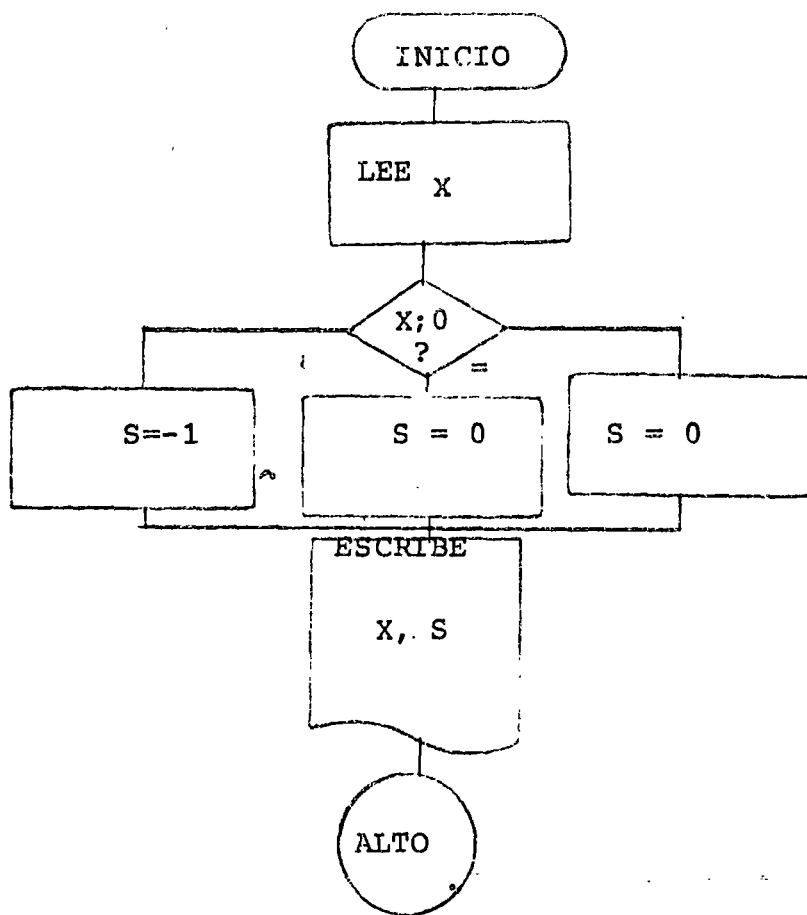
- 1.- Evalúa la expresión aritmética
- 2.- Compara el resultado con cero y toma cualquiera de las siguientes acciones:
 - a) Si el resultado es > 0, transfiere el control a la proposición con el número n₁
 - b) Si el resultado es = 0, transfiere el control a la proposición con el número n₂
 - c) Si el resultado es < 0, transfiere el control a la proposición con el número n₃

Como ejemplo, veamos un programa que calcula la función

SIGNUM (x), (x) definida como:

$$f(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases}$$

Hagamos primero un diagrama de Flujo:



La codificación en Fortran IB-1130 queda como sigue:

```

1  READ ( 2,2 ) X
2  FORMAT ( F10.0 )
3  IF (X) 4,6,8
4  S = - 1
5  GO TO 9
6  S = 0

```

26

```

7   GO TO 9
8   S = 1
9   WRITE (3,10) X,S
10  FORMAT (F10.6, 3)
    END

```

El IF lógico tiene la forma

IF (expresión lógica) proposición ejecutable

Una expresión lógica es una expresión que puede ser cierta o falsa, como es el caso de las comparaciones:

¿ Es A B ?

¿ Es $\text{sen } x + y \log. z$?

Este tipo de expresiones son llamadas expresiones de relación y consisten de dos expresiones aritméticas (reales o enteras) separadas por un operador de relación. Los operadores de relación son las 6 comparaciones matemáticas: $= \neq < > \leq \geq$ que en Fortran quedan expresadas como se muestra en la tabla:

<u>OPERADOR DE RELACION</u>	<u>NOMBRE EN FORTRAN</u>
=	.EQ.
≠	.NE.
<	.LT.
≤	.LE.
>	.GT.
≥	.GE.

Además de estos operadores, se usan tres operadores lógicos para construir expresiones más elaboradas. Estos operadores son llamados disyunción, conjunción y negación; sus nombres en Fortran son .OR., .AND., NOT. Respectivamente. Su funcionamiento es el siguiente:

Si X y Y son expresiones lógicas, entonces X.OR. Y es cierta, a menos que X o Y sean ambas falsas.

X. AND. Y es falsa, a menos que X y Y sean ambas ciertas.

NOT.X es falsa si X es cierta y viceversa

Veamos el siguiente ejemplo comparativo del uso del IF Lógico y del IF aritmético.

Supongamos que queremos hacer $R = R1$ solamente si $S = 3$, y $T = 4$, hay 3 maneras de hacer esto:

<u>IF ARITMETICO</u>	<u>IF LOGICO</u>
IF (S - 3) 1,2,1 a)	b)
2 IF (T-4) 1,3,1	IF(S.NE.3) GO TO 1 IF(S.EQ.3.AND.
	IF (T.EQ.4) R = R1 T. EQ.4) R=R1
1 - - - -	

Vemos que usando el IF aritmético necesitamos 3 etiquetas. En el IF Lógico, el funcionamiento es el siguiente:

a) Se evalúa (n) la (s) expresión (es) aritmética (s) involucrada (s) en la expresión lógica.

b) Se va evaluando el valor lógico (cierto o falso) de la expresión lógica, siguiendo un orden de - prioridad de operadores lógicos. El orden es el siguiente: La más alta prioridad es .NOT. Después se evalúa .AND. y finalmente .OR.

c) Una vez que se ha evaluado la expresión lógica entre los paréntesis del IF, se observa su valor y ocurre una de dos situaciones:

Si la expresión lógica es cierta, se procede a - ejecutar la proposición a la derecha del If;

Si la expresión lógica es falsa, se ignora dicha proposición y se continúa con la secuencia normal del programa.

ARREGLOS.- Generalmente, en los problemas científicos, es necesario usar vectores, matrices o estructuras con más di mensiones.

Debido a que todas las variables usadas en el programa ocupan un lugar en la memoria de la computadora, los problemas deben ser "declarados" (esto es, se le debe dar informa ción al compilador) a fin de que se les asignen localidades en la Memoria. La manera de declarar un arreglo en Fortran es por medio de la proposición "dimension" y es como sigue:

DIMENSION A (n₁, n₂), B (n₃), C(n₄, n₅, n₆)

Donde A, B, C son los nombres de los arreglos y n₁, n₂, ...n₆ son las dimensiones máximas de dichos arreglos.

Por Ejemplo:

Dimension NOM (10,10)

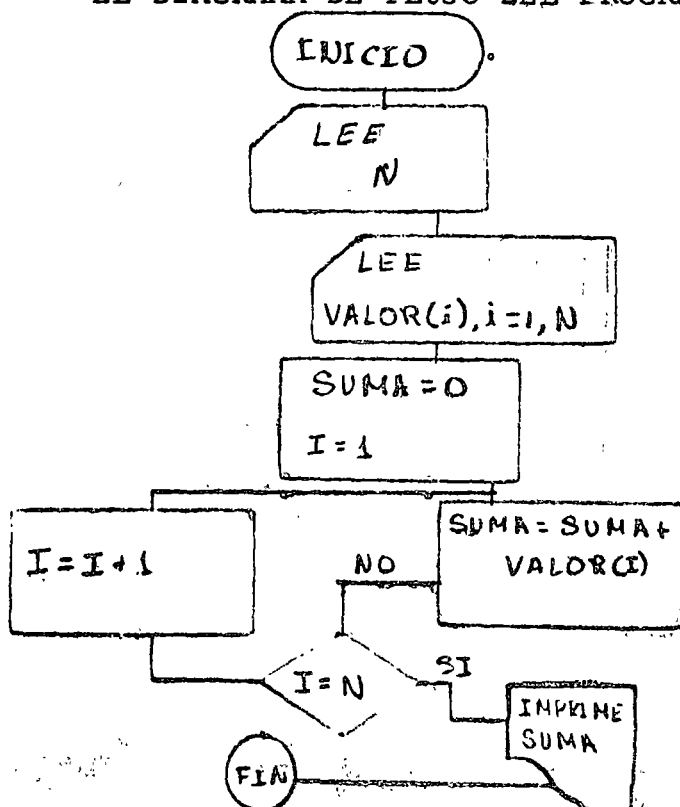
Hará que el compilador reserve 100 localidades para una matriz llamada NOM con 10 renglones y 10 columnas.

Para referirse a algún elemento de un arreglo, basta con poner el nombre del arreglo, seguido por el índice (o los índices) encerrado entre paréntesis por ejemplo, - si queremos sumar los dos primeros elementos del arreglo A y guardar el resultado en el tercero, lo haríamos por medio de la siguiente proposición.

$$A(3) = A(1) + A(2)$$

En general, los índices de un arreglo son variables enteras, que van cambiando de valores en el transcurso del programa. Como ejemplo, veamos un programa que lee un vector llamado valor, de a lo más 100 elementos; y calcula la suma de sus elementos.

EL DIAGRAMA DE FLUJO DEL PROGRAMA ES:



La codificación del programa en Fortran quedaría como sigue:

567

C		SE DECLARA EL ARREGLO DIMENSION VALOR (100)
C	1	SE LEE EL NUMERO REAL DE ELEMENTOS A PROCESAR READ (2,1) N FORMAT (13)
C	2	LA SIGUIENTE PROPOSICION LEE LOS ELEMENTOS DEL ARREGLO MEDIANTE UNA PROPOSICION DE ITERACION QUE SE EXPLICARA CON MAS DETALLE READ (2,2) (VALOR (I), I - 1,N) FORMAT (8F 10.0)
C	3	DE AQUI EN ADELANTE SON LOS CALCULOS SUMA = 0.0 I = 1 SUMA = SUMA + VALOR (I) IF (I.EQ.N) GO TO 4 I = I + 1 GO TO 3
	4	WRITE (3,5) SUMA
	5	FORMAT (1H, 'SUMA = ' , F12.4) CALL EXIT END

En el programa usamos una proposición de la forma

(VALOR (I), I = 1, N)

La cual es usada con mucha frecuencia para lectura e impresión de arreglos (de cualquier dimensión) y su funcionamiento es equivalente a escribir.

(VALOR (1), VALOR (2), VALOR (3), . . . , VALOR (N)

Este tipo de proposiciones de iteración son las que quizá dan más poder al lenguaje, ya que es posible realizar grandes cantidades de cálculos mediante pocas proposiciones.

Una de las proposiciones de interacción más usadas por los programadores en Fortran en la proposición DO.

PROPOSICION DO.

La proposición de control DO nos permite efectuar una serie de iteraciones con una sola proposición, por ejemplo: si queremos inicializar un arreglo A de N elementos a ceros, se puede hacer usando If's o usando una proposición Do. Veamos las dos formas:

Con If's

I = 1

1 A(I) = 0

I = I + 1

Con Do

Do 1 I = 1, N

1 A (I) = 0

La forma general de la proposición Do es:

Do etiqueta variable entera = valor inicial, valor final, incremento

La etiqueta que aparece en la proposición DO le indica a la máquina hasta donde llegar el alcance del DO, esto es, define los límites dentro de los cuales se efectuará la iteración.

La variable que servirá como un "contador" para el DO su valor inicial está dado en la proposición y se irá incrementando cada vez que llegue al final del DO, comparando su valor con el valor final especificado. En el caso de -- que sea mayor o igual, el ciclo termina y se ejecuta la proposición siguiente del bloque definido por el DO.

La última proposición en el bloque de un DO no puede ser Go To, If Return o Do. En el caso en que sea necesario usar algunas veces estas proposiciones, se recurre a una proposición "muda" que es el Continue cuya única función es la de definir a una etiqueta.

Se puede dar el caso de tener varios bloques de DO's -- "anidados", esto es, uno dentro del otro, siempre y cuando cada uno está completamente abarcado por el más largo.

Para ejemplificar lo que se ha dicho, veamos un segmento de programa que multiplica dos matrices A, B cada una de NxN y guarda el producto en la matriz C de N x N.

Recordemos que si $C = A * B$

$$C_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

El programa quedaría como:

```
.  
. .  
DO 1 I = 1, N  
DO 1 J = 1, N  
C (I,J) = 0  
DO 2 K = 1, N  
2 C (I,J) = C(I,J) + A(I,K) * B(K,J)  
1 CONTINUE
```

SUBROUTINAS Y FUNCIONES

Muy frecuentemente sucede que una Sección de programa, o secuencia de instrucciones es frecuentemente usada. Si tal caso sucede, tal sección del programa es usualmente -- identificada como una rutina separada llamada SUBROUTINE en Fortran (PROCEDURE ALGOL).

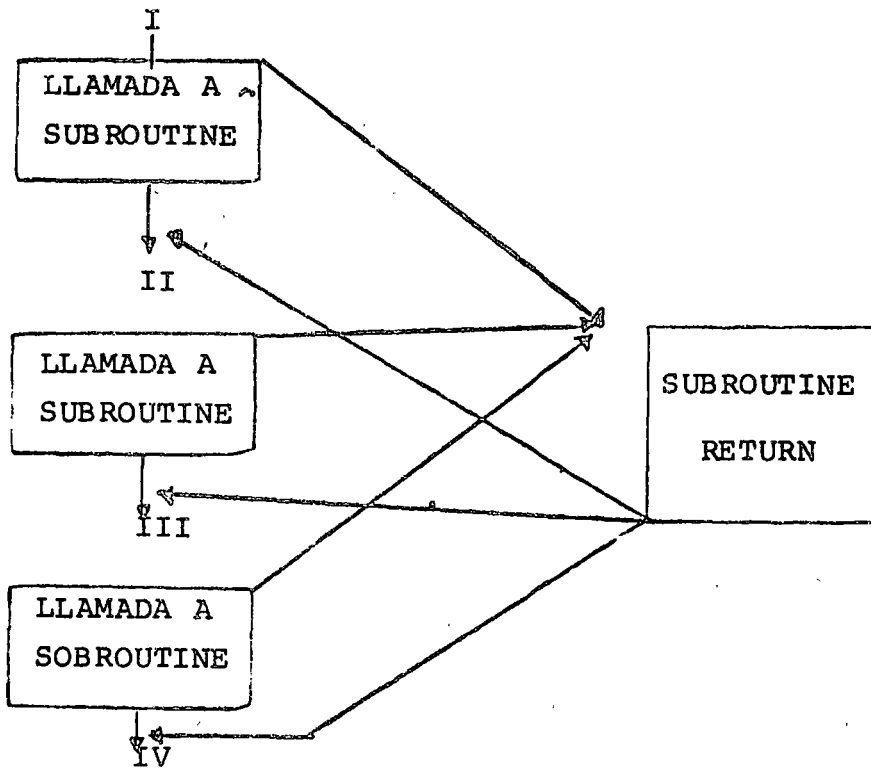
Cuando una subrutina es definida, se le dá un nombre de identificación, y sus argumentos son identificados, estos argumentos son sus variables. A continuación estudiaremos la naturaleza, uso y objeto de las subrutinas.

Cada subrutina debe de empezar con la proposición subrutina, su nombre, y una lista de argumentos y terminar con las PROPOSICIONES RETURN Y END.

```
SUBROUTINE SUMPRO ( A, B, SUM, PRO ).  
REAL A (20), B (20), SUM (20), PRO (20).  
1 SUM (1) = A (I) + B (I)  
PROD (I) = A(I) * B(I)  
1 = I + 1
```

```
IF (I - 20) 1, 1, 2
2 RETURN
END
```

La manera como se transfiere el control se ilustra en la siguiente figura. El control se transfiere a la subrutina cada vez que es llamada. El control se transfiere al programa principal (u otra subrutina) cuando encuentra la proposición RETURN. La siguiente proposición a ejecutar es la siguiente A la llamada.



La estructura de la subrutina es la siguiente:

```
# Tarjetas de control
```

```
Real a (1000)
```

```
.
```

```
.
```

```
.
```

```
.
```

```
Call error
```

```
.
```

```
.
```

```
.
```

```
.
```

```
End.
```

```
Subroutine error
```

```
(WRITE) ^ (6, 1)
```

```
1 Format ("un error del tipo a ocurrió")
```

```
Return
```

```
End
```

Cuando una subrutina es usada, algunos de los argumentos (argumentos en la subrutina de referencia) pueden ser expresiones, veamos a través de un ejemplo como son estos tratados.

```
Programa .
```

```
.
```

```
Principal .
```

```
Call suma ( A * A, B )
```

```
.
```

```
.
```

```
.
```

```
.
```

```
End
```

Cuando ejecuta la preposición call suma (A * B, B), -
 la expresión A * A, es valuada y su valor es asignado a una
 variable temporal no accesible al programador llamémosle t.
 El segundo argumento es simplemente una variable, su nom--
 bre es pasado a la subrutina suma. O sea que call suma (A*
 A, B) es equivalente a;

$$t = A * A$$

A esta manera de tratar argumentos se le conoce como
 "Llamada por valor;

En general, una subrutina admite determinados valores de
 entrada y "regresa" al programa principal otros valores, por
 ejemplo, en la subrutina SUMPRO (A, B, SUM, PRO) los valo--
 res de entrada son A, B; y los valores de regreso son SUM y
 PRO.

Existe otro tipo de subrutinas que regresan un solo va-
 lor, por lo que son llamadas funciones. En este tipo de sub-
 rutinas todos los argumentos representan valores de entrada y
 el valor de salida queda asociado al nombre de la subrutina.

Veamos un ejemplo: queremos una subrutina que admita --
 tres valores A, B, C y calcule $A^2 + B^2 + C^2$ si A y B son po-
 sitivos o $-\frac{C}{A+B}$ si A ó B son negativos, se procede a decla-
 rar la subrutina:

```

FUNCION      F ( A, B, C )
IF ( A, GE. 0. OR. B. GE. 0 ) GO TO 2
F = - ( C / ( A + B ) ) .
RETURN
2  F + SORT ( A * * 2 + B * * 2 + C * * 2 )
RETURN
END

```

METODO DE NEWTON

Este método sea tal vez el método más popular para encontrar los ceros (raíces) de una función de una variable $f (X)$. Es decir se encuentra una X tal que $f (X) = 0$.

El método de Newton es un método iterativo que produce una secuencia de aproximación a la raíz, siempre y cuando:

- a) $f (X)$ sea continua y diferenciable en la vecindad de la raíz, y que las segundas derivadas de $f (X)$ no lleguen a ser excesivamente grandes.
- b) Se puede dar un intento inicial del valor de la raíz "bueno".

Para funciones de variables reales, el método de Newton tiene una interpretación geométrica simple como se ilustra en la siguiente figura:

Suponga que queremos encontrar una raíz de la función $f(x)$, es decir el punto donde $f(x)$ corta el eje x . Supongamos que la curva tiene la forma de la figura anterior, si nuestro primer intento es x_1 , x_2 será una mejor aproximación de la raíz la cual se obtiene encontrando la intersección de la tangente $(x_1, f(x_1))$ en el eje x . Este proceso se repite varias veces, cada vez utilizando la x_2 calculada de la x_1 anterior; hasta encontrar la raíz con la aproximación deseada.

Refiriéndose nuevamente a la figura anterior deje que $f^1(x)$ sea la derivada de $f(x)$ valuada en el punto x_1 , por consideraciones geométricas.

$$\hat{f}^1(x_1) = \frac{f(x_1)}{x_1 - x_2} \quad 1$$

Y la nueva aproximación de la raíz

$$x_2 = x_1 - \frac{f(x_1)}{f^1(x_1)} \quad 2$$

Ejemplo:

Método para calcular la raíz cuadrada.

Si la ecuación $x^2 = A$, la ecuación a resolver será:

$$f(x) = A - x^2 = 0$$

$$f^1(x) = -2x$$

De la fórmula 2

$$x_2 = x_1 - \frac{A - x_1^2}{-2x_1} \quad \text{ESCRIBIENDO: } x_2 = \frac{1}{2} \frac{A}{x_1} + x_1$$

METODO DE NEWTON

```

C   PROGRAMA ENCONTRAR LOS CEROS DE UNA FUNCION
C
C
C   AQUI SE DEFINEN LA FUNCION Y SU DERIVADA
C
F (X) =
DF (X) =
C   LEE EL VALOR INICIAL DE LA SOLUCION Y LA TOLERANCIA
    DE ERROR
    READ (5, 1 )  XVIEJA, EPS
1  FORMAT ( ' 2 F10.0 )
C   COMIENZAN LAS ITERACIONES
2  XNUEVA = XVIEJA - F (XVIEJA) / DF (XVIEJA)
    DIF = ABS (XVIEJA - XNUEVA)
    IF ( DIF. LT. EPS ) GO TO 3
    XVIEJA = XNUEVA
    GO TO 2
3  WRITE (6, 4) XNUEVA, DIF
4  FORMAT ( " X = " , F12.4, 5 X, "ERROR = " , E14.10 )
    END

```



centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



METODOS NUMERICOS Y APLICACIONES CON LA COMPUTADORA
DIGITAL

TEMA I : LENGUAJE FORTRAN

(Complemento)

PROF. ING. HERIBERTO OLGUIN ROMO.
PROF. ING. RICARDO CIRIA MERCE.

ABRIL, 1978.

HERIBERTO OLGUIN ROMO

RICARDO CIRIA MERCÉ

F O R T R A N

ELEMENTOS DE UN SUPERLENGUAJE DE PROGRAMACION: FØRTRAN

- 1.- Introducción al lenguaje FØRTRAN
 - 1.1 El alfabeto
- 2.- Números
 - 2.1 Constantes enteras
 - 2.2 Constantes reales
 - 2.3 Variables enteras
 - 2.4 Variables reales
- 3.- Operaciones aritméticas
- 4.- Expresiones aritméticas
 - 4.1 Reglas para las expresiones aritméticas
 - 4.2 Funciones predefinidas disponibles en lenguaje FØRTRAN
- 5.- Enunciados
 - 5.1 Los enunciados aritméticos de asignación
 - 5.2 Los enunciados de control
 - 5.2.1 El enunciado GØ TØ

- 5.2.2 El enunciado IF
- 5.2.3 El enunciado DØ
- 5.2.4 El enunciado STØP
- 5.3 Los enunciados de entrada y salida
 - 5.3.1 El enunciado READ
 - 5.3.2 El enunciado WRITE
- 5.4 Los enunciados de especificación
 - 5.4.1 El enunciado FØRMAT
 - 5.4.1.1 La especificación I : Iw
 - 5.4.1.2 La especificación F : Fw.d
 - 5.4.1.3 La especificación E : Ew.d
 - 5.4.1.4 La especificación A : Aw
 - 5.4.1.5 La especificación T : Tw
 - 5.4.1.6 Las especificaciones X, H y /
 - 5.4.2 El enunciado END
- 6.- Arreglos
 - 6.1 Variables con subíndices
 - 6.1.1 Reglas para los subíndices
 - 6.2 El enunciado DIMENSIØN
 - 6.2.1 Reglas para las variables con subíndices
 - 6.3 Arreglos de entrada y salida
- 7.- SUBPROGRAMAS
 - 7.1 Funciones
 - 7.1.1 Ejemplos
 - 7.2 Subrutinas
 - 7.2.1 CØMMØN

1.- Introducción al lenguaje FØRTRAN

El lenguaje FØRTRAN, cuyo nombre corresponde a las primeras letras de las palabras inglesas FORMula-(fórmula) y TRANslation (traducción), es un lenguaje de programación orientado a problemas ^{matemáticos} y se emplea en casi todas las computadoras del mundo. Debido a su parecido con el lenguaje aritmético común, el FØRTRAN simplifica la preparación de problemas que pueden resolverse mediante una computadora. Los datos e instrucciones se pueden organizar mediante una secuencia de enunciados fortran; estos constituyen el llamado Programa Fuente.

Todas las computadoras que "entienden" el lenguaje FØRTRAN, tienen lo que se llama un Compilador Fortran, llamado también traductor o interprete, el cual analiza los enunciados fortran y los traduce a un Programa Objeto, el cual queda en Lenguaje de Máquina.

Un programa escrito en lenguaje FØRTRAN se puede procesar en cualquier máquina que tenga un Compilador Fortran. Esto nos indica que el lenguaje es independiente para cada máquina, o sea que el compilador se debe preparar en cada caso teniendo en cuenta la máquina que ha de usarse en particular; puesto que las máquinas difieren en su organización interna, se ha desarrollado un número de "dialectos" del Lenguaje FØRTRAN, cada uno de los cuales es apropiado para una clase de máquinas. Las diferencias entre los

varios dialectos son mínimas y se ajustan el uno al otro fácilmente.

1.1 El alfabeto

El alfabeto FØRTRAN esta constituido de caracteres que son símbolos familiares de escritura y de teclados de máquinas de escribir, así como de dispositivos especiales de perforación; dichos caracteres son:

Alfabéticos: A B C D E F G H
 I J K L M N
 * Ø P Q R S T U V W X Y Z

Numéricos: 0 1 2 3 4 5 6 7 8 9

Símbolos: + - * / = . , () ' @

De este alfabeto se construyen todos nuestros símbolos, expresiones y enunciados que se utilizan en el lenguaje FØRTRAN.

2.- Números

Los números pueden representarse en diferentes formas, las cuales se asemejan a los símbolos de la aritmética general; pero debido a la estructura interna de las computadoras se establecen las convenciones de: Punto Fijo y Punto Flotante que proporcionan facilidades para su manejo en FØRTRAN. Los símbolos de punto fijo

* La letra O la expresaremos como Ø para diferenciarla del N° cero.

se usarán solamente con números enteros y los cálculos asociados se denominarán aritmética de los enteros o modo entero; mientras que la aritmética de los números reales se hará en la forma de punto flotante y se llamará aritmética de los reales o modo real. Debido a que también es necesario distinguir las constantes (números que no cambian durante toda la ejecución de un programa) de las variables (números que pueden cambiar), surgen cuatro clases de símbolos para los números.

2.1 Constantes enteras.

Dependiendo del tipo de computadora se podrán representar por un cierto número de dígitos, así para IBM-1130 se representan mediante cinco dígitos sin el punto decimal. Si el entero es negativo, los dígitos deberán ser precedidos del signo menos; si el entero es positivo el signo es opcional.

Ejem. Símbolos para constantes enteras pueden ser entre otras:

1976 +1 0 +1976 -1976

Símbolos que no se aceptan para constantes enteras:

7483282 (más de cinco dígitos)

1976. (el punto decimal no se permite)

2.2 Constantes reales

Dependiendo del tipo de computadora, las constantes reales se podrán representar por varios dígitos, pero en el caso de la

IBM-1130 sólo se admiten siete dígitos con punto decimal pudiéndose colocar al principio de los dígitos, al final o entre dos dígitos cualesquiera. Cuando aparece un punto en una constante su tratamiento será de punto flotante. Si la constante real es precedida de un signo menos, se indicará que es negativa, si es positiva el signo es opcional.

Ejem. Símbolos para constantes reales pueden ser entre otras:

1976.	-.00001976	+12.345	-12.345
-.007	.007	5.348	0.3

Símbolos que no se aceptan para constantes reales:

123456789.32 (más de siete dígitos significativos)

5343 (falta el punto decimal)

Para representar las constantes reales existe también la llamada forma exponencial; esta la podemos representar mediante una letra E y una constante entera de uno o dos dígitos, positiva o negativa. Esta constante entera es un exponente del número diez; el signo menos es para los exponentes negativos y para los positivos, el signo es opcional. En FORTRAN, la presencia del exponente hace que el uso del punto decimal sea opcional.

Ejem.	Forma exponencial	Forma no exponencial
	1.328E2	132.8
	1.328E02	132.8

1.328E00	132.8
-4.724E-03	-.004724
+7.61E3	7610.
-6432E-3	-6.432

2.3 Variables enteras

Estas se representan por combinaciones de una a cinco letras y dígitos (IBM-1130), no se permiten otros caracteres y el primer caracter deberá ser una de las letras I, J, K, L, M ó N. El primer caracter de una variable es el que indica si es entera o real. Durante la ejecución de un programa, las variables enteras deberán restringirse a valores enteros.

Ejem. Símbolos para variables enteras pueden ser, entre otros:

NUMCT	KILO	N1	N2	M10	KONT
IIALC	JCLAV	MARY	KONT1	L1976	

Símbolos no aceptables para variables enteras:

CUENT (el primer caracter debe ser I, J, K, L, M ó N).

KONTADOR (demasiados caracteres)

12.34 (sólo se aceptan letras y números)

2.4 Variables reales

Estas se representan por combinaciones de una a cinco letras y dígitos (IBM-1130), no se permiten otros caracteres y el primer

caracter tiene que ser necesariamente una letra diferente a I, J, K, L, M ó N. Durante la ejecución de un programa dichas variables se deben restringir a valores reales.

Ejem. Símbolos para variables reales pueden ser, entre otros:

FUERZ VELOC ACEL1 CUENT A1 A2
ALFA VIELA RA42 X1 PROD SUMA

Símbolos no aceptables para variables reales:

A3.8 (el punto no es letra o número)

CORRIEN (demasiados caracteres)

3 BASO (el primer caracter debe ser una letra)

MUMCT (el primer caracter no puede ser M)

3.- Operaciones aritméticas

Las operaciones aritméticas y los símbolos que se utilizan en FØRTRAN son:

	Ejem.	Algebra	FØRTRAN
Adición	+	$a + b$	A + B
Sustracción	-	$a - b$	A - B
Multiplicación	*	$a b$	A * B
División	/	$\frac{a}{b}$	A / B
Exponenciación	**	a^2	A * A
		a^2	A** 2

4.- Expresiones aritméticas

En base a lo expuesto anteriormente podemos ahora formular

expresiones aritméticas en lenguaje FØRTRAN y nos daremos cuenta que son muy similares a las expresiones aritméticas del algebra común.

Expresiones FØRTRAN	Expresiones Comunes
A**2-B**2	a^2-b^2
B**2-4.*A*C	b^2-4ac
(A+B)/2.	$\frac{1}{2}(a+b)$
2*K-J+N	$2k-j+n$
C+B-3.*A	$c+b-3a$

4.1 Reglas para las expresiones aritméticas

Las reglas a las que debemos sujetar las expresiones aritméticas son necesarias debido a la estructura de las computadoras y al observarlas tendremos un ahorro en el tiempo de ejecución de un programa.

Regla 1

Si nos fijamos en las expresiones FØRTRAN anteriores nos damos cuenta que: Todas las constantes y variables en una expresión deben estar en el mismo modo, esto es, todas deben ser enteras o todas deben ser reales. (Como toda regla existe su excepción que mencionaremos más adelante).

Es necesario consultar los manuales de cada máquina, ya que como hemos mencionado anteriormente dependerá esta regla del tipo de computadora. Por lo pronto la consideraremos como se ha indicado.

Regla 2

$A^{**}I$, $I^{**}J$ y $A^{**}B$ son exponenciaciones permitidas. En el caso $A^{**}I$ se mezclan los modos y es la excepción a la Regla 1, pero sabemos que esta exponenciación significa multiplicaciones sucesivas (así $B^{**}3 = B*B*B$), mientras que las potencias no enteras implican cálculos más sofisticados. Nos damos cuenta que $I^{**}A$, no es forma de exponenciación permitida (en algunas máquinas sí se permite).

Regla 3

Deberá tenerse en cuenta que las operaciones se ejecutarán con las siguientes prioridades:

- 1) Las operaciones indicadas dentro de los paréntesis más internos se ejecutan en primer lugar.
- 2) Exponenciación.
- 3) Multiplicación y división.
- 4) Adición y sustracción.

Entre las operaciones de igual prioridad, el orden de ejecución es de izquierda a derecha.

Ejem. Si $A=5.$, $B=8.$ y $C=2.$

$A+B-3.*C$ se calculará en el siguiente orden:

$$3.*2.=6. \quad 5.+8.=13. \quad 13.-6.=7.$$

$B^{**}2-4.*A*C$ se calcula en el siguiente

orden:

$$8.**2=64. \quad 4.*5.=20. \quad 20.*2.=40.$$

$$64.-40.=24.$$

Si $A=5.$, $B=8.$, $C=2.$ y $D=1.6$

Entonces $(A+B)/C$ se calcula en el siguiente orden:

$$5.+8.=13. \quad 13./2.=6.5$$

Mientras que $A+B/C$ se calcula en el siguiente orden:

$$8./2.=4. \quad 5.+4.=9.$$

Ahora si deseamos calcular $(A+C)**2$ conducirá a:

$$5.+2.=7. \quad 7.**2=49.$$

Mientras que $A+C**2$ conducirá a:

$$2.**2=4. \quad 5.+4.=9$$

Ahora si: $(A*B)/(C*D)=40./3.2=12.5$

Entonces: $A*B/C*D=40./C*D=20.*D=32.$

Finalmente si tenemos paréntesis dentro de otros paréntesis se tiene:

$$(A*(B+C))**2=(A*10.)**2=50.**2=2500.$$

$B+C$ tiene la más alta prioridad por encontrarse en el paréntesis más interno.

$$(A*B+C)**2=(40.+2)**2=42.**2=1764.$$

$$A*(B+C)**2=A*10.**2=A*100.=500.$$

$$A*B+C**2=A*B+4.=40.+4.=44$$

Debemos tener cuidado en expresar lo que deseamos realizar.

Regla 4 No deberemos colocar un signo de operación antes de un signo más o menos, esto es, no deberemos poner dos signos de operación juntos.

Ejem. $A*-B$ $I+-J$ $M-+N$ $A/-B$

Estas expresiones deberán sustituirse por:

$A*(-B)$ $I+(-J)$ $M-(+N)$ $A/(-B)$

4.2 Funciones predefinidas en lenguaje FØRTRAN

Estas funciones predefinidas que proporciona el lenguaje FØRTRAN son de tipo de biblioteca. Para utilizarlas usaremos el nombre de la función seguido de un argumento que deberá estar entre paréntesis. Dichos argumentos pueden ser variables simples ó con subíndices, constantes, expresiones aritméticas u otras funciones predefinidas en FØRTRAN.

Para IBM - 1130 tenemos:

<u>NOMBRE</u>	<u>FUNCION EJECUTADA</u>	<u>NUM. DE ARGUMENTOS</u>	<u>TIPO DE ARGUMENTO(S)</u>	<u>TIPO DE FUNCION</u>
SIN	Seno trigonométrico (argumento en radianes)	1	Real	Real
COS	Coseno trigonométrico (argumento en radianes)	1	Real	Real
ALOG	Logaritmo natural	1	Real	Real
EXP	Argumento de potencia del número e.	1	Real	Real
SQRT	Raíz cuadrada	1	Real	Real
ATAN	Arco tangente	1	Real	Real

del número e

ABS	Valor absoluto	1	Real	Real
IABS	Valor absoluto	1	Entero	Entero
FLOAT	Convertir argumento de entero a real	1	Entero	Real
IFIX	Convertir argumento de real a entero	1	Real	Entero
SIGN	Transferencia de signo (Arg.1 recibe signo de Arg.2)	2	Real	Real
ISIGN	Transferencia de signo (Arg.1 recibe signo de Arg.2)	2	Entero	Entero
TANH	Tangente Hiperbólica	1	Real	Real

Ejem. $\text{SQRT}(B^{**2}-4.*4.*A*C)$ indica que a lo que se encuentra entre paréntesis se le sacará la raíz cuadrada.
 SIN (BETA) indica que se obtendrá el seno trigonométrico de el valor de la variable BETA.

5.- Enunciados

Los enunciados son las unidades básicas con las cuales se construyen los programas FORTRAN. Podemos clasificarlos de acuerdo a su función en grupos como:

- 1.- Aritméticos de asignación
- 2.- De control
- 3.- De entrada y salida
- 4.- De especificación

5.1 Los enunciados aritméticos de asignación

Se forman con las expresiones presentadas anteriormente y

nos indican los cálculos particulares que deben hacerse. Su forma es:

Variable = Expresión aritmética

El significado del signo = es el de asignación, esto es, que deberá calcularse el valor de la expresión a la derecha del signo = y su valor se asignará a la variable que se encuentre a la izquierda del signo, la cual tiene una localidad en la memoria de la computadora.

Ejem. Si A=5., B=8., C=2. y D=1.6

$X=(A+B)/C$ se le asignará a la X el valor 6.5

$ALO=(A+B)**2$ se le asignará a ALO el valor 169.

$RAI=SQRT(B*C)$ se le asignará a RAI el valor 4.

Algo diferente al algebra normal es el enunciado

$A=A+3$. el cual no debe alarmarnos ya que indica que a la localidad de memoria con el nombre A se le asignará el nuevo valor $A+3$. esto es:

Si $A=5$. y $A=A+3$. entonces:

$A=5+3$. $A=8$. o sea que la variable A se le asigna el valor de 8. y el valor anterior que fué 5. se pierde.

5.2 Los enunciados de control

Debido a que los enunciados de un programa FØRTRAN se ejecutan en el orden que aparecen y que en muchas ocasiones queremos transferir la ejecución a otros enunciados si se satisface una

cierta condición, FORTRAN nos permite numerar dichos enunciados. Un número de enunciado debe ser una constante entera de uno a cinco caracteres sin el signo más o menos; el número se coloca a la izquierda del enunciado.

Ejem. 3 CONT = CONT+1.
 24 RAIZ = SQRT (A**2+B**2)

5.2.1 El enunciado GØ TØ

Este toma la forma GØ TØ N en donde N es un número de enunciado.

El GØ TØ produce un salto incondicional; así GØ TØ 3 envía la ejecución al enunciado número 3 que puede ser la instrucción de conteo del ejemplo anterior. GØ TØ 24 pasa el control al enunciado 24 que puede ser el del ejemplo anterior.

Ejem. Supongamos que unos de los enunciados de un programa son:

I = 1	Esto nos representa la suma de
ISUM = 0	los números enteros, desde luego
1 ISUM = ISUM+1	es necesario ponerle otros enuncia-
I = I+1	dos pero por el momento nos aclara
GØ TØ 1	lo indicado.

5.2.2 El enunciado IF

Debido a que las computadoras están diseñadas a base de circuitos lógicos y el pensamiento del ser humano debe

ser de este tipo, nos concretaremos el IF lógico, además de que el alumno ya tiene elementos de algunos operadores de relación como OR, AND y NOT.

El IF lógico es de la forma:

IF (L) S

L= expresión lógica que puede tener dos valores: Verdadero o Falso.

S= cualquier enunciado FORTRAN diferente de: un DO, un enunciado de especificación o de otro IF lógico.

Si L es falso (.FALSE.) entonces se ignora S y la computación continúa al siguiente enunciado. Si L es verdadero (.TRUE.) el enunciado S se ejecuta en seguida.

Resulta interesante hacer notar que si L es relativamente complicada, éste IF puede ser el equivalente de varios IF aritméticos.

Para formar las expresiones lógicas (L) utilizaremos los operadores de comparación y los de relación.

Operadores de comparación:

<u>Símbolo Matemático</u>	<u>Significado</u>	<u>Símbolo FORTRAN</u>	<u>Significado Inglés</u>
<	Menor que	.LT.	Less than
>	Mayor que	.GT.	Greater than
≤	Menor o igual a	.LE.	Less or equal
≥	Mayor o igual a	.GE.	Greater or equal

$=$	Igual a	.EQ.	Equal
\neq	Diferente a ó No igual a	.NE.	Not equal
Operadores de relación:			
\cup	Unión	.OR.	ó ("o inclusive)
\cap	Intersección	.AND.	y ("al mismo tiempo)
$-$	Complemento	.NOT.	no

Para valuar una expresión lógica se hará con las siguientes prioridades:

- 1.- Expresiones entre paréntesis
- 2.- Operadores aritméticos
- 3.- Operadores de comparación (.LT., .GT., .LE., .GE., .EQ. y .NE.)
- 4.- .NOT.
- 5.- .AND.
- 6.- .OR.

En caso de igual jerarquía la evaluación será de izquierda a derecha.

Ejem. (1) $X=5.$ $y=0.5$

IF (X.GT.3..AND. Y .LE.2.) Z=X**3+X*Y

Significa que si $X > 3.$ y (al mismo tiempo) $y \leq 2.$ se asignará a Z el valor que se obtenga al calcular $X^3 + XY$, esto es $Z = 125. + 2.5 = 127.5$

(2) IF (A.LE.X.AND.B.GE. Y .OR.C.GT.Z) GO TO 12

Significa que si $A \leq X$ y (al mismo tiempo) $B \geq Y$ es

verdadero ó $C > Z$ es verdadero ó ambos, entonces se transfiere el control al enunciado 12.

```
(3)  I = 1
      ISUM = 0
1     ISUM = ISUM+1
      I = I+1
      IF (I.LE.100) GO TO 1
      STOP
```

Esto nos indica que sólo sumaremos los números enteros del 1 al 100

5.2.3 El enunciado DØ

Este toma la forma:

$$DØ K I = L, M, N$$

$$DØ K I = L, M$$

La segunda forma sólo se aplica cuando $N=1$, lo que es bastante frecuente.

K representa un número de enunciado

I representa una variable entera

L, M, N son variables enteras ó constantes sin signo.

El DØ produce la ejecución repetida de todos los enunciados que le siguen, hasta el enunciado número K .

La primera vez que se ejecutan estos enunciados la variable I es igual a L , en cada paso subsiguiente I se incrementa en la cantidad N , hasta hacerse mayor ó igual a M en el paso final; en este momento se termina el llamado

lazo $D\emptyset$ y el control pasa al enunciado que está a continuación del enunciado K. Así, L es el valor inicial de la variable I y M su valor final. I se llama el índice del enunciado $D\emptyset$ y su valor corriente se puede usar en cálculos durante la ejecución del lazo. Todos los enunciados que le siguen al $D\emptyset$ hasta el número K inclusive constituyen el rango del $D\emptyset$. También es posible que la variable I no se encuentre en ninguno de los enunciados del rango del $D\emptyset$ y esto nos indica que se realice la ejecución de todos los enunciados del rango del $D\emptyset$ M entre N veces (la parte entera de este cociente M/N). Debemos tomar en cuenta que: el índice I se incrementa secuencial y automáticamente durante la ejecución del lazo y que se puede, en estos momentos, tratar como cualquier variable entera; el índice I queda indefinido después de terminado el lazo del $D\emptyset$ y puede utilizarse para cualquier uso general. El enunciado K no debe ser un enunciado de especificación ni una transferencia de control esto incluye cosas como $G\emptyset$, $T\emptyset$, $.IF$ y $D\emptyset$, así como $FORMAT$, END y algunos otros. Debemos considerar que no se puede desde ningún punto del programa llegar a un enunciado dentro del rango de un $D\emptyset$. Y que la entrada a un $D\emptyset$ deberá hacerse a través del enunciado $D\emptyset$. Y por último es muy frecuente que un $D\emptyset$ esté completamente dentro de otro.

Ilustrando graficamente tenemos:

Correcto

Incorrecto

DØ DØ

DØ DØ

DØ DØ

DØ DØ

DØ

DØ

Ejém. Utilizaremos un DØ para sumar los números enteros del 1 al 100, ejemplo que ya hemos visto anteriormente.

ISUM = 0

Nos damos cuenta que el DØ tie-

DØ 1 I = 1,100

ne la misma función que un IF,

1 ISUM = ISUM+1

un GØ TØ y un contador; como po-

STOP

drá observarse con el ejemplo anterior.

5.2.4 El enunciado STØP

Este aparece simplemente como STØP y es el que nos indica que ha terminado la ejecución y en el caso de IBM - 1130 la computadora se detiene y el operador tendrá que hacer que continúe trabajando. Debido a ello se recomienda que se utilice el enunciado CALL EXIT, el cual pasa el control a un programa monitor que hace que la computadora continúe ejecutando los otros programas que siguen a continuación.

Tanto el STØP como el CALL EXIT podrán aparecer después de cualquier enunciado.

5.3 Los enunciados de entrada y salida

Estos, como su nombre lo indica, sirven para introducir y sacar información de la computadora.

5.3.1 El enunciado READ

Este enunciado tiene la forma READ (I, N) LISTA. I y N son enteros sin signo y LISTA representa una lista de nombres de variables para las cuales se leerán valores. I designa el tipo de periférico de entrada que se utilice (lectora de tarjetas, consola, etc.). N es el número de un enunciado FØRMAT asociado al READ.

Ejem. El enunciado READ (2, 101) J, B, H

Producirá la lectura de tres números: un entero y dos reales y se almacenarán en las localidades de la memoria de la computadora designadas con las variables J, B y H en su orden. Las comas que separan éstos nombres de variables en el READ son indispensables, 2 es la unidad de entrada y 101 un FØRMAT.

5.3.2 El enunciado WRITE

Este tiene la forma WRITE (I, N) LISTA. I y N son enteros sin signo y LISTA representa una lista de variables para las cuales se imprimen valores. I designa el tipo de periférico de salida que se utilice (Impresora, cinta, etc.). N es el número de un enunciado FØRMAT aso-

ciado al WRITE.

Ejem. El enunciado WRITE (3, 108) L, X, Y
 Producirá que se impriman los valores de las variables L, X y Y que se encuentren en las localidades de memoria con esos nombres, en el formato especificado por el enunciado número 108 y por la unidad de salida número 3; las comas que separan éstos nombres de variables en el WRITE son indispensables.

5.4 Los enunciados de especificación

Este tipo de enunciados no inician por si mismos los cálculos, no producen transferencia de control ni estimulan el flujo de información, pero proveen al compilador FORTRAN de los detalles esenciales para la traducción del programa fuente en FORTRAN al programa objeto en lenguaje de máquina ó para la conversión de datos a la entrada o la salida.

Si queremos introducir datos a la computadora lo podemos hacer mediante un enunciado que esté dentro del programa, como $A = 3.1416$, ésto es lo que podríamos llamar inicializar una variable; y el programa se compilaría cada vez que quisieramos darle un valor diferente a A, lo cual resulta muy costoso, ya que las compilaciones son laboriosas. Para evitar esto se usa el enunciado READ y los valores que se le den a A podrán estar en tarjetas de datos, los cuales son independientes del programa.

ma fuente.

5.4.1 El enunciado FØRMAT

Este tiene la forma: N FØRMAT (, , , ...) en la cual N es el número del enunciado FØRMAT y corresponde al N de los enunciados READ y WRITE. Los espacios entre las comas están disponibles para las especificaciones del tipo que se describen más adelante, siendo el número de espacios uno o más, de acuerdo a las necesidades del programador.

5.4.1.1 La especificación l:lW

Aquí l indica un valor entero y W es un entero que indica el número de columnas o ancho de campo, que ocupa ese valor en la tarjeta de entrada o en el papel de impresión. El número w deberá incluir un lugar para el signo de ese valor, siendo + opcional.

Ejem. Valor de los datos

de entrada o salida: 1130 +1620 -370 0 +14

Especificación: 14 15 14 11 13

5.4.1.2 La especificación F:Fw.d

Aquí F indica un valor real, w indica el número de columnas que ocupará el valor en la tarjeta de entrada o en el papel de impresión; d indica el número de cifras que se encontrarán des-

pués del punto decimal. w deberá incluir un lugar para el signo y otro para el punto decimal.

Ejem. Valor de los datos de
 entrada ó salida: 32.787 - .007 1130. +3.70
 Especificación: F6.3 F5.3 F5.0 F5.2

5.4.1.3 La especificación E:Ew.d

Aquí E indica un valor real en forma exponencial y w indica la anchura de campo para ese valor y debe de incluir el signo, si lo hay, el punto decimal, el lugar para la letra E, un lugar para el signo del exponente, si es negativo, y dos lugares para el exponente; d indica el número de dígitos a la derecha del punto decimal.

Ejem. Valor de los datos
 de entrada o salida: .1403E04 - .7E-02 .1442E+04
 Especificación: E8.4 E7.1 E9.4

Es conveniente que cuando deseemos sacar información de la computadora, tomemos en cuenta para el ancho del campo lo siguiente:

- 1.- El signo, aún cuando el + generalmente no se imprime.
- 2.- El punto decimal para las especificaciones F y E.
- 3.- Por lo menos un dígito a la izquierda

del punto decimal, puesto que muchas máquinas imprimirán allí un cero si otro dígito no ocurre.

- 4.- Suficientes lugares para todos los dígitos significativos deseados, debido a que para los dígitos que no se les deja espacio se truncan o redondean.
- 5.- Cuatro lugares para el exponente de la especificación E.
- 6.- El primer lugar se deja en blanco para el control de carro.

5.4.2 El enunciado END

Este se lee simplemente END e informa al compilador que el programa fuente ha terminado y debe ser el último enunciado de cualquier programa FORTRAN.

6.- Arreglos

Frecuentemente tratamos con un grupo de variables que forman ó pertenecen a una clase o colección. Cuando las variables forman un conjunto ordenado, pueden relacionarse unas con otras por la notación de subíndices; entonces designamos esa colección como arreglo y las variables que pertenecen a ésta serie son los elementos del arreglo. A veces se emplea como sinónimo de

arreglo el nombre de matriz y, en consecuencia, hablamos de elementos de la matriz.

6.1 Variables con subíndices

Un conjunto de números que pueda arreglarse en un renglón ó columna se considera como un arreglo lineal ó unidimensional, y ésta serie puede llamarse vector. Identificamos los elementos de un vector renglón ó columna por un sólo subíndice.

Ejem. La columna de números del vector llamado A, consiste de los elementos A_1 , hasta A_n inclusive y se representa como sigue:

Notación asoctumbrada	Notación FORTRAN
A_1	A (1)
A_2	A (2)
A_3	A (3)
A_i	A (I)
A_n	A (N)

Cada una de estas $A(I)$, en donde I varía de 1 a N, son el nombre de una variable, el conjunto de todas ellas es lo que llamamos arreglo.

Si se usan dos subíndices para identificar los elementos de un arreglo se considera éste como un arreglo bidimensional. Los cuadros de un tablero de ajedrez, pueden considerarse como un arreglo bidimensional. Y si llamamos a cualquiera de los cua-

dros con la variable CTAJ tendremos 64 variables; pero como el tablero tiene 8 renglones y 8 columnas, podemos referirnos al cuadro que se encuentra en el renglón 3 y la columna 5 con la variable CTAJ-(3,5).

Dependiendo del tipo de computadora será el número de subíndices que podremos asignarle a un arreglo; en IBM - 1130 sólo se admiten arreglos con un máximo de tres subíndices.

Las variables que se utilicen para designar arreglos deberán observar las reglas que se dieron anteriormente al hablar de variables enteras y reales considerando que para los cinco caracteres alfanuméricos son independientes de los índices que se encuentran entre paréntesis.

6.1.1 Reglas para los subíndices.

Regla 1 Un subíndice debe ser un entero, puede ser constante, variable ó una de las expresiones aritméticas siguientes:

$$A * V + b \quad A * V - b$$

en donde v es una variable entera y a y b son constantes enteras sin signo.

Ejem. Algunos subíndices pueden ser:

$$1 \quad 1972 \quad 10 * KONT \quad 2 * I \quad J$$

$$1976 * N - 8 \quad 2 * I - 4 \quad 2 * I + 3$$

No se pueden usar como subíndice:

$$1 + I \quad -I \quad 2 - 10 * CONT \quad -1932 \quad -KILO$$

Regla 2 Un subíndice sólo debe tomar valores positivos.

Regla 3 Un subíndice en sí no debe ser una variable con subíndices. Así $X(I(2))$ no es permitido.

Regla 4 Un símbolo que representa un arreglo, una variable con subíndice, no debe usarse sin subíndices para representar otra variable diferente en el mismo programa. Esto es $A(I)$ y A no deben referirse a variables diferentes. Como siempre hay una excepción que por ahora no tocaremos.

Ejem. Los símbolos para variables reales con subíndices podrían incluir:

$X(I)$ $SUM(K+2)$ $A(I, 2*J+1)$ $B(INT)$ $C(I,J)$

Para variables enteras con subíndices podemos tener:

$INT(M,N)$ $I(J)$ $ICTA(J,2*I)$

6.2 El enunciado DIMENSION

Siempre que en un programa utilicemos variables con subíndices deberemos poner como primer enunciado el DIMENSION, el cual indica al compilador qué tanto espacio de memoria se debe reservar para las variables con subíndices. Su forma es:

DIMENSION u, v, w, \dots

Donde u, v, w, \dots son nombres de variables, cada una de las cuales va seguida por el máximo número de elementos en el

arreglo correspondiente. Deberán observarse las siguientes reglas:

Regla 1 Cada variable con subíndices se debe mencionar en un enunciado DIMENSION antes de su primer uso en el programa.

Regla 2 Los símbolos representados anteriormente por u, v, w, ... deben tener la forma:

nombre de variable (máximo número de elementos)

el número entre paréntesis debe ser una constante entera sin signo.

Ejem. DIMENSION A(20), B(4,8), CARR(5,3,4)

Esto indica que el compilador reservará 20 localidades para el arreglo A, sus veinte variables serán A(1), A(2), ..., A(20) al mismo tiempo se reservarán 32 (4x8) localidades para las variables B(1,1), B(1,2), B(1,3), ..., B(1,8), B(2,1), B(2,2), ..., B(2,8), B(3,1), B(3,2), ..., B(3,8), B(4,1), B(4,2), ..., B(4,8) y por último se reservarán 60 (5x3x4) localidades para las variables del arreglo CAR, con tres subíndices cada una.

Regla 3 El arreglo que se use en particular, dentro del programa podrá tener menos elementos que los especificados en la magnitud del enunciado DIMENSION, pero no más.

Regla 4

La variable tal como aparece en el enunciado DIMEN-
SION debe tener exactamente el mismo número de sub-
índices que en cualquier otra parte del programa.

7.- SUBPROGRAMAS.

Los subprogramas, también llamados subrutinas, son programas que pueden ser puestos en uso por otros programas cuando sea necesario.

Las funciones de biblioteca ó funciones del sistema constituyen una variedad de subprogramas.

7.1- FUNCIONES

Cuando el valor de una variable depende de una ó más variables ó constantes y además de una serie de cálculos, y dicha variable ha de calcularse repetidamente y en diferentes puntos de un programa, es posible definirla como una función. En otras palabras, Además de las funciones con que cuenta la biblioteca del sistema, el usuario puede escribir sus propias funciones para uso específico de su programa.

Tomemos un ejemplo para visualizar lo anterior:

Supongamos que para un programa en especial, en el cual trabajamos con grados en lugar de radianes, deseamos calcular continuamente $\text{SENØ} (X)$, sin el uso de funciones sería necesario transformar el argumento deseado de grados a radianes y después llamar a la función del sistema $\text{SIN} (X)$. A continuación presentamos una función que calculará $\text{SENØ} (X)$, (X en grados) :

```

FUNCTION SENØ ( X )
  X = X * 3.141592/180.
  SENØ = SIN (X)
  RETURN
END

```

que es llamada desde el programa como:

```

GRAD= SENØ (GRADØS)

```

En base a éste ejemplo podemos generalizar el uso de la proposición FUNCTION .

- a) Debe ser codificada en forma independiente del programa que la usará, es decir, no debe aparecer "dentro" del programa.
- b) Debe empezar con la palabra FUNCTION

FUNCTION nombre (parámetro)

- c) A continuación se escribe el nombre con que será llamada.
 d) Después, entre paréntesis y separados por comas, aparecen los argumentos.

7.1.1 EJEMPLOS.-

```

FUNCTION RAIZ 1 (A,B,C )
  RAIZ1= (-B + SQRT (B **2 - 4. * A * C )) / ( 2.* A )
  RETURN
END
FUNCTION RAIZ2 ( A, B, C )
  RAIZ2 = ( -B - SQRT (B**2.4.* A * C )) / (2.* A )
  RETURN
END

```

```

C      EC. SEGUNDØ GRADØ
      READ ( 2,100) A, B, C
100    FORMAT ( 3F10.5)
      X1 = RAIZ1 (A,B,C)
      X2 = RAIZ2 (A,B,C)
      WRITE (3,200) A,B,C, X1,X2
200    FORMAT ( 5 ( ;,F10.5' )
      CALL EXIT
      END

```

Este ejemplo es solamente para mostrar el uso de la proposición FUNCTION y no contempla algunas situaciones como raíces complejas, etc.

7.2 SUBROUTINAS

Como es fácil notar, la proposición FUNCTION nos "regresa" un sólo valor y lo hace a través de su nombre. En muchos casos es conveniente ó necesario que se nos regrese más de un valor, para éstos casos usamos la proposición o enunciado:

SUBROUTINE.

Una subrutina es un subprograma que puede "recibir" cualquier número de parámetros (desde cero hasta un número determinado por el tipo de compilador) y puede "regresar" diferentes valores calculados.

Veamos algunos ejemplos:

Supongamos que al imprimir resultados de un cierto programa tenemos que escribir algún título usando los primeros renglones de la hoja. En tal caso podemos hacer uso de una subrutina como sigue:

```

SUBROUTINE ENCA
WRITE (3,200)
200 .. FORMAT--(/,1X, ' REPORTE SEMANAL' , / )
RETURN
END

```

Como vemos no hemos pasado ningún parámetro ó valor a la subrutina. Para que se ejecute ésta se debe hacer uso de la proposición CALL, de la siguiente forma:

```
CALL ENCA
```

dentro del programa y en el lugar donde deseemos que ocurra la impresión.

Discutamos ahora un ejemplo muy simple para ejemplificar el uso de parámetros. Hagamos una subrutina que "reciba" como entrada dos números, los sume y el resultado lo "regrese" en otra variable. Sean A y B los números a sumar, y C la variable en donde se pondrá el resultado.

```

SUBROUTINE SUMA (A,B,C)
C = A + B
RETURN
END

```

Es importante detenerse a ver el significado de los parámetros para las subrutinas:

La subrutina anterior SUMA puede ser llamada de diversas formas:

```

CALL SUMA (AA,BB,CC)
CALL SUMA (4, 7, X )
etc.

```

Como vemos, las variables A, B y C que aparecen en la subrutina son variables

mudas o dormidas y solo tienen sentido dentro de la subrutina. Veamos lo anterior:

Supóngase el siguiente programa:

```

X1= 3.
X2= 4.
CALL SUMA ( X1,X2,X3)
SUM= X3
WRITE (3,200) X1,X2,X3, SUM
200  FØRMAT(4 F10.5)
CALL EXIT
END

```

Se propone como ejercicio al lector que haga las veces de la máquina y escriba lo que ésta imprimiría.

La máquina imprimirá :

```

3.0      4.0      7.0      7.0

```

Una de las facilidades más útiles en subrutinas es la de ^{pasar} arreglos como parámetros, ej:

```

SUBROUTINE MAXIM (A, MAX )
DIMENSION      A ( 10)
-----
-----
-----
-----
-----
RETURN
END

```

Supóngase que ésta subrutina encuentra el elemento del arreglo A (10) con mayor valor y lo regresa a través de la variable MAX. Es importante notar que si pasamos como parámetro uno o más arreglos hay que dimensionarlos otra vez dentro de la subrutina, lo cual se puede hacer de al menos dos formas: 1)

2) Poniéndole dimensión 1 (UNO)

Ejemplo:

```
DIMENSION A (10), B (20)
```

```
-----
-----
-----
-----
```

```
CALL ØRDEN (A)
```

```
CALL MAXIM (B)
```

```
CALL MAXIM (A)
```

```
-----
-----
-----
-----
```

```
CALL EXIT
```

```
END
```

Caso 1:

```
SUBRØUTINE ØRDEN (X)
```

```
DIMENSION X (10)
```

```
-----
-----
-----
-----
```

```
RETURN
```

```
END
```

Caso 2:

```
SUBRØUTINE MAXIM (Y)
```

```
DIMENSION Y (1)
```

```
-----
-----
-----
```

```
RETURN
```

```
END
```

7.2.1 COMMON.-

Como es posible visualizar en los párrafos anteriores, las variables usadas en las subrutinas, o mejor dicho, dentro de las subrutinas, son totalmente independientes a las variables usadas en el programa principal. Muchas veces es conveniente que tanto las subrutinas como el programa que las llama tengan variables en COMUN. Para lograr esto existe la declaración

```
COMMON
```

La forma general de ésta proposición es:

```
COMMON lista de variables
```

donde "lista de variables" es un conjunto de variables y/o arreglos separados por comas a las cuales queremos adjudicarles la propiedad anterior, es decir, sean comunes a varios subprogramas.

Ej.

```
COMMON A,B, X (10), AB (30)
```

Esta declaración debe aparecer al principio de cualquier programa o subrutina en que se desee usar. Veamos un ejemplo:

```
C   SUMA DE DOS NUMEROS
COMMON A, B, C
A= 3
B= 7
CALL SUMA
Z = C
WRITE (3,200) A, B, C, Z
200 FORMAT( 4 F10.5 )
CALL EXIT
END
```

```
SUBROUTINE SUMA
COMMON A, B, C
C = A + B
RETURN
END
```

Este programa debe imprimir :

3.0 7.0 10.0 10.0

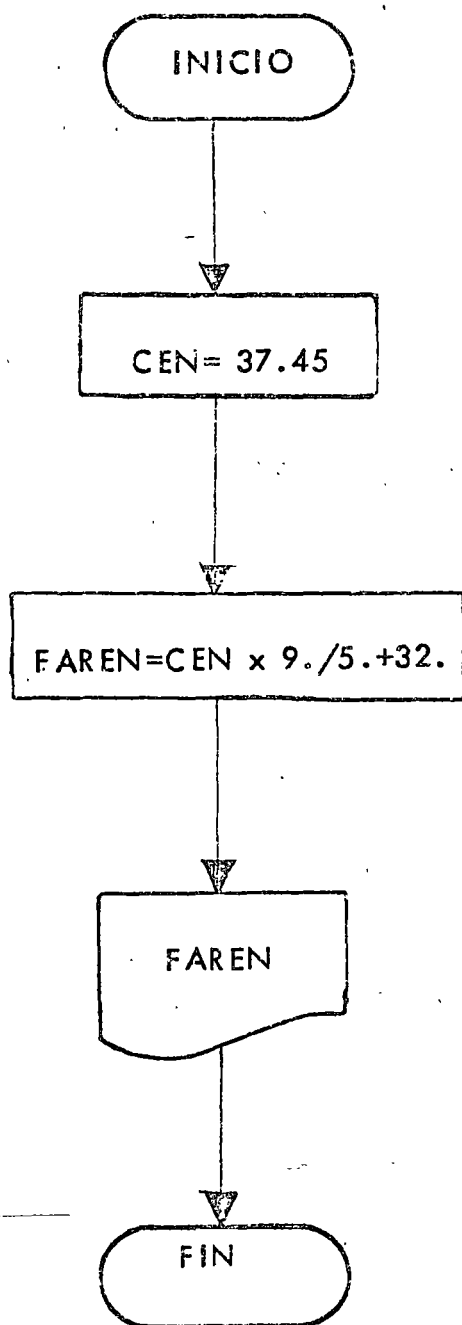
Una propiedad importante del COMMON es que si un arreglo es especificado en COMMON que dá automáticamente dimensionado, es decir, no hay que especificar dicho arreglo a través de la declaración DIMENSION .

En las siguientes páginas se muestran veintiún programas, que incluyen sus diagramas de flujo, codificaciones, datos y resultados; el objeto es que el lector pueda complementar la parte teórica con la práctica, amén de que deberá hacer los propios y procesarlos en una computadora a su alcance.

REFERENCIAS BIBLIOGRAFICAS:

- 1.- J.K. Hughes : Programación del Sistema IBM-1130
Limusa-Wiley-1969
- 2.- D.D. McCracken : Fortran IV
Limusa-Wiley-1964
- 3.- E.I. Organick : Fortran IV
Fondo Educativo Interamericano, S. A.
1966
- 4.- Francis Scheid : Introducción a la Ciencia de
las Computadoras.
Serie de Compendios Schaum.
McGraw-Hill-1970
- 5.- W.Schick y Ch. J. : Fortran para Ingeniería.
Merz, Jr. McGraw-Hill - 1972
- 6.- R.E. Smith y D.E. : Fortran, Texto Programado.
Johnson Limusa Wiley - 1971

"CONVERSION DE GRADOS CENTIGRADOS A
GRADOS FARENHEIT"

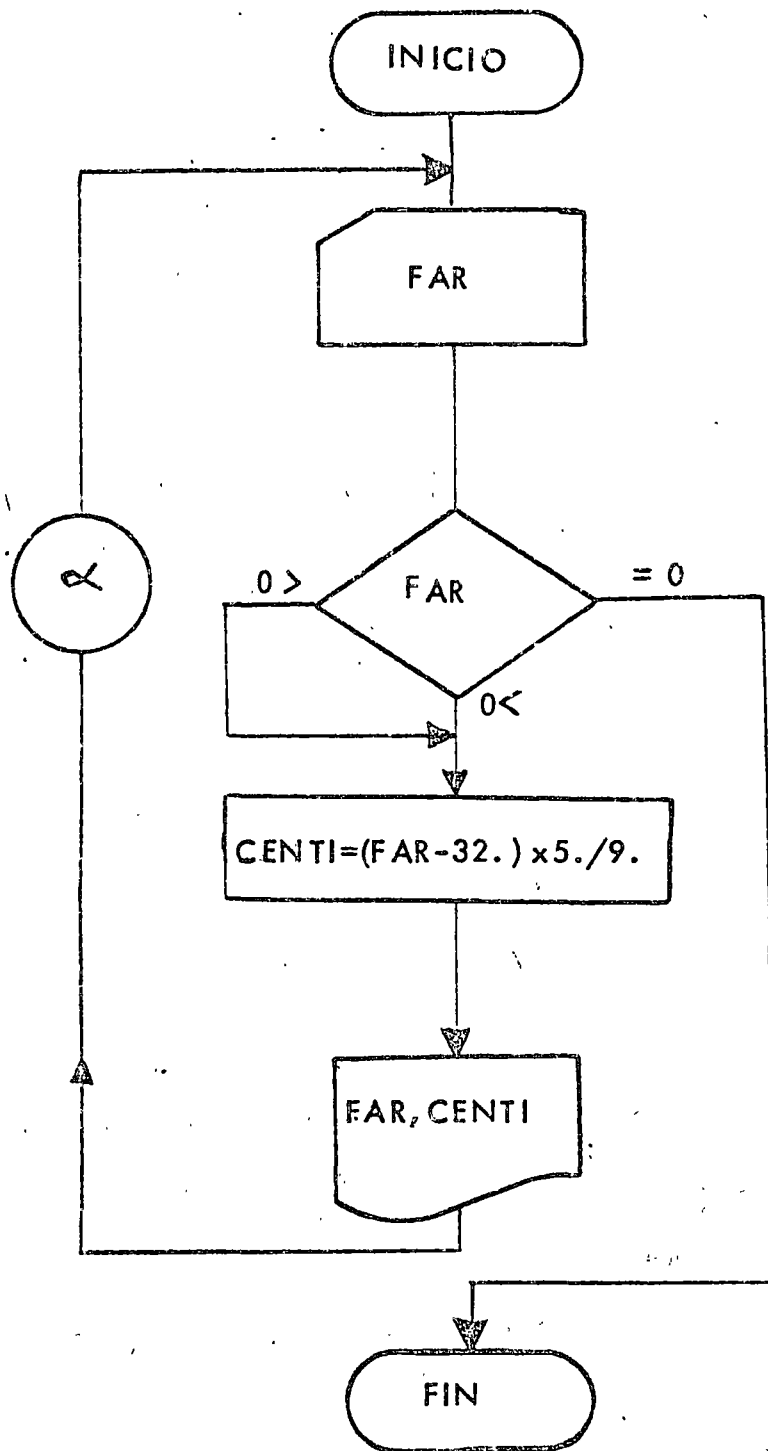


```
// JOB T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----U N O-----
C      CONVERSION DE GRADOS CENTIGRADOS A
C      GRADOS FARENHEIT
  100 FORMAT(F10.4)
      IMP=3
      CEN=37.45
      FAREN=CEN*9./5.+32.
      WRITE(IMP,100)FAREN
      CALL EXIT
      END
// XEQ
/*
```

RESULTADOS

99.4100

"CONVERSION DE GRADOS FARENHET A GRADOS CENTIGRADOS"



```

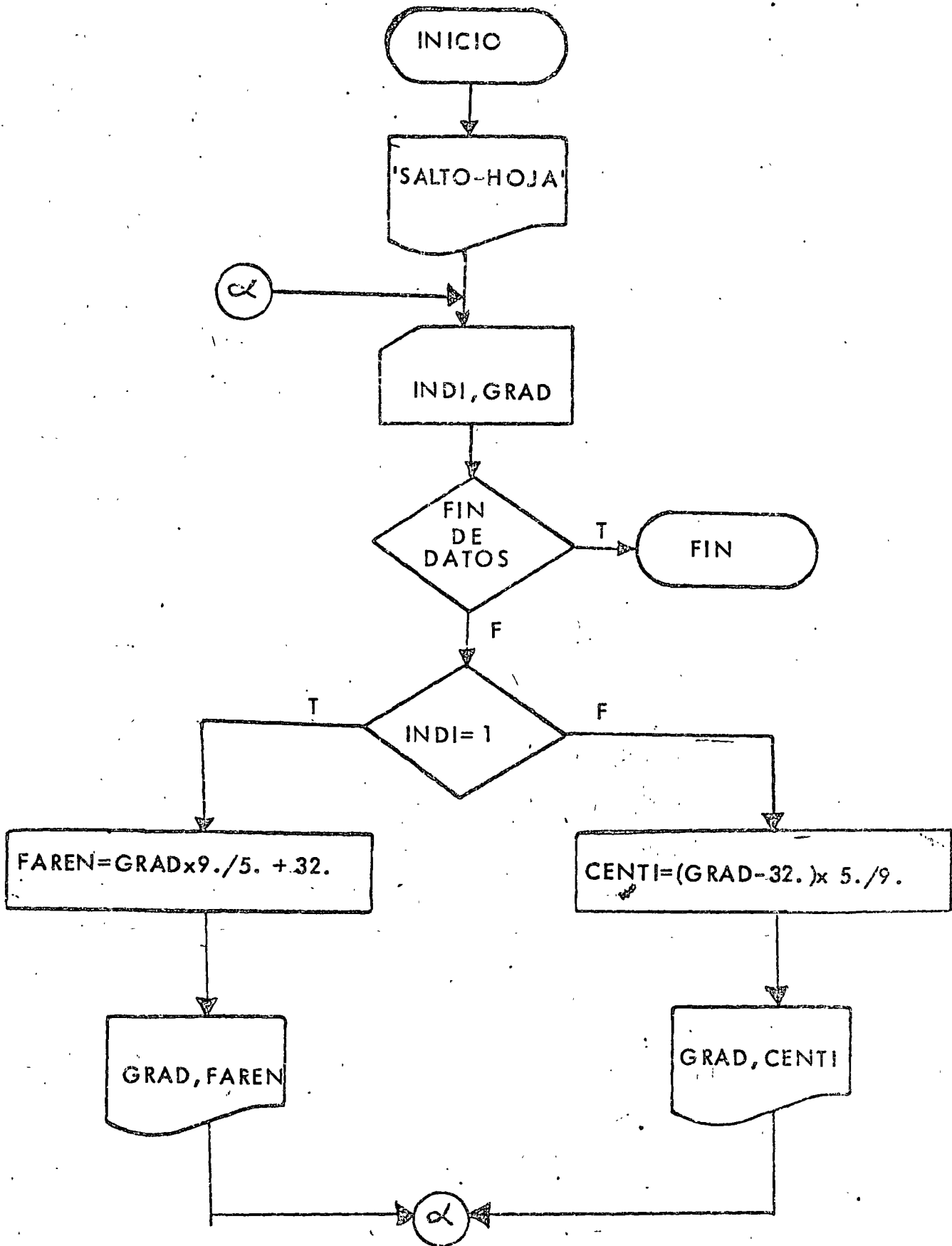
// JOB T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----D O S-----
C      CONVERSION DE GRADOS FARENHEIT A
C      GRADOS CENTIGRADOS
100 FORMAT(F10.4)
101 FORMAT(F10.4,22H GRADOS FARENHEIT SON ,F10.4,20H GRADOS CENTIGRADO
1S.)
LEE=2
IMP=3
200 READ(LEE,100)FAR
C      FAR IGUAL CERO INDICA TERMINO DE DATOS.
IF(FAR)210,220,210
210   CENTI=(FAR-32.)*5./9.
      WRITE(IMP,101)FAR,CENTI
      GO TO 200
220 CALL EXIT
END
// XEQ
1260000
126.
-14.
18.26
0.0
/*

```

RESULTADOS

126.0000 GRADOS FARENHEIT SON	52.2222 GRADOS CENTIGRADOS.
126.0000 GRADOS FARENHEIT SON	52.2222 GRADOS CENTIGRADOS.
-14.0000 GRADOS FARENHEIT SON	-25.5556 GRADOS CENTIGRADOS.
18.2600 GRADOS FARENHEIT SON	-7.6333 GRADOS CENTIGRADOS.

"CONVERSION ENTRE GRADOS FARENHEIT Y GRADOS CENTIGRAOS"



```

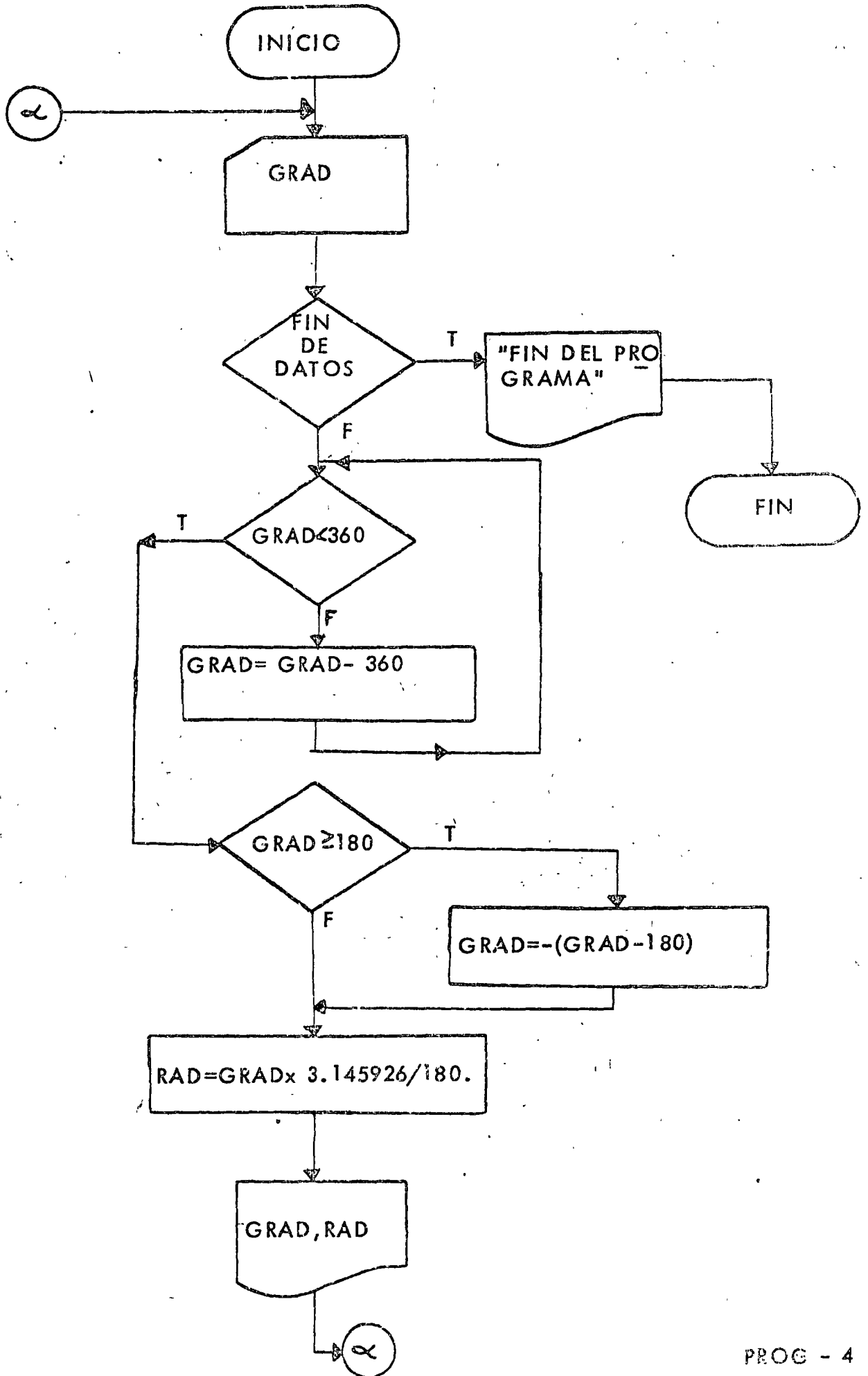
// JOB T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----T R E S-----
C      CONVERSION ENTRE GRADOS FARENHEIT
C      Y GRADOS CENTIGRADOS
  100 FORMAT(1H1)
  101 FORMAT(I1,F10.3)
  102 FORMAT(F10.2,15H FARENHEIT SON ,F11.3,13H CENTIGRADOS.)
  103 FORMAT(F10.2,17H CENTIGRADOS SON ,F9.3,11H FARENHEIT.)
    LEE=2
    IMP=3
    WRITE(IMP,100)
  200 READ(LEE,101,END=220)INDI,GRAD
    IF(INDI.EQ.1)GO TO 210
C      INDI DIFERENTE DE 1 DATO EN GRADOS FARENHEIT.
C      SE CONVIERTE A CENTIGRADOS.
    CENTI=(GRAD-32.)*5./9.
    WRITE(IMP,102)GRAD,CENTI
    GO TO 200
  210 CONTINUE
C      EL DATO ES EN GRADO CENTIGRADO.
C      SE CONVIERTE A FARENHEIT.
    FAREN=GRAD*9./5.+32.
    WRITE(IMP,103)GRAD,FAREN
    GO TO 200
  220 CALL EXIT
    END
// XEQ
1      12000
0      11.48
1      0.
2      32.00
1      -16.
1      18.
/*

```

R E S U L T A D O S

12.00	CENTIGRADOS SON	53.600	FARENHEIT.
11.48	FARENHEIT SON	-11.400	CENTIGRADOS.
0.00	CENTIGRADOS SON	32.000	FARENHEIT.
32.00	FARENHEIT SON	0.000	CENTIGRADOS.
-16.00	CENTIGRADOS SON	3.200	FARENHEIT.
18.00	CENTIGRADOS SON	64.400	FARENHEIT.

"CONVERSION DE GRADOS A RADIANES"




```

// JOB T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----C U A T R O-----
C   CONVERSION DE GRADOS A RADIANES
  101 FORMAT(F8.3)
  102 FORMAT(F9.3,12H GRADOS SON ,F6.3,10H RADIANES.)
  103 FORMAT(///,10X,16HFIN DEL PROGRAMA)
    LEE=2
    IMP=3
  200 READ(LEE,101,END=230)GRAD
  210 IF(GRAD.LT.360)GO TO 220
C   EL DATO ES IGUAL O SOBREPASA LOS 360 GRADOS(SE AJUSTA).
    GRAD=GRAD-360.
    GO TO 210
  220 CONTINUE
C   SE TRABAJA ENTRE +180 Y -180 GRADOS.
    IF(GRAD.GE.180.)GRAD=- (GRAD-180.)
    RAD=GRAD*3.1415926/180.
    WRITE(IMP,102)GRAD,RAD
    GO TO 200
  230 WRITE(IMP,103)
    CALL EXIT
    END

```

```

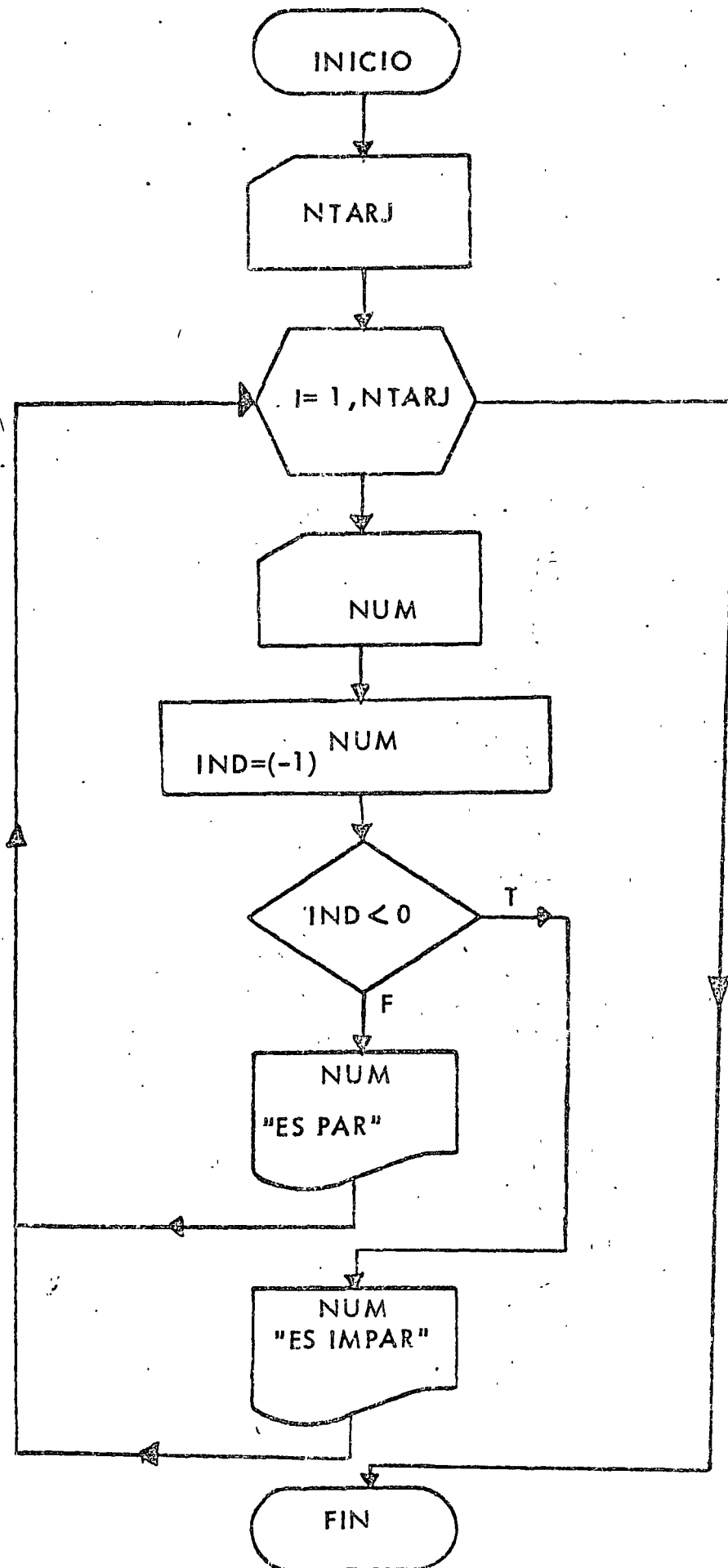
// XEQ
  90000
-90.
  3600.
 -380.
   0.0
  185.27
  132.4
 -79.9
/*

```

RESULTADOS

90.000	GRADOS	SON	1.571	RADIANES.
-90.000	GRADOS	SON	-1.571	RADIANES.
0.000	GRADOS	SON	0.000	RADIANES.
-380.000	GRADOS	SON	-6.632	RADIANES.
0.000	GRADOS	SON	0.000	RADIANES.
185.270	GRADOS	SON	3.236	RADIANES.
132.400	GRADOS	SON	2.311	RADIANES.
-79.900	GRADOS	SON	-1.395	RADIANES.
FIN DEL PROGRAMA				

"DETERMINACION DE NUMEROS PARES E IMPARES"



```

// JOB T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----C I N C O-----
C   DETERMINACION DE NUMEROS PARES
C   E IMPARES
100 FORMAT(I3)
101 FORMAT(I4,8H ES PAR.)
102 FORMAT(I4,10H ES IMPAR.)
    LEE=2
    IMP=3
    READ(LEE,100)NTARJ
C   NTARJ INDICA NO. DE TARJETAS CON DATOS.
    DO 202 I=1,NTARJ
      READ(LEE,100)NUM
      IND=(-1)**NUM
      IF(IND.LT.0)GO TO 200
C     EL NUMERO ES PAR.
      WRITE(IMP,101)NUM
      GO TO 201
200  CONTINUE
C     EL NUMERO ES IMPAR.
      WRITE(IMP,102)NUM
201  CONTINUE
202  CONTINUE
    CALL EXIT
    END

```

```

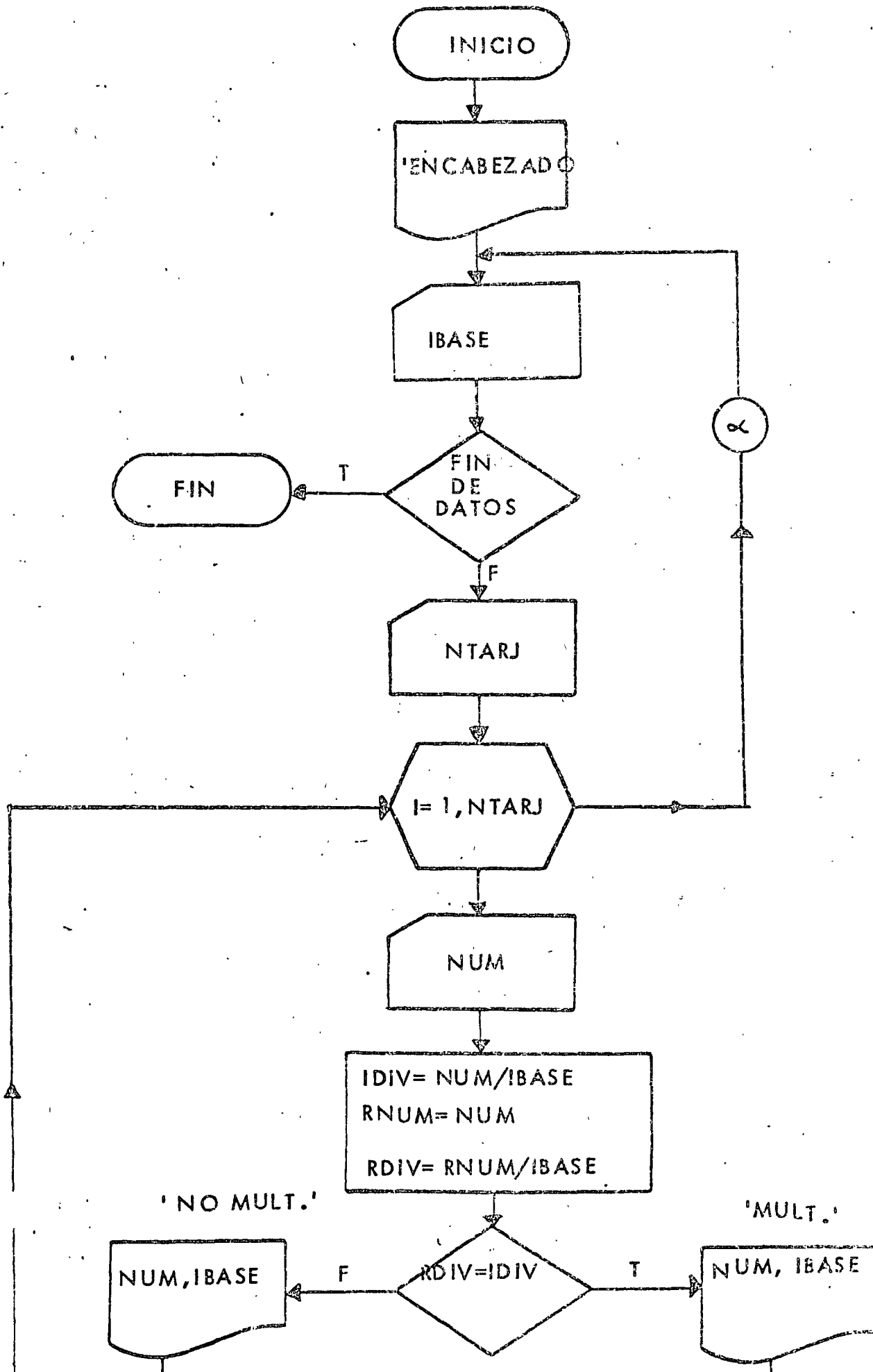
// XEQ
005
17
1
14
291
8
/*

```

RESULTADOS

17	ES	IMPAR.
1	ES	IMPAR.
14	ES	PAR.
291	ES	IMPAR.
8	ES	PAR.

"DETERMINACION DE MULTIPLOS DE UN NUMERO"



```

// JOB T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)

```

```

C-----S E I S-----
C   DETERMINACION DE MULTIPLOS
C   DE UN NUMERO
 100 FORMAT(I3)
 101 FORMAT(34H1NUMERO=MULTIPLO DE=NO MULTIPLO DE)
 102 FORMAT(2X,I3,7X,I3)
 103 FORMAT(2X,I3,21X,I3)
    LEE=2
    IMP=3
    WRITE(IMP,101)
 200 READ(LEE,100,END=240)IBASE
    READ(LEE,100)NTARJ
    DO 230 I=1,NTARJ
      READ(LEE,100)NUM
      IDIV=NUM/IBASE
      RNUM=NUM
      RDIV=RNUM/IBASE
      IF(RDIV.EQ.IDIV)GO TO 210
C   NUM NO ES MULTIPLO DE IBASE.
      WRITE(IMP,103)NUM,IBASE
      GO TO 220
 210 CONTINUE
C   NUM SI ES MULTIPLO DE IBASE.
      WRITE(IMP,102)NUM,IBASE
 220 CONTINUE
 230 CONTINUE
      GO TO 200
 240 CALL EXIT
    END

```

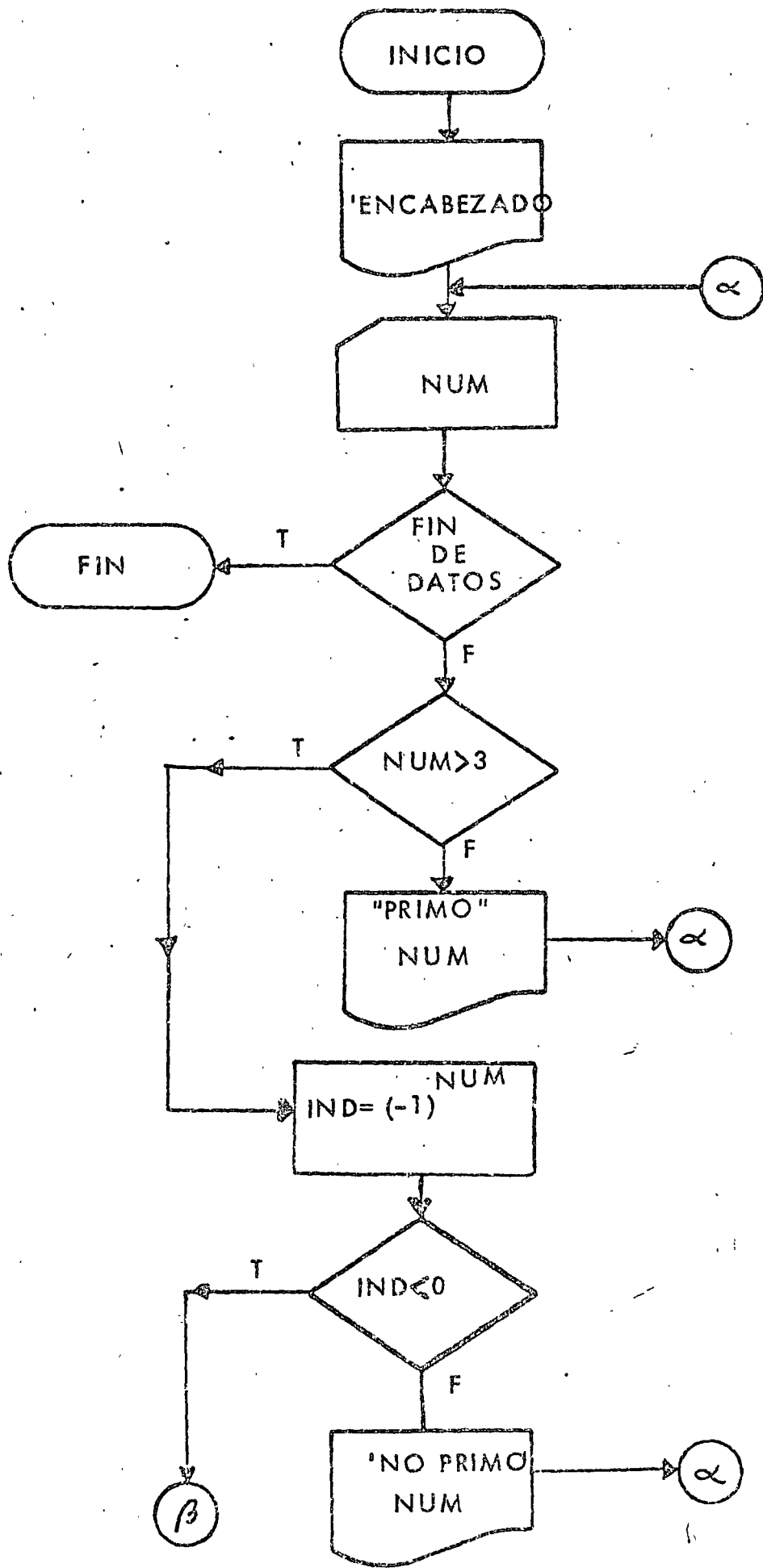
```

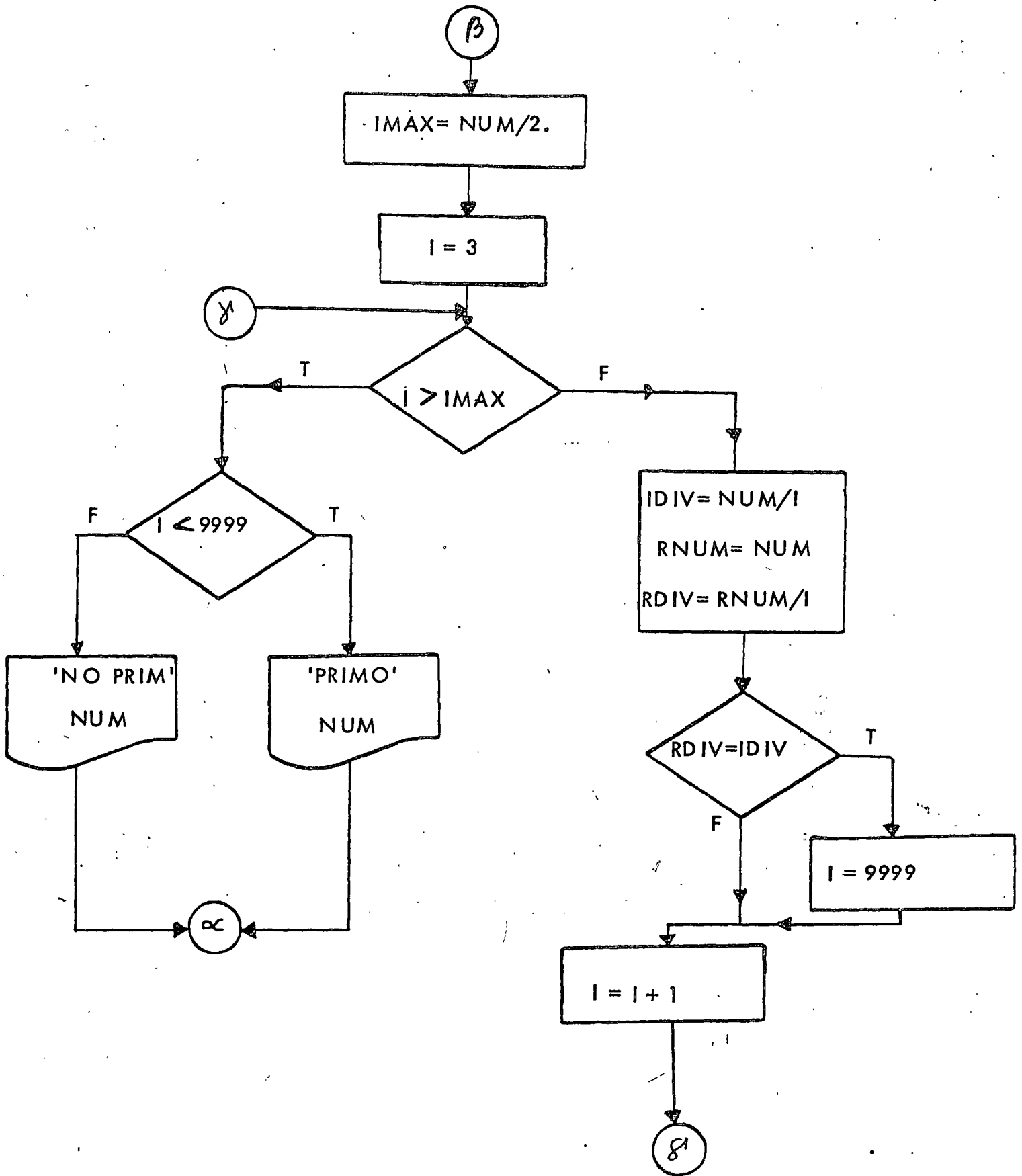
// XEQ
002
005
 17
001
 14
291
 8
003
002
 9
 11
005
001
009
/*

```

RESULTADOS

NUMERO=MULTIPLO DE=NO MULTIPLO DE		
17		2
14	2	2
291		2
8	2	
9	5	
11		3
9		5





```

// JOB T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----S I E T E-----
C   NUMEROS PRIMOS
C   100 FORMAT(I3)
C   101 FORMAT(19H1PRIMOS -NO PRIMOS)
C   102 FORMAT(2X,I3)
C   103 FORMAT(14X,I3)
C   LEE=2
C   IMP=3
C   WRITE(IMP,101)
C 200 READ(LEE,100,END=290)NUM
C   IF(NUM.GT.3)GO TO 210
C   NUM ES MENOR O IGUAL A 3(TODO NUMERO NATURAL MENOR O IGUAL A 3 ES
C   WRITE(IMP,102)NUM
C   GO TO 280
C 210 CONTINUE
C   NUM ES MAYOR QUE 3.
C   IND=(-1)*NUM
C   IF(IND.LT.0)GO TO 220
C   NUM ES PAR(TODO NUMERO PAR MAYOR QUE 3 NO ES PRIMO.).
C   WRITE(IMP,103)NUM
C   GO TO 270
C 220 CONTINUE
C   NUM ES IMPAR(SE INICIA PROCESO DE PRIMO O NO=PRIMO).
C   IMAX=NUM/2
C   I=3
C 230 IF(I.GT.IMAX)GO TO 240
C   SE OBTIENEN LAS DIVISIONES ENTERA Y REAL DE NUM/I
C   CON I DE 3 HASTA NUM/2.
C   IDIV=NUM/I
C   RNUM=NUM
C   RDIV=RNUM/I
C   IF(RDIV.EQ.IDIV)I=9999
C   I=9999 INDICA QUE NUM ES PRIMO.
C   I=I+1
C   GO TO 230
C 240 CONTINUE
C   IF(I.LT.9999)GO TO 250
C   NUM NO ES PRIMO.
C   WRITE(IMP,103)NUM
C   GO TO 260
C 250 CONTINUE
C   NUM ES PRIMO.
C   WRITE(IMP,102)NUM
C 260 CONTINUE
C 270 CONTINUE
C 280 CONTINUE
C   GO TO 200
C 290 CALL EXIT
C   END

```

```

// XEQ

```

```

1
2
3
4
5
6
7
8

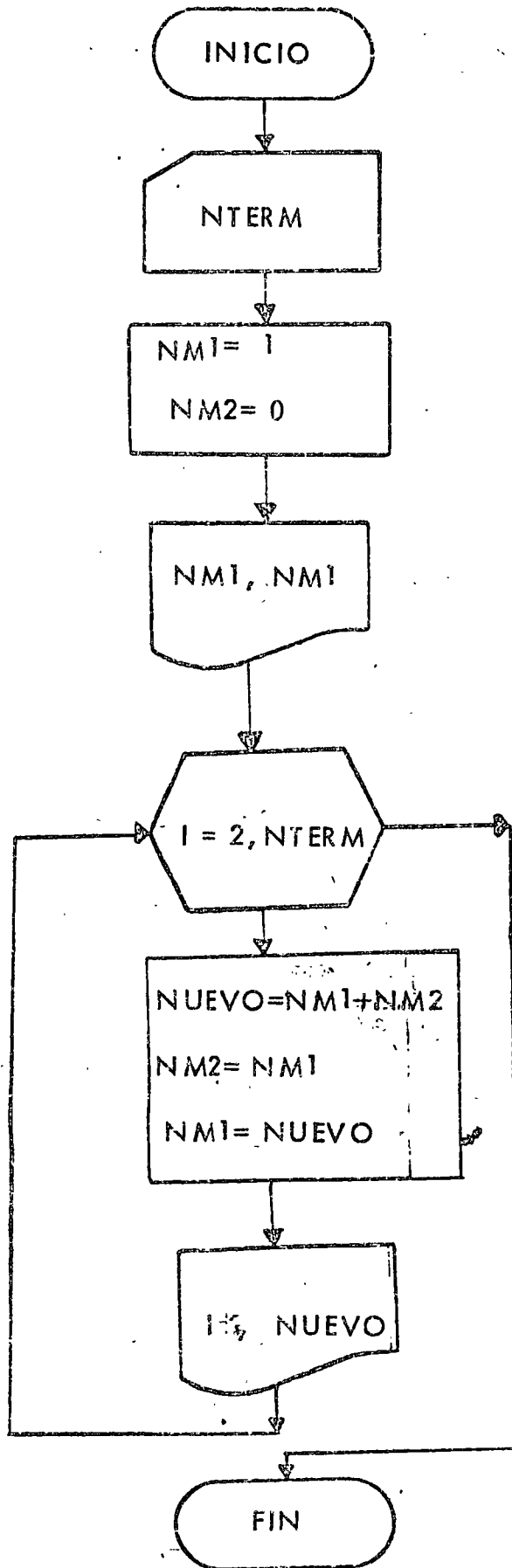
```


9
10
11
12
13
15
17
19
21
/*

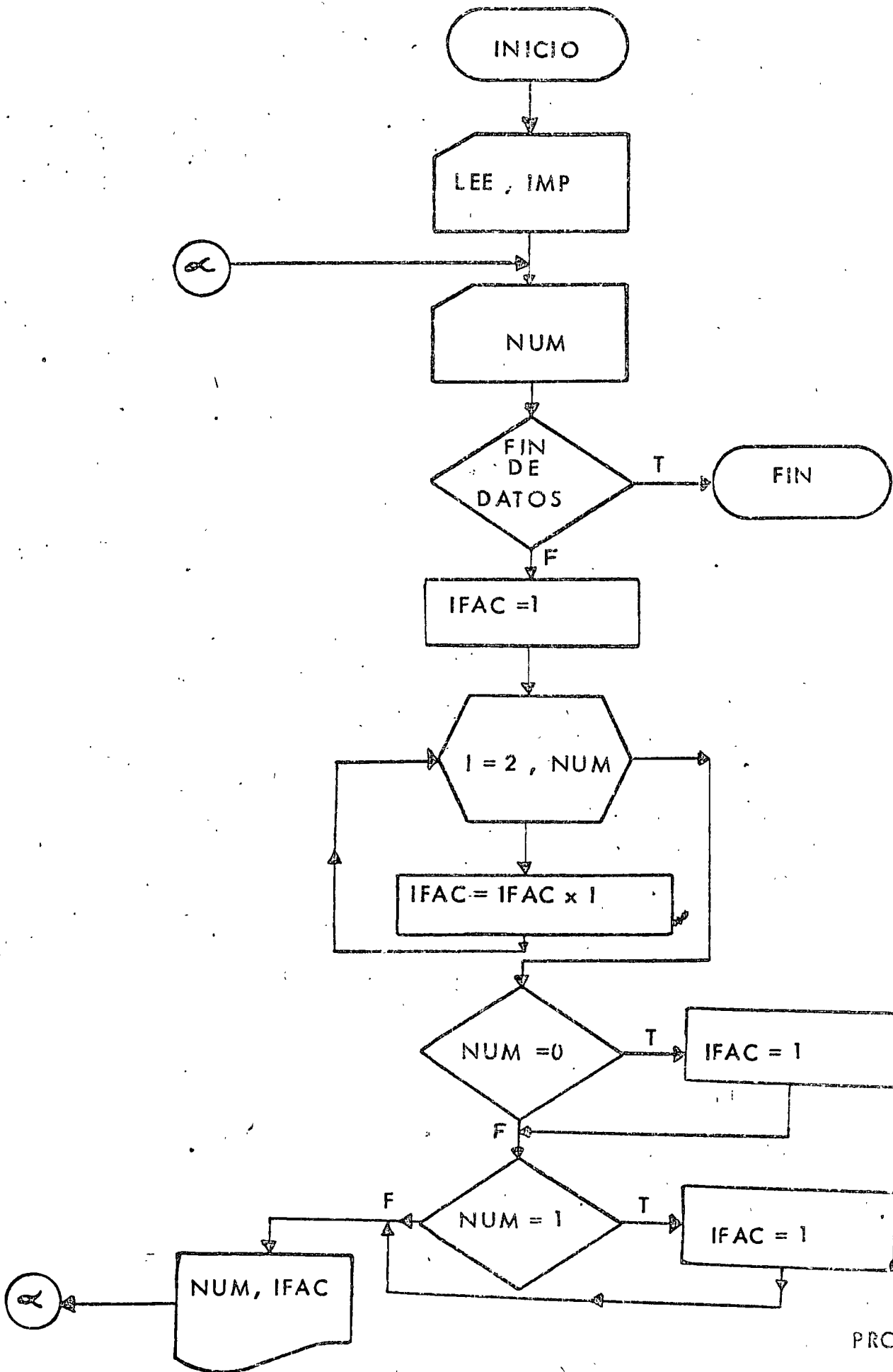
RESULTADOS

PRIMOS	NO PRIMOS
1	
2	
3	
5	4
7	6
	8
	9
	10
11	12
13	15
17	
19	21

" SERIE DE FIBONACCI "



" FACTORIAL "



```

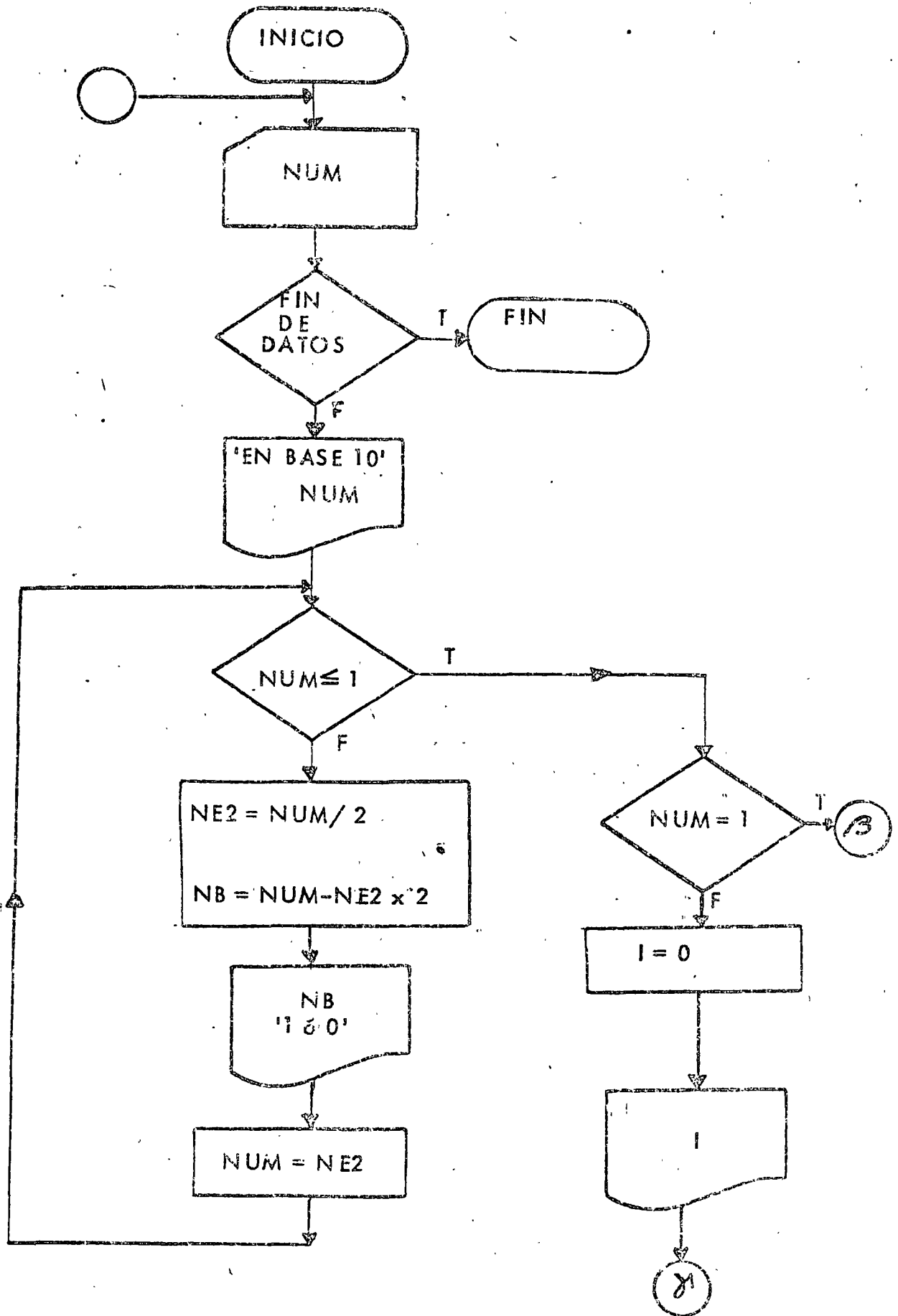
// JOB T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----N U E V E-----
C   FACTORIAL
   100 FORMAT(2I1)
   101 FORMAT(I2)
   102 FORMAT(I3,3X,I5)
C   SE LEEN LAS UNIDADES LOGICAS DE LECTURA E IMPRESIONN
   READ(2,100)LEE,IMP
   200 READ(L EE,101,END=220)NUM
      IFAC=1
      DO 210 I=2,NUM
         IFAC=IFAC*I
   210 CONTINUE
      IF(NUM.EQ.0)IFAC=1
      IF(NUM.EQ.1)IFAC=1
C   SE IMPRIME EL NUMERO Y SU FACTORIAL
   WRITE(IMP,102)NUM,IFAC
   GO TO 200
   220 CALL EXIT
      END

// XEQ
23
01
02
03
04
05
00
/*

```

RESULTADOS

1	1
2	2
3	6
4	24
5	120
0	1



```

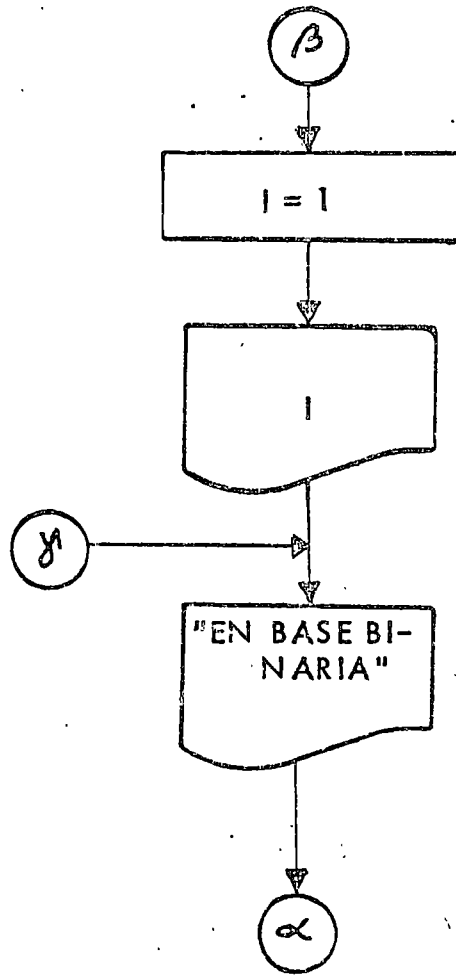
// JOB T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----D I E Z-----
C   CAMBIO DE BASE : DECIMAL A BINARIA
100 FORMAT(I5)
101 FORMAT(IX,I5,19H EN BASE DECIMAL ES)
102 FORMAT(IX,I1)
103 FORMAT(17H EN BASE BINARIA.)
   LEE=2
   IMP=3
200 READ(LEE,100,END=250)NUM
   WRITE(IMP,101)NUM
210 IF(NUM.LE.1)GO TO 220
C   NUM ES MAYOR QUE UNO. SE SIGUE DESCOMPONRIENDO
   NE2=NUM/2
   NB=NUM-NE2*2
C   NB ES UNO O CERO
   WRITE(IMP,102)NB
   NUM=NE2
   GO TO 210
220 CONTINUE
   IF(NUM.EQ.1)GO TO 230
C   SE IMPRIME EL ULTIMO CERO EN LA REPRESENTACION BINARIA
   I=0
   WRITE(IMP,102)I
   GO TO 240
230 CONTINUE
C   SE IMPRIME EL ULTIMO 1 EN LA REPRESENTACION BINARIA
   I=1
   WRITE(IMP,102)I
240 CONTINUE
   WRITE(IMP,103)
   GO TO 200
250 CALL EXIT
   END

```

```

// XEQ
12
163
10
00001
011
-131
0
1
13
/*

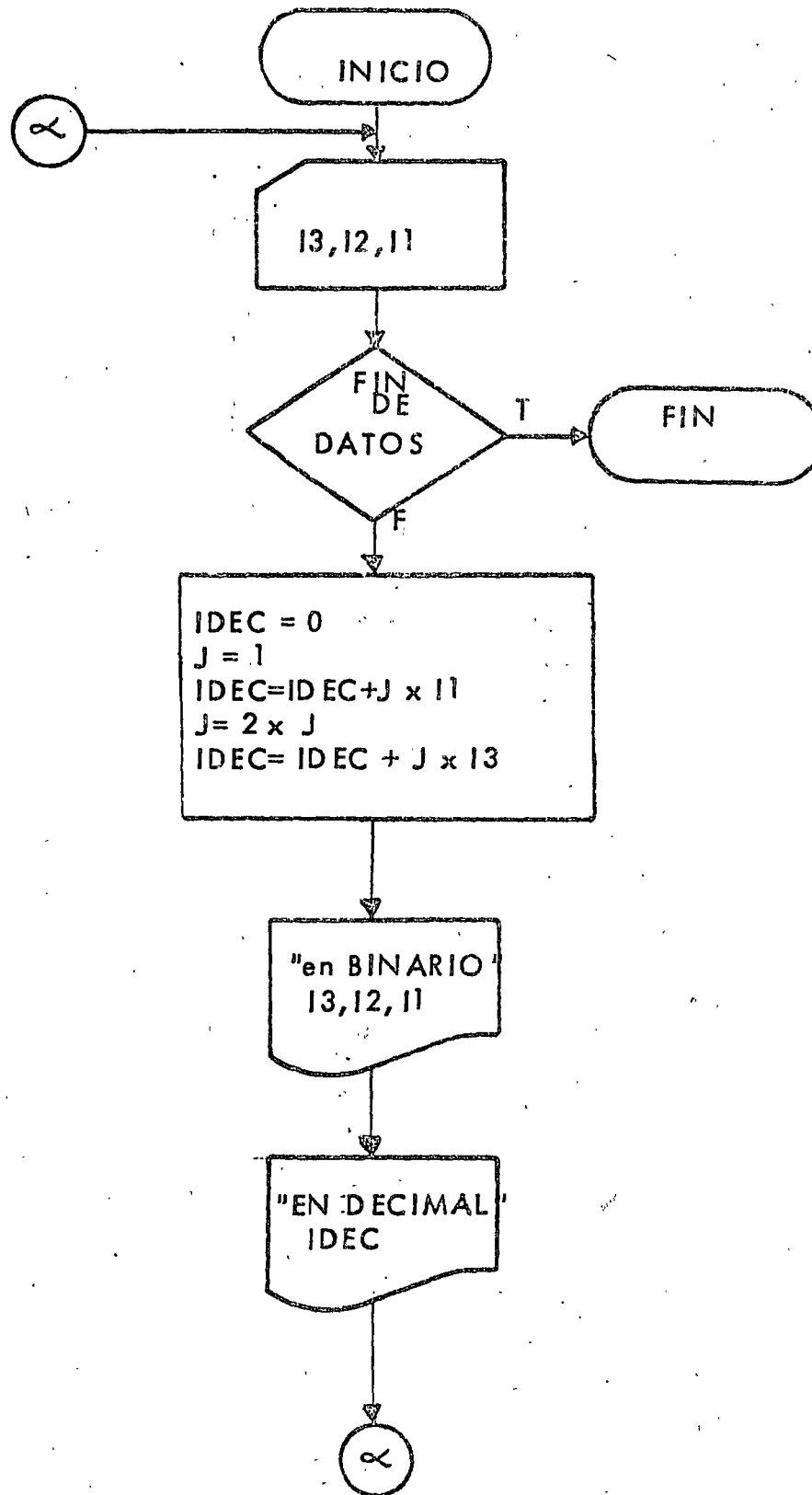
```



RESULTADOS

0	12 EN BASE DECIMAL ES
0	
1	
1	EN BASE BINARIA.
1	163 EN BASE DECIMAL ES
1	
0	
0	
0	
1	
1	EN BASE BINARIA.
0	10 EN BASE DECIMAL ES
0	
1	
1	EN BASE BINARIA.
1	1 EN BASE DECIMAL ES
1	EN BASE BINARIA.
1	11 EN BASE DECIMAL ES
1	
0	
1	
1	EN BASE BINARIA.
1	131 EN BASE DECIMAL ES
0	
0	EN BASE BINARIA.
0	0 EN BASE DECIMAL ES
0	
0	EN BASE BINARIA.
1	1 EN BASE DECIMAL ES
1	
1	EN BASE BINARIA.
1	13 EN BASE DECIMAL ES
1	
0	
1	
1	EN BASE BINARIA.

" CAMBIO DE BASE , BINARIO A DECIMAL "



```

// JOB Y
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----0 N C E-----
C   CAMBIO DE BASE ; BINARIA A DECIMAL
  100 FORMAT(3I1)
  101 FORMAT(1X, 3I1,19H EN BASE BINARIA ES)
  102 FORMAT(1X,15,17H EN BASE DECIMAL.)
      LEE=2
      IMP=3
  200 READ(2,100,END=210)I3,I2,I1
      IDEC=0
C   J CONTIENE LAS POTENCIAS DE 2.
      J=1
      IDEC=IDEC+J*I1
      J=2*J
      IDEC=IDEC+J*I2
      J=2*J
      IDEC=IDEC+J*I3
      WRITE(IMP,101)I3,I2,I1
C   IDEC CONTIENE LA REPRESENTACION DECIMAL DEL NUMERO BINARIO.
      WRITE(IMP,102)IDEC
      GO TO 200
  210 CALL EXIT
      END

```

```

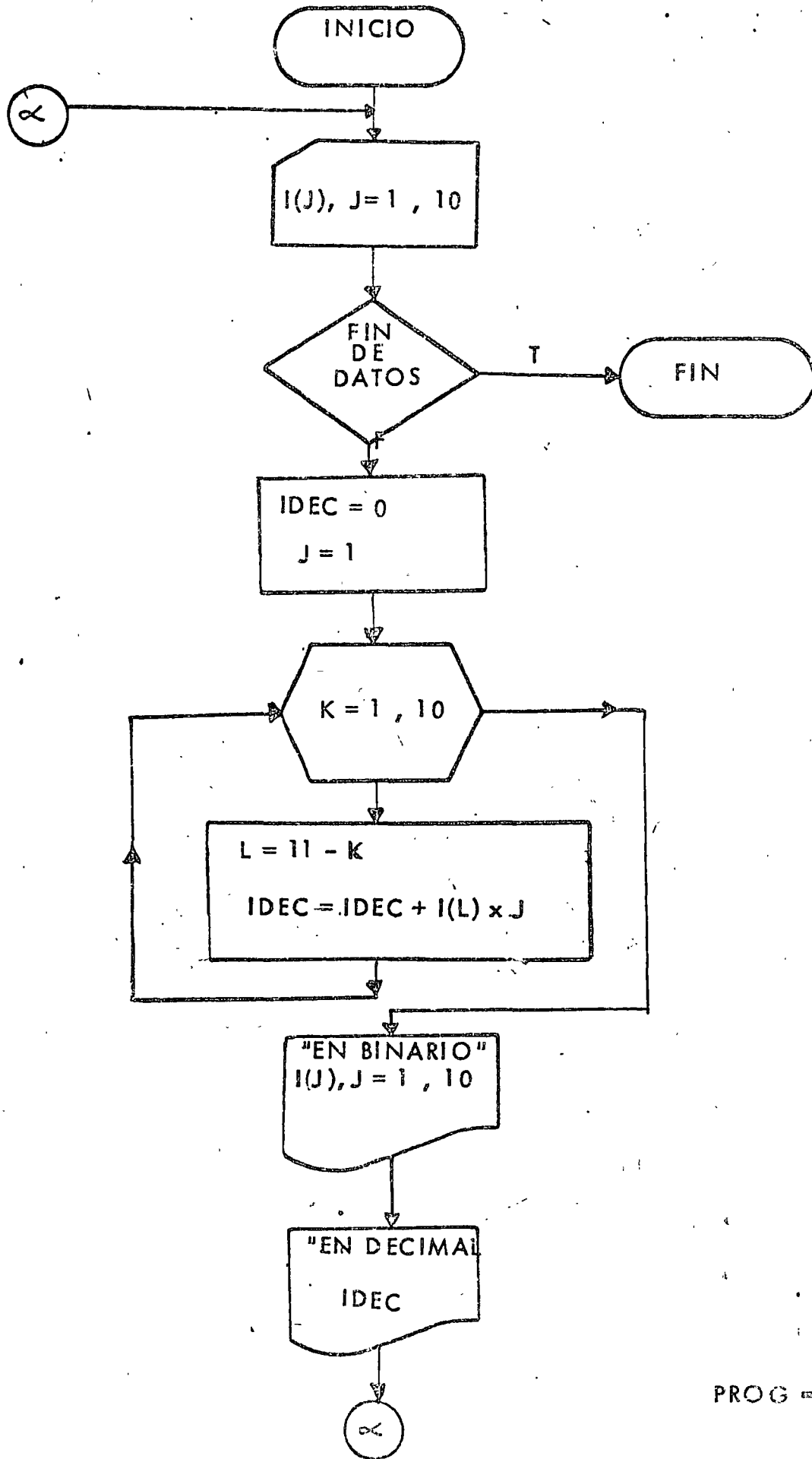
// XEQ
001
010
011
100
101
110
111
/*

```

RESULTADOS

001	EN BASE BINARIA ES
1	EN BASE DECIMAL.
010	EN BASE BINARIA ES
2	EN BASE DECIMAL.
011	EN BASE BINARIA ES
3	EN BASE DECIMAL.
100	EN BASE BINARIA ES
4	EN BASE DECIMAL.
101	EN BASE BINARIA ES
5	EN BASE DECIMAL.
110	EN BASE BINARIA ES
6	EN BASE DECIMAL.
111	EN BASE BINARIA ES
7	EN BASE DECIMAL.

"CAMBIO DE BASE, BINARIA A DECIMAL USANDO ARREGLOS"



```

// JOB T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----D O C E-----
C     CAMBIO DE BASE ; BINARIA A DECIMAL
C     USANDO ARREGLOS
      DIMENSION I(10)
      100 FORMAT(10I1)
      101 FORMAT(1X,10I1,19H EN BASE BINARIA ES)
      102 FORMAT(1X,15,17H EN BASE DECIMAL.)
      LEE=2
      IMP=3
      200 READ(2,100,END=220) I
          IDEC=0
          J=1
          DO 210 K=1,10
C       SE ANALIZA EL VECTOR I DE DERECHA A IZQUIERDA
          L=11-K
          IDEC=IDEC+I(L)*J
C       J CONTIENE LAS POTENCIAS DE 2
          J=J*2
      210 CONTINUE
          WRITE(IMP,101) (I(J),J=1,10)
          WRITE(IMP,102) IDEC
          GO TO 200
      220 CALL EXIT
          END

```

```

// XEQ
      1
      11
1000100010
  1
  1
  1001001
    00

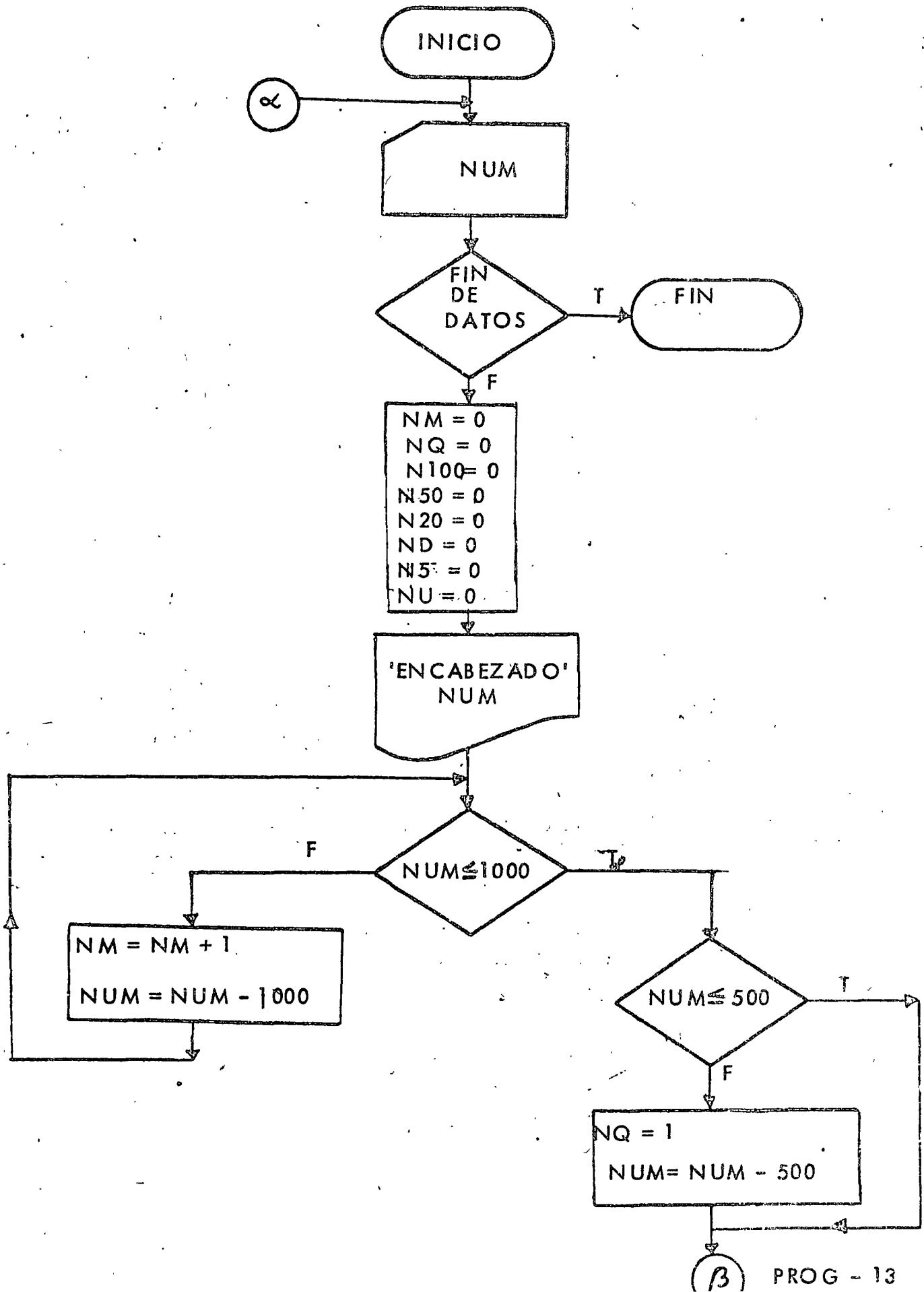
```

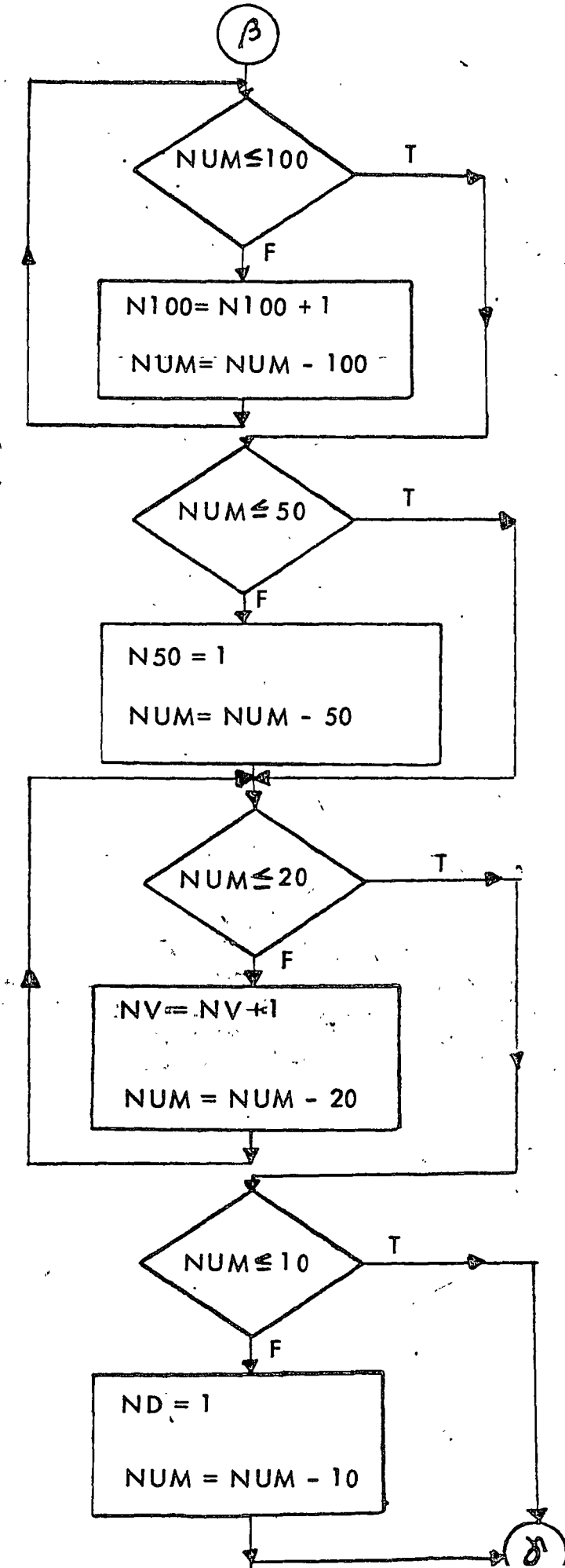
RESULTADOS

000000001	EN BASE BINARIA ES
1	EN BASE DECIMAL.
0000000110	EN BASE BINARIA ES
6	EN BASE DECIMAL.
1000100010	EN BASE BINARIA ES
546	EN BASE DECIMAL.
1000000000	EN BASE BINARIA ES
512	EN BASE DECIMAL.
1000000001	EN BASE BINARIA ES
513	EN BASE DECIMAL.
001001001	EN BASE BINARIA ES
73	EN BASE DECIMAL.
000000000	EN BASE BINARIA ES
0	EN BASE DECIMAL.

" CALCULO DEL NUMERO DE BILLETES "

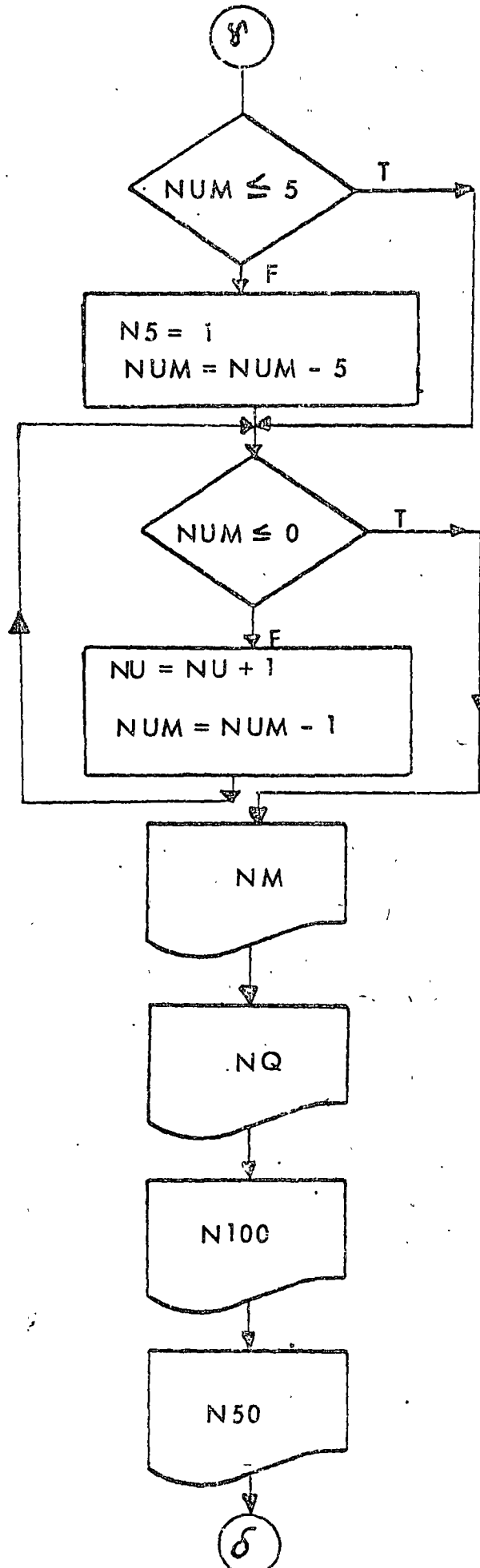
1a.

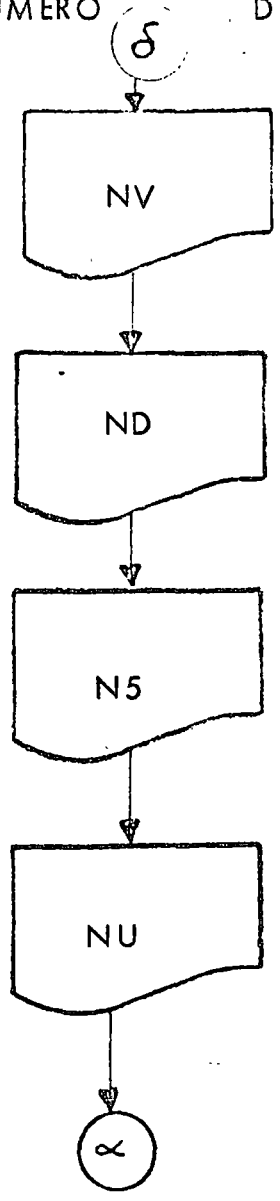




" CALCULO DEL NUMERO DE BILLETES "

3a.





```

// JOB T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----T R E C E-----
C   CALCULO DEL NUMERO DE BILLETES
  100 FORMAT(I4)
  101 FORMAT(/,11H EL NUMERO-,I4,24H.00-PUEDE DESGLOSARSE EN)
  102 FORMAT(I5, 7H DE MIL)
  103 FORMAT(I5,14H DE QUINIENTOS)
  104 FORMAT(I5, 8H DE CIEN)
  105 FORMAT(I5,13H DE CINCUENTA)
  106 FORMAT(I5,10H DE VEINTE)
  107 FORMAT(I5, 8H DE DIEZ)
  108 FORMAT(I5, 9H DE CINCO)
  109 FORMAT(I5, 7H DE UNO)
      LEE=2
      IMP=3
  200 READ(LEE,100,END=330)NUM
      NM=0
      NQ=0
      N100=0
      N50=0
      NV=0
      ND=0
      N5=0
      NU=0
      WRITE(IMP,101)NUM
  210 IF(NUM.LE.1000)GO TO 220
C     NUM ES MAYOR QUE 1000
      NM=NM+1
      NUM=NUM-1000
      GO TO 210
  220 CONTINUE
      IF(NUM.LE.500)GO TO 230
C     NUM ES MAYOR QUE 500 (MAXIMO UN BILLETE DE 500)
      NQ=1
      NUM=NUM-500
  230 CONTINUE
  240 IF(NUM.LE.100)GO TO 250
C     NUM ES MAYOR QUE 100
      N100=N100+1
      NUM=NUM-100
      GO TO 240
  250 CONTINUE
      IF(NUM.LE.50)GO TO 260
C     NUM ES MAYOR QUE 50 (MAXIMO UN BILLETE DE 50)
      N50=1
      NUM=NUM-50
  260 CONTINUE
  270 IF(NUM.LE.20)GO TO 280
C     NUM ES MAYOR QUE 20
      NV=NV+1
      NUM=NUM-20
      GO TO 270
  280 CONTINUE
      IF(NUM.LE.10)GO TO 290
C     NUM ES MAYOR QUE 10 (MAXIMO UN BILLETE DE 10)
      ND=1
      NUM=NUM-10
  290 CONTINUE
      IF(NUM.LE.5)GO TO 300

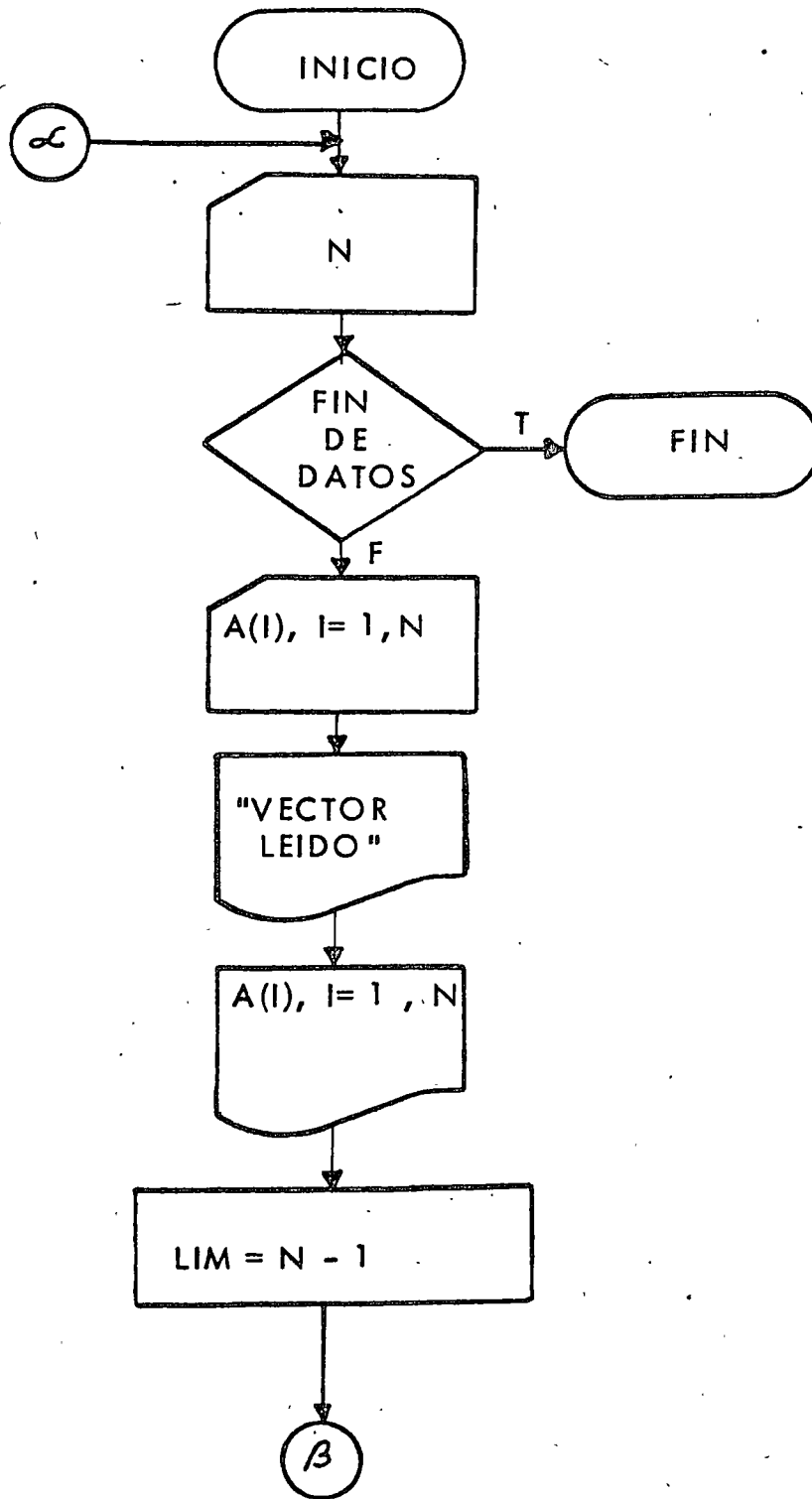
```

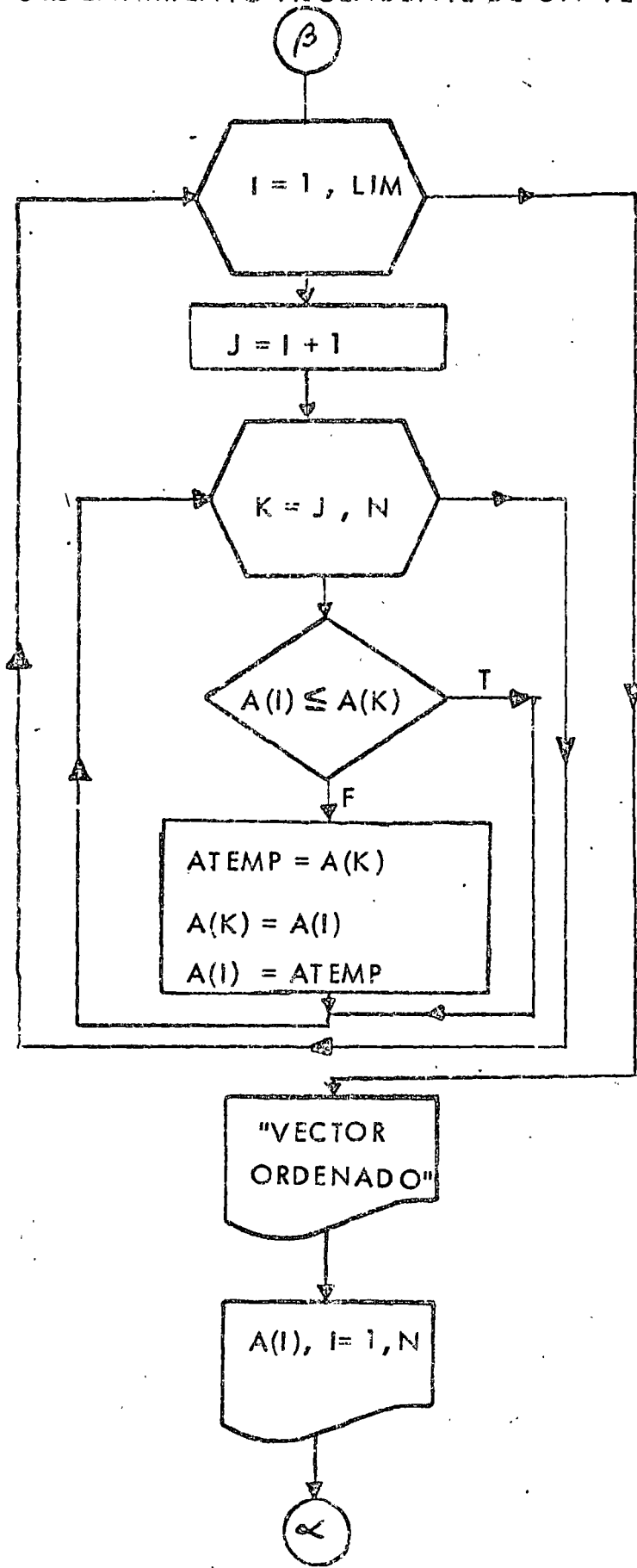
```
C      NUM ES MAYOR QUE 5 (MAXIMO UN BILLETE DE 5)
      N5=1
      NUM=NUM-5
300 CONTINUE
310 IF (NUM.LE.0) GO TO 320
C      NUM ES MAYOR QUE 1
      NU=NU+1
      NUM=NUM-1
      GO TO 310
320 CONTINUE
      WRITE (IMP,102) NM
      WRITE (IMP,103) NQ
      WRITE (IMP,104) N100
      WRITE (IMP,105) N50
      WRITE (IMP,106) NV
      WRITE (IMP,107) ND
      WRITE (IMP,108) N5
      WRITE (IMP,109) NU
      GO TO 200
330 CALL EXIT
      END

// XEQ
9000
1314
6893
1000
500
13
/*
```

RESULTADOS

EL NUMERO=9000.00=PUEDE DESGLOSARSE EN
8 DE MIL
1 DE QUINIENTOS
4 DE CIEN
1 DE CINCUENTA
2 DE VEINTE
0 DE DIEZ
1 DE CINCO
5 DE UNO
EL NUMERO=1314.00=PUEDE DESGLOSARSE EN
1 DE MIL
0 DE QUINIENTOS
3 DE CIEN
0 DE CINCUENTA
0 DE VEINTE
1 DE DIEZ
0 DE CINCO
4 DE UNO
EL NUMERO=6893.00=PUEDE DESGLOSARSE EN
6 DE MIL
8 DE QUINIENTOS
3 DE CIEN
1 DE CINCUENTA
2 DE VEINTE
0 DE DIEZ
0 DE CINCO
3 DE UNO
EL NUMERO=1000.00=PUEDE DESGLOSARSE EN
0 DE MIL
1 DE QUINIENTOS
4 DE CIEN
1 DE CINCUENTA
2 DE VEINTE
0 DE DIEZ
1 DE CINCO
5 DE UNO
EL NUMERO=500.00=PUEDE DESGLOSARSE EN
0 DE MIL
0 DE QUINIENTOS
4 DE CIEN
1 DE CINCUENTA
2 DE VEINTE
0 DE DIEZ
1 DE CINCO
5 DE UNO
EL NUMERO=13.00=PUEDE DESGLOSARSE EN
0 DE MIL
0 DE QUINIENTOS
0 DE CIEN
0 DE CINCUENTA
0 DE VEINTE
1 DE DIEZ
0 DE CINCO
3 DE UNO





```

// JOB T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----C A T O R C E---
C   ORDENAMIENTO ASCENDENTE DE UN VECTOR
   DIMENSION A(100)
   100 FORMAT(I2)
   101 FORMAT(8F10.0)
   102 FORMAT(13H VECTOR LEIDO,/)
   103 FORMAT(10(1X,F11.4))
   104 FORMAT(16H VECTOR ORDENADO,/)
   LEE=2
   IMP=3
   200 READ(LEE,100,END=240)N
C   N REPRESENTA EL NUMERO DE ELEMENTOS A ORDENAR
   READ(LEE,101)(A(I),I=1,N)
   WRITE(IMP,102)
   WRITE(IMP,103)(A(I),I=1,N)
   LIM=N-1
   DO 230 I=1,LIM
     J=I+1
C     SE ASUME QUE A(I) ES EL MENOR
     DO 220 K=J,N
       IF(A(I).LE.A(K))GO TO 210
C       A(I) FUE MAYOR QUE A(K)
       ATEMP=A(K)
       A(K)=A(I)
       A(I)=ATEMP
   210   CONTINUE
   220   CONTINUE
C   AHORA SE TIENE EN A(I) EL MENOR
   230 CONTINUE
   WRITE(IMP,104)
   WRITE(IMP,103)(A(I),I=1,N)
   GO TO 200
   240 CALL EXIT
   END

```

```

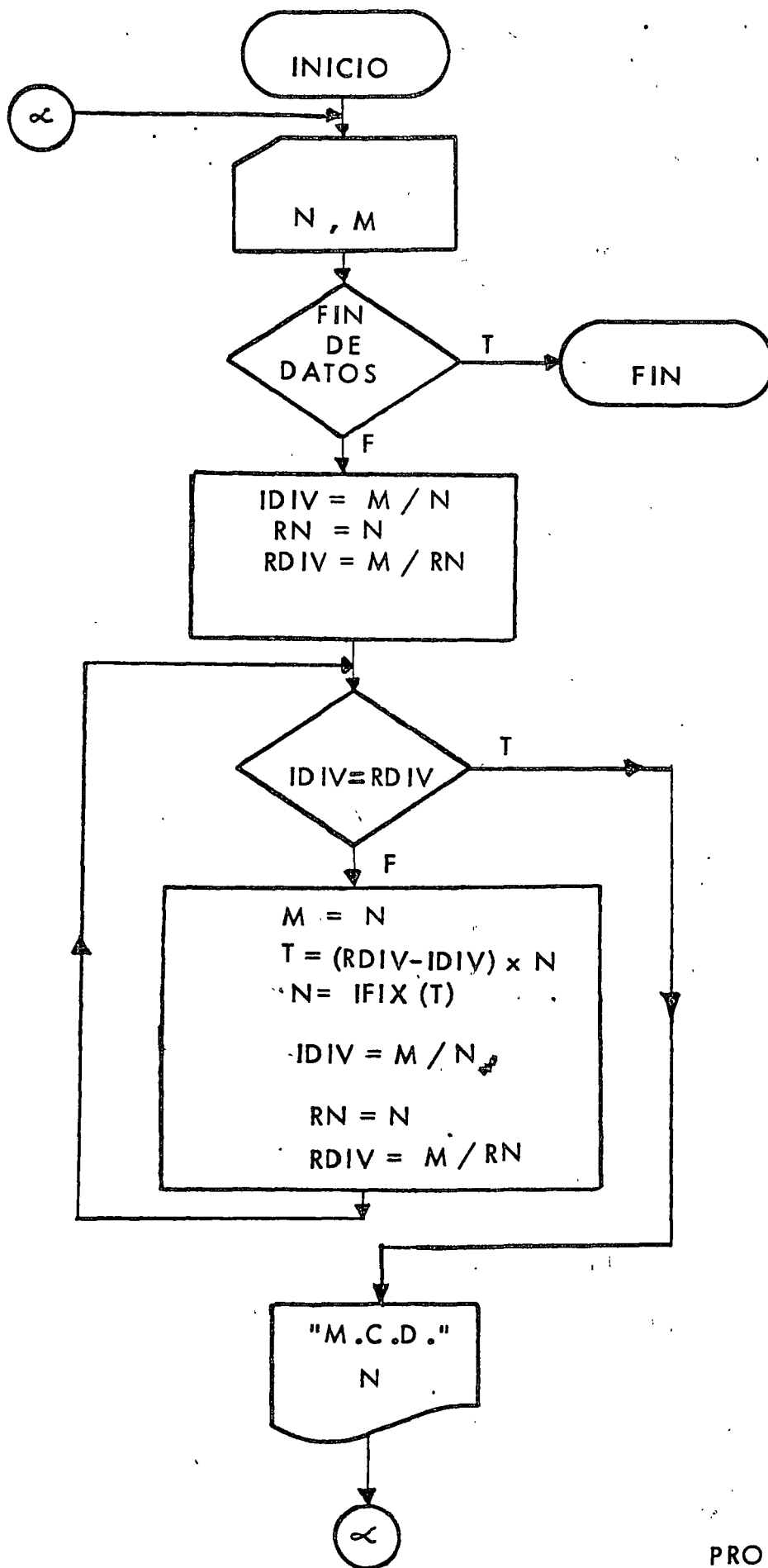
// XEQ
04
-4.      1.      -3.      17.
03
0.      -287.    32.
04
-28.    -32.     11.     0.
/*

```

RESULTADOS

VECTOR LEIDO			
-4.0000	1.0000	-3.0000	17.0000
VECTOR ORDENADO			
-4.0000	-3.0000	1.0000	17.0000
VECTOR LEIDO			
0.0000	-287.0000	32.0000	
VECTOR ORDENADO			
-287.0000	0.0000	32.0000	
VECTOR LEIDO			
-28.0000	-32.0000	11.0000	0.0000
VECTOR ORDENADO			
-32.0000	-28.0000	0.0000	11.0000

" MAXIMO COMUN MULTIPLO ALGORITMO DE EUCLIDES "



```

// JOB T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----Q U I N C E-----
C     MAXIMO COMUN MULTIPLO
C     ALGORITMO DE EUCLIDES
  100 FORMAT(2I3)
  101 FORMAT(3H N=,I3,3H M=,I3)
  102 FORMAT(8H M.C.D.=,I3,/)
     LEE=2
     IMP=3
  200 READ(LEE,100,END=230)N,M
C     SE CALCULA EL RESIDUO
     IDIV=M/N
     RN=N
     RDIV=M/RN
  210 IF(IDIV.EQ.RDIV)GO TO 220
     M=N
     T=(RDIV-IDIV)*N
     N=IFIX(T)
     IDIV=M/N
     RN=N
     RDIV=M/RN
     GO TO 210
  220 CONTINUE
C     N REPRESENTA EL MAXIMO COMUN DIVISOR
     WRITE(IMP,102)N
     GO TO 200
  230 CALL EXIT
     END

```

```

// XEQ
  5  7
  3  6
  8 16
 11 98
 16 24
  8 12
/*

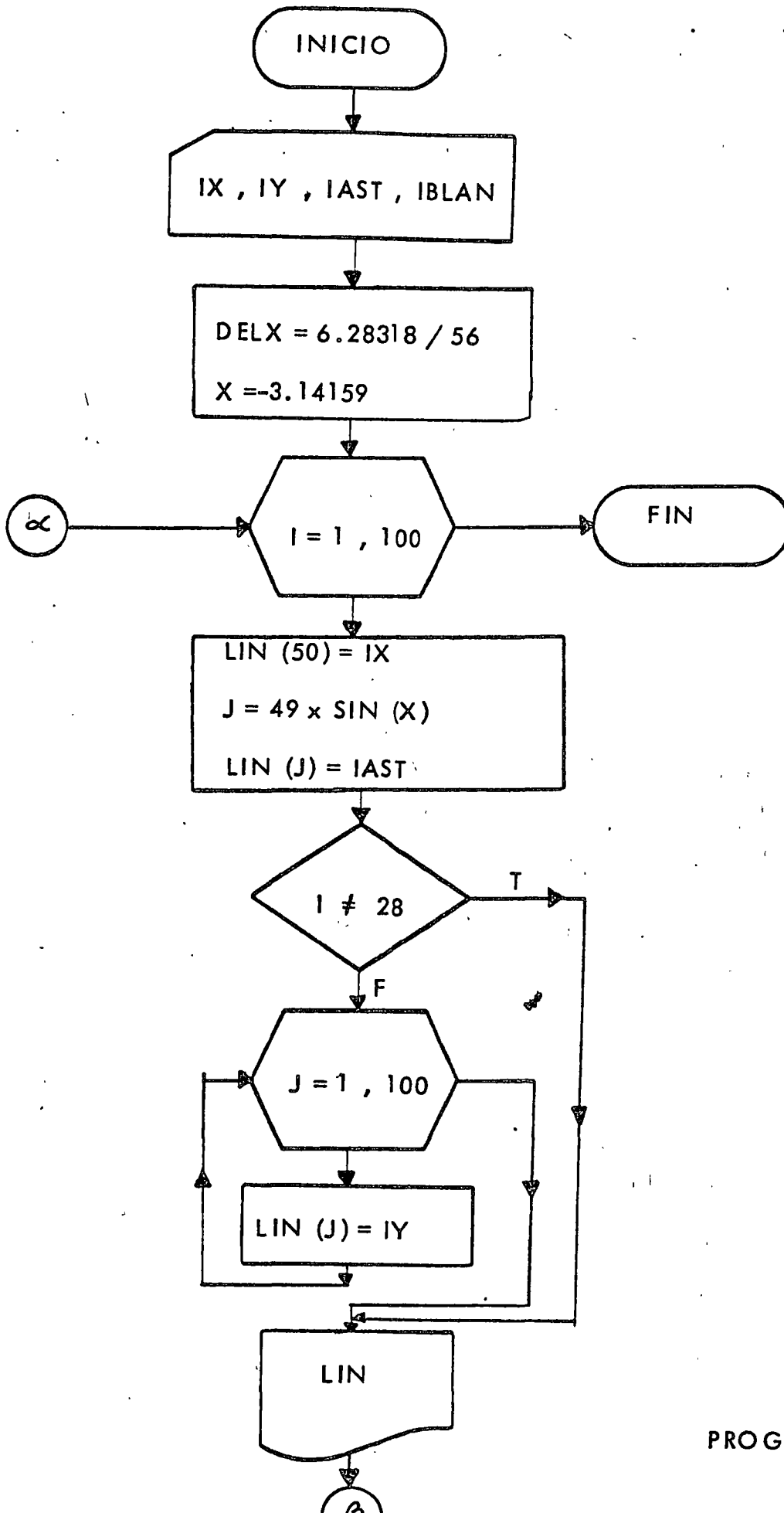
```

RESULTADOS

M.C.D.=	1
M.C.D.=	3
M.C.D.=	8
M.C.D.=	1
M.C.D.=	8
M.C.D.=	4

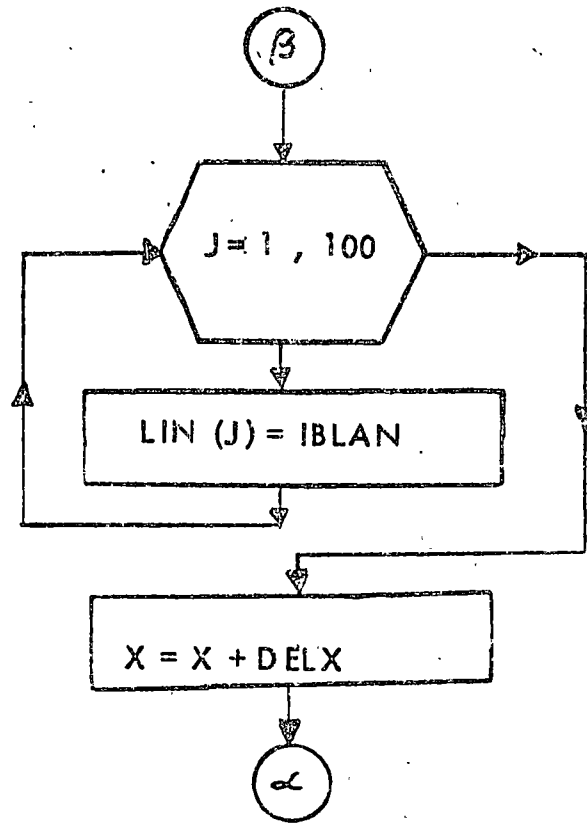
" GRAFICA DE SEN (X) "

1a.



" GRAFICA DE SEN (X) "

2a.

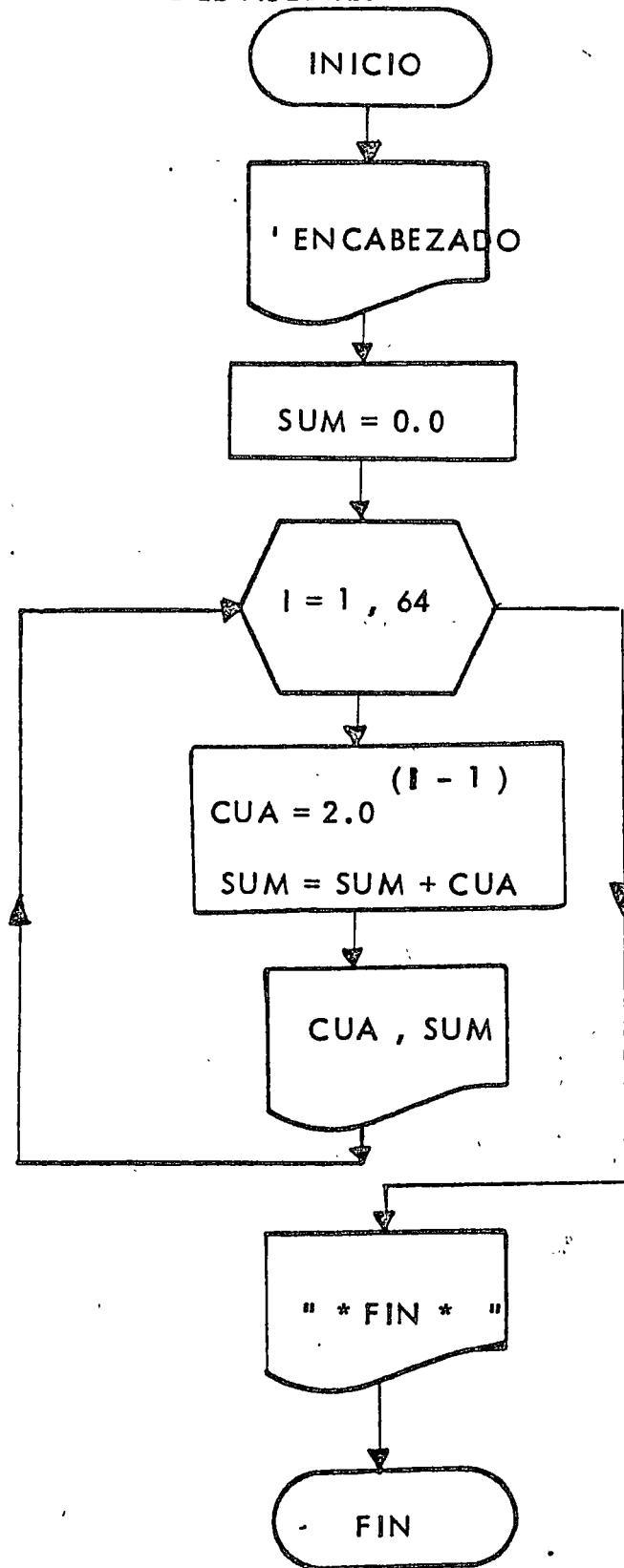


```

// JOB T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----D I E C I S E I S-----
C   GRAFICA DE SEN(X)
   DIMENSION LIN(100)
100 FORMAT(4A1)
101 FORMAT(10X,100A1)
   LEE=2
   IMP=3
   READ(LEE,100)IX,IY,IAST,IBLAN
   DELX=6.28318/56
   X =-3.14159
   DO 200 I=1,100
     LIN(I)=IBLAN
200 CONTINUE
   DO 240 I=1,56
     LIN(50)=IX
     J=49*SIN(X)+50
     LIN(J)=IAST
     IF(I.NE.28)GO TO 220
     DO 210 J=1,100
       I FUE IGUAL A 28, SE IMPRIME EL EJE Y
       LIN(J)=IY
210 CONTINUE
220 CONTINUE
     WRITE(IMP,101)LIN
     DO 230 J=1,100
       LIN(J)=IBLAN
230 CONTINUE
     X=X+DELX
240 CONTINUE
   CALL EXIT
   END
// XEQ
XY*
/*

```


" No. DE GRANOS DE MAIZ GANADOS POR EL INVENTOR
DEL AJEDRES "



```

// JOB T
// FOR
*LIST SOURCE PROGRAM
*IOCS(CARD,1132 PRINTER)
*ONE WORD INTEGERS
C-----D I E C I S I E T E-----
C ESTE PROGRAMA CALCULA EL NUMERO DE GRANOS DE MAIZ QUE COBRO EL
C INVENTOR DE AJEDRES
C FILES.
    LEE=2
    IMP=3
C FORMATOS
100 FORMAT(10X,6HCUADRO,9X,4HSUMA,/)
101 FORMAT(3X,I2,2E15.7)
102 FORMAT(//)
103 FORMAT(53X,11H*****
104 FORMAT(53X,11H* FIN *)
    WRITE(IMP,100)
    SUM=0.0
    DO 200 I=1,64
        CUA=2.0**(I-1)
        SUM=SUM+CUA
        WRITE(IMP,101)I,CUA,SUM
200 CONTINUE
    WRITE(IMP,102)
    WRITE(IMP,103)
    WRITE(IMP,104)
    WRITE(IMP,103)
    CALL EXIT
    END
// XEQ
/*

```


CUADRO

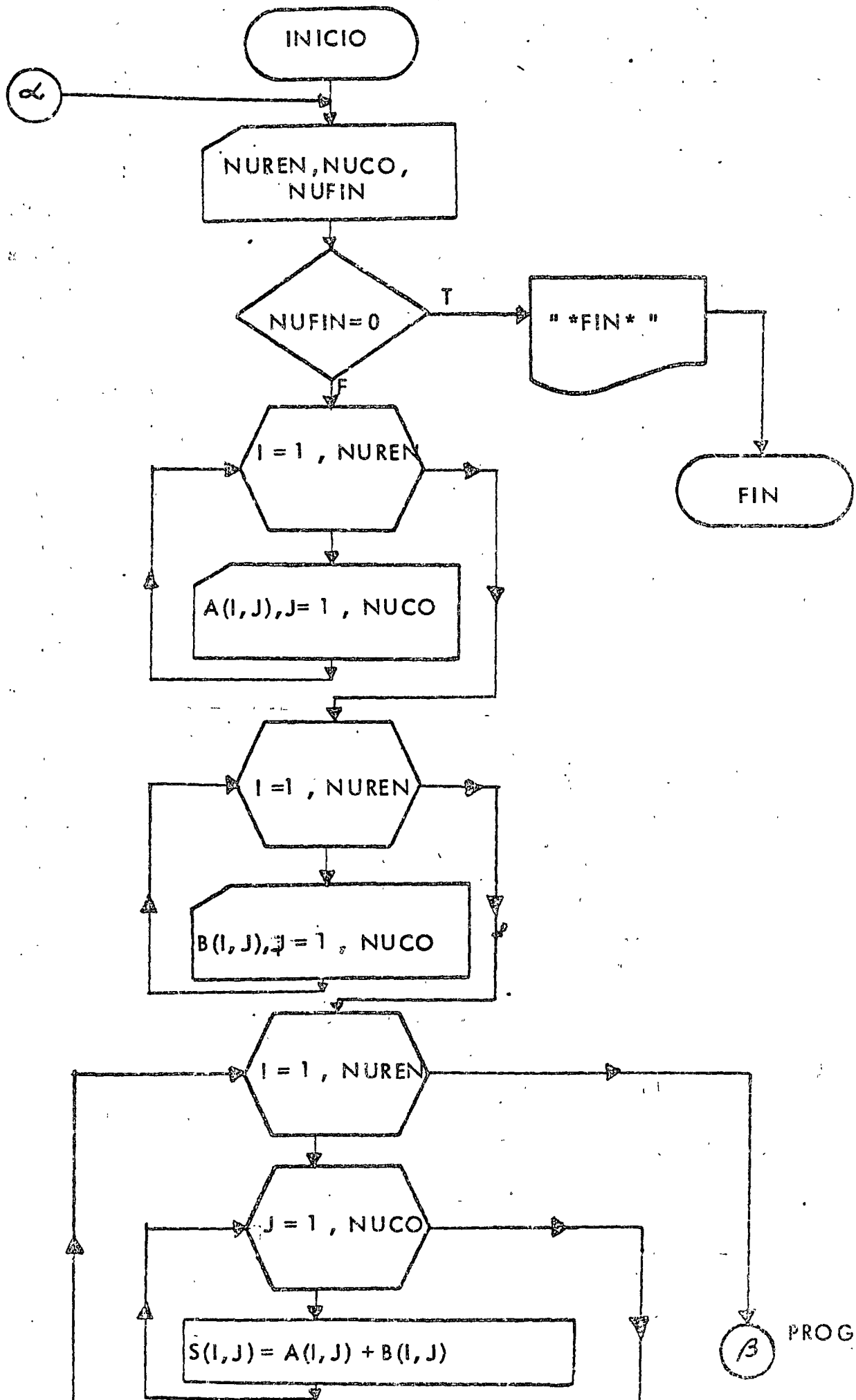
SUMA

RESULTADOS

1	10000000	+01	10000000	+01
2	20000000	+01	20000000	+01
3	40000000	+01	40000000	+01
4	80000000	+01	80000000	+01
5	160000000	+02	160000000	+02
6	320000000	+02	320000000	+02
7	640000000	+02	640000000	+02
8	1280000000	+03	1280000000	+03
9	2560000000	+03	2560000000	+03
10	5120000000	+03	5120000000	+03
11	10240000000	+04	10240000000	+04
12	20480000000	+04	20480000000	+04
13	40960000000	+04	40960000000	+04
14	81920000000	+04	81920000000	+04
15	163840000000	+05	163840000000	+05
16	327680000000	+05	327680000000	+05
17	655360000000	+05	655360000000	+05
18	1310720000000	+06	1310720000000	+06
19	2621440000000	+06	2621440000000	+06
20	5242880000000	+06	5242880000000	+06
21	10485760000000	+07	10485760000000	+07
22	20971520000000	+07	20971520000000	+07
23	41943040000000	+07	41943040000000	+07
24	83886080000000	+07	83886080000000	+07
25	167772160000000	+08	167772160000000	+08
26	335544320000000	+08	335544320000000	+08
27	671088640000000	+08	671088640000000	+08
28	1342177280000000	+09	1342177280000000	+09
29	2684354560000000	+09	2684354560000000	+09
30	5368709120000000	+09	5368709120000000	+09
31	10737418240000000	+10	10737418240000000	+10
32	21474836480000000	+10	21474836480000000	+10
33	42949672960000000	+10	42949672960000000	+10
34	85899345920000000	+10	85899345920000000	+10
35	171798691840000000	+11	171798691840000000	+11
36	343597383680000000	+11	343597383680000000	+11
37	687194767360000000	+11	687194767360000000	+11
38	1374389534720000000	+12	1374389534720000000	+12
39	2748779069440000000	+12	2748779069440000000	+12
40	5497558138880000000	+12	5497558138880000000	+12
41	10995116277760000000	+13	10995116277760000000	+13
42	21990232555520000000	+13	21990232555520000000	+13
43	43980465111040000000	+13	43980465111040000000	+13
44	87960930222080000000	+13	87960930222080000000	+13
45	175921604441600000000	+14	175921604441600000000	+14
46	351843208883200000000	+14	351843208883200000000	+14
47	703686417766400000000	+14	703686417766400000000	+14
48	1407372835332800000000	+15	1407372835332800000000	+15
49	2814745670665600000000	+15	2814745670665600000000	+15
50	5629491341331200000000	+15	5629491341331200000000	+15
51	11258982682662400000000	+16	11258982682662400000000	+16
52	22517965365324800000000	+16	22517965365324800000000	+16
53	45035930730649600000000	+16	45035930730649600000000	+16
54	90071861461292800000000	+16	90071861461292800000000	+16
55	180143722922579200000000	+17	180143722922579200000000	+17
56	360287445845158400000000	+17	360287445845158400000000	+17
57	720574891690316800000000	+17	720574891690316800000000	+17
58	1441149783380633600000000	+18	1441149783380633600000000	+18
59	2882299566761267200000000	+18	2882299566761267200000000	+18
60	5764599133522534400000000	+18	5764599133522534400000000	+18
61	1152919826644868800000000	+19	1152919826644868800000000	+19
62	2305839653289737600000000	+19	2305839653289737600000000	+19
63	4611679306579475200000000	+19	4611679306579475200000000	+19
64	9223358613158950400000000	+19	9223358613158950400000000	+19

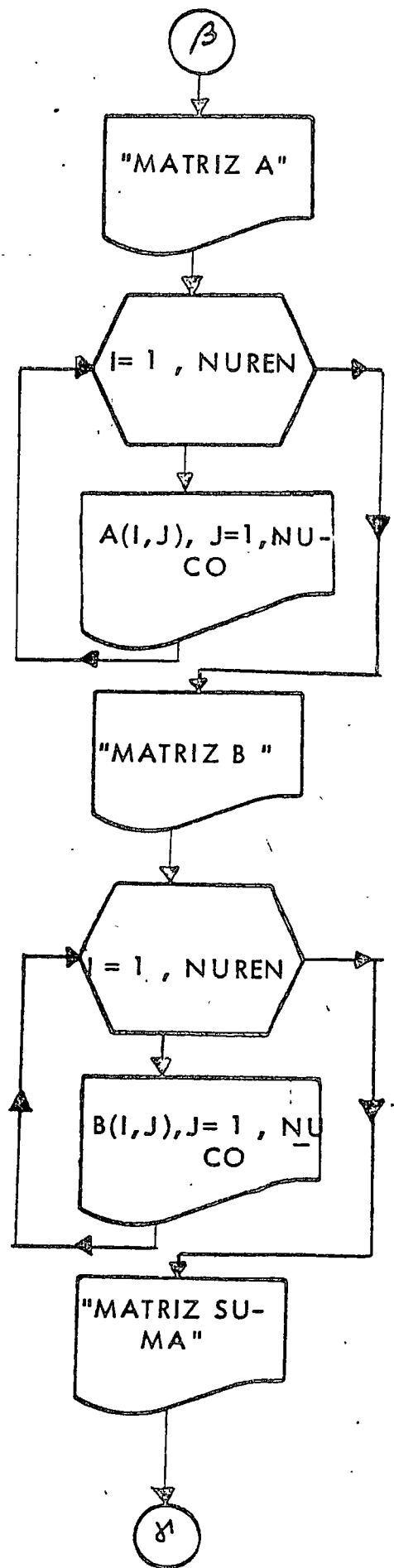
PROG - 17

 * FIN *



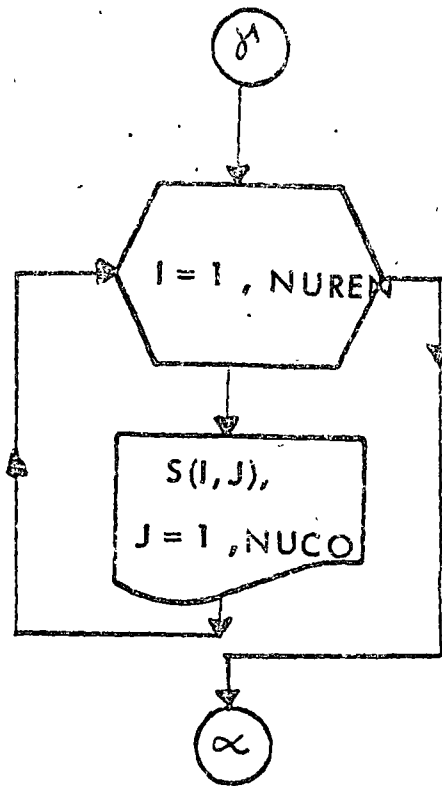
" SUMA DE DOS MATRICES, A y B "

2a.



" SUMA DE DOS MATRICES , A y B "

3a.



```

// JOB T
// FOR
*IOCS(CARD,1132 PRINTER)
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
C-----D I E C I O C H O-----
C   SUMA DE DOS MATRICES: A Y B
C   EL PROGRAMA ESTA HECHO PARA SUMAR DOS MATRICES DE 10X 10 MAXIMO.
C   SE RESERVAN LUGARES EN LA MEMORIA PARA LAS MATRICES A SUMAR Y PARA
C   LA MATRIZ SUMA.
C   DIMENSION A(10,10),B(10,10),S(10,10)
C   FILES
C       LEE=2
C       IMP=3
C   FORMATOS
100  FORMAT(3I2)
101  FORMAT(10F8.3)
102  FORMAT(///,5X,9HMATRIZ A:,//)
103  FORMAT(5X,10(F8.3,2X),/)
104  FORMAT(///,5X,9HMATRIZ B:,//)
105  FORMAT(///,5X,18HLA MATRIZ SUMA ES:,//)
106  FORMAT(53X,11H*****
107  FORMAT(53X,11H*   FIN   *)
C   LECTURA DEL NUMERO DE RENGLONES DE LAS MATRICES (NUREN) Y DEL NUME
C   RO DE COLUMNAS (NUCO),Y DE UN DETECTOR (NUFIN)
199  READ(LEE,100)NUREN,NUCO,NUFIN
C   ANALISIS DE NUFIN. SI VALE CERO YA NO SE EJECUTA EL PROGRAMA,DE LO
C   CONTRARIO SI.
C   IF(NUFIN.EQ.0)GO TO 1000
C   LECTURA POR RENGLONES DE LA MATRIZ A.
C   DO 200 I=1,NUREN
C       READ(LEE,101)(A(I,J),J=1,NUCO)
200  CONTINUE
C   LECTURA POR RENGLONES DE LA MATRIZ B.
C   DO 201 I=1,NUREN
C       READ(LEE,101)(B(I,J),J=1,NUCO)
201  CONTINUE
C   SE HARA LA SUMA ELEMENTO A ELEMENTO
C   DO 203 I=1,NUREN
C       DO 202 J=1,NUCO
C           S(I,J)=A(I,J)+B(I,J)
202  CONTINUE
203  CONTINUE
C   IMPRESION DE LA MATRIZ A POR RENGLONES
C   WRITE(IMP,102)
C   DO 204 I=1,NUREN
C       WRITE(IMP,103)(A(I,J),J=1,NUCO)
204  CONTINUE
C   IMPRESION DE LA MATRIZ B POR RENGLONES
C   WRITE(IMP,104)
C   DO 205 I=1,NUREN
C       WRITE(IMP,103)(B(I,J),J=1,NUCO)
205  CONTINUE
C   IMPRESION DE LA MATRIZ S POR RENGLONES
C   WRITE(IMP,105)
C   DO 206 I=1,NUREN
C       WRITE(IMP,103)(S(I,J),J=1,NUCO)
206  CONTINUE
C   GO TO 199
1000 CONTINUE
C   WRITE(IMP,106)
C   WRITE(IMP,107)
C   WRITE(IMP,106)

```

CALL EXIT
END

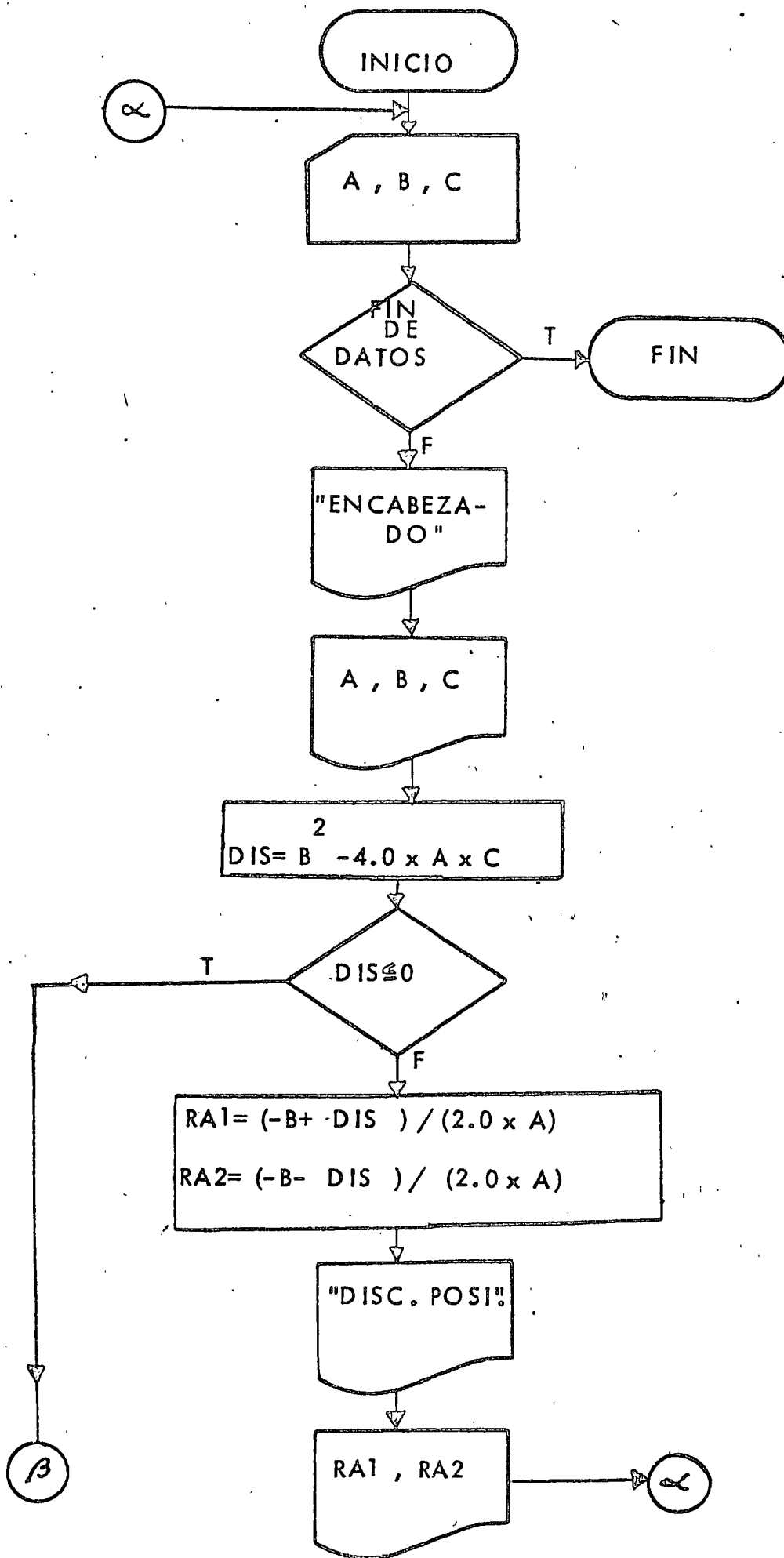
```
// XEQ  
2 2 1  
5.0 0.0  
12.2 3.4  
-4.7 2.1  
0.0 -1.2  
2 2 1  
90.15 -87.02  
5.478 12.22  
-90.15 87.02  
-5.478 -12.22  
2 2 0  
/*
```

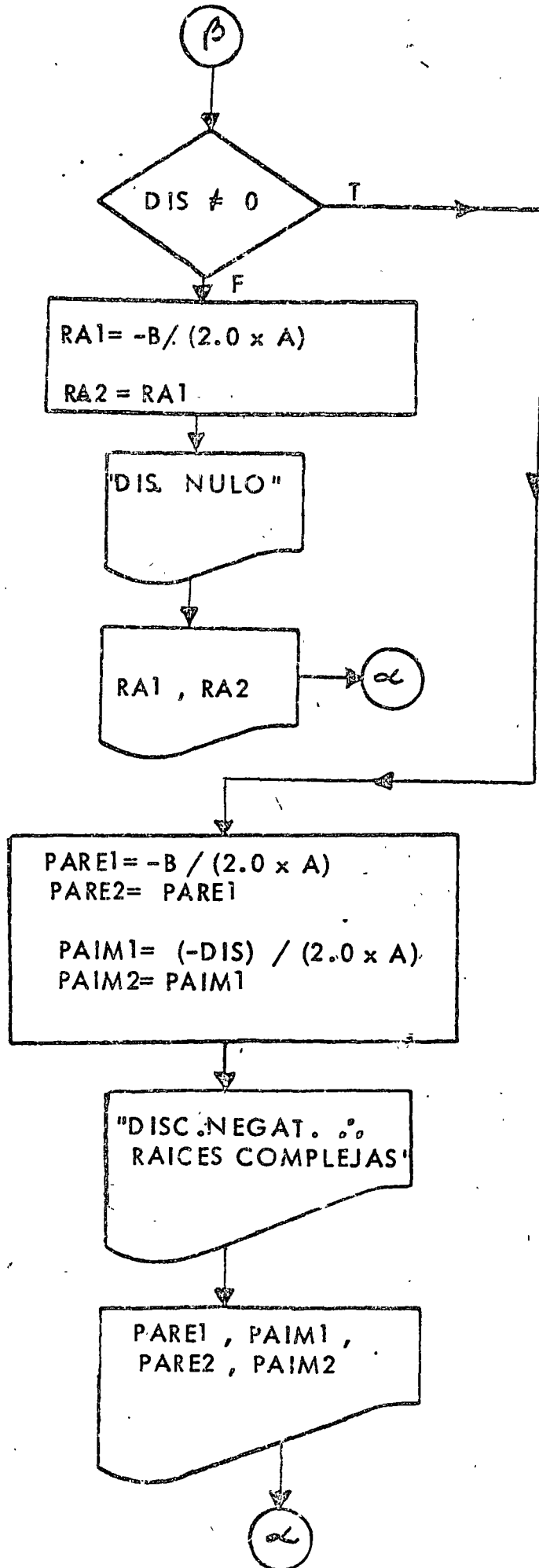
RESULTADOS

MATRIZ A:	
5.000	0.000
12.200	3.400
MATRIZ B:	
-4.700	2.100
0.000	-1.200
LA MATRIZ SUMA ES:	
0.300	2.100
12.200	2.200
MATRIZ A:	
90.150	-87.020
5.478	12.220
MATRIZ B:	
-90.150	87.020
-5.478	-12.220
LA MATRIZ SUMA ES:	
0.000	0.000
0.000	0.000

PROG - 18

* FIN *






```

// JOB T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----D I E C I N U E V E-----
C      SOLUCION DE ECUACIONES CUADRATICAS.
100 FORMAT(3F11.5)
101 FORMAT(///,2X,36HLOS COEFICIENTES DE LA ECUACION SON,/)
102 FORMAT(2X,3HA= ,F11.5,2X,3HB= ,F11.5,2X,3HC= ,F11.5,/)
103 FORMAT(2X,52HEL DISCRIMINANTE ES POSITIVO,POR TANTO RAICES REALES,
1/)
104 FORMAT(5X,3HX1=,F11.5,10X,3HX2=,F11.5,/)
105 FORMAT(2X,49HEL DISCRIMINANTE ES NULO,POR TANTO RAICES IGUALES,/)
106 FORMAT(5X,3HX1=,F11.5,10X,3HX2=,F11.5,/)
107 FORMAT(2X,55HEL DISCRIMINANTE ES NEGATIVO,POR TANTO RAICES COMPLEJ
IAS,/)
108 FORMAT(5X,3HX1=,F11.5,2H +,F11.5,4H IMA,10X,3HX2=,F11.5,2H -,F11.5
1,4H IMA,/)
      LEE=2
      IMP=3
C      LEE LOS COEFICIENTES
200 READ(LEE,100,END=240)A,B,C
C      IMPRIME LA ECUACION
      WRITE(IMP,101)
      WRITE(IMP,102)A,B,C
C      CALCULO DEL DISCRIMINANTE
      DIS=B**2-4.0*A*C
      IF(DIS.LE. 0.0)GO TO 210
C      RAICES REALES DIFERENTES
      RA1=(-B+SQRT(DIS))/(2.0*A)
      RA2=(-B-SQRT(DIS))/(2.0*A)
      WRITE(IMP,103)
      WRITE(IMP,104)RA1,RA2
      GO TO 230
210 CONTINUE
      IF(DIS.NE.0.0)GO TO 220
C      RAICES REALES IGUALES
      RA1=-B/(2.0*A)
      RA2=RA1
      WRITE(IMP,105)
      WRITE(IMP,106)RA1,RA2
      GO TO 230
220 CONTINUE
C      RAICES COMPLEJAS
      PARE1=-B/(2.0*A)
      PARE2=PARE1
      PAIM1=SQRT(-DIS)/(2.0*A)
      PAIM2=PAIM1
      WRITE(IMP,107)
      WRITE(IMP,108)PARE1,PAIM1,PARE2,PAIM2
      GO TO 230
C      ENDIF
C      ENDIF
230 CONTINUE
      GO TO 200
240 CONTINUE
      CALL EXIT
      END

```

```

// XEQ
1.0      2.0      3.0      1.
5.0      10.0     5.0      1.
1.0      10.0     27.0     1.

```

3.0 20.0 1.0 1.

RESULTADOS

LOS COEFICIENTES DE LA ECUACION SON

A= 1.00000 B= 2.00000 C= 3.00000

EL DISCRIMINANTE ES NEGATIVO, POR TANTO RAICES COMPLEJAS

X1= -1.00000 + 1.41421 IMA X2= -1.00000 - 1.41421 IMA

LOS COEFICIENTES DE LA ECUACION SON

A= 5.00000 B= 10.00000 C= 5.00000

EL DISCRIMINANTE ES NULO, POR TANTO RAICES IGUALES

X1= -1.00000 X2= -1.00000

LOS COEFICIENTES DE LA ECUACION SON

A= 1.00000 B= 10.00000 C= 27.00000

EL DISCRIMINANTE ES NEGATIVO, POR TANTO RAICES COMPLEJAS

X1= -5.00000 + 1.41421 IMA X2= -5.00000 - 1.41421 IMA

LOS COEFICIENTES DE LA ECUACION SON

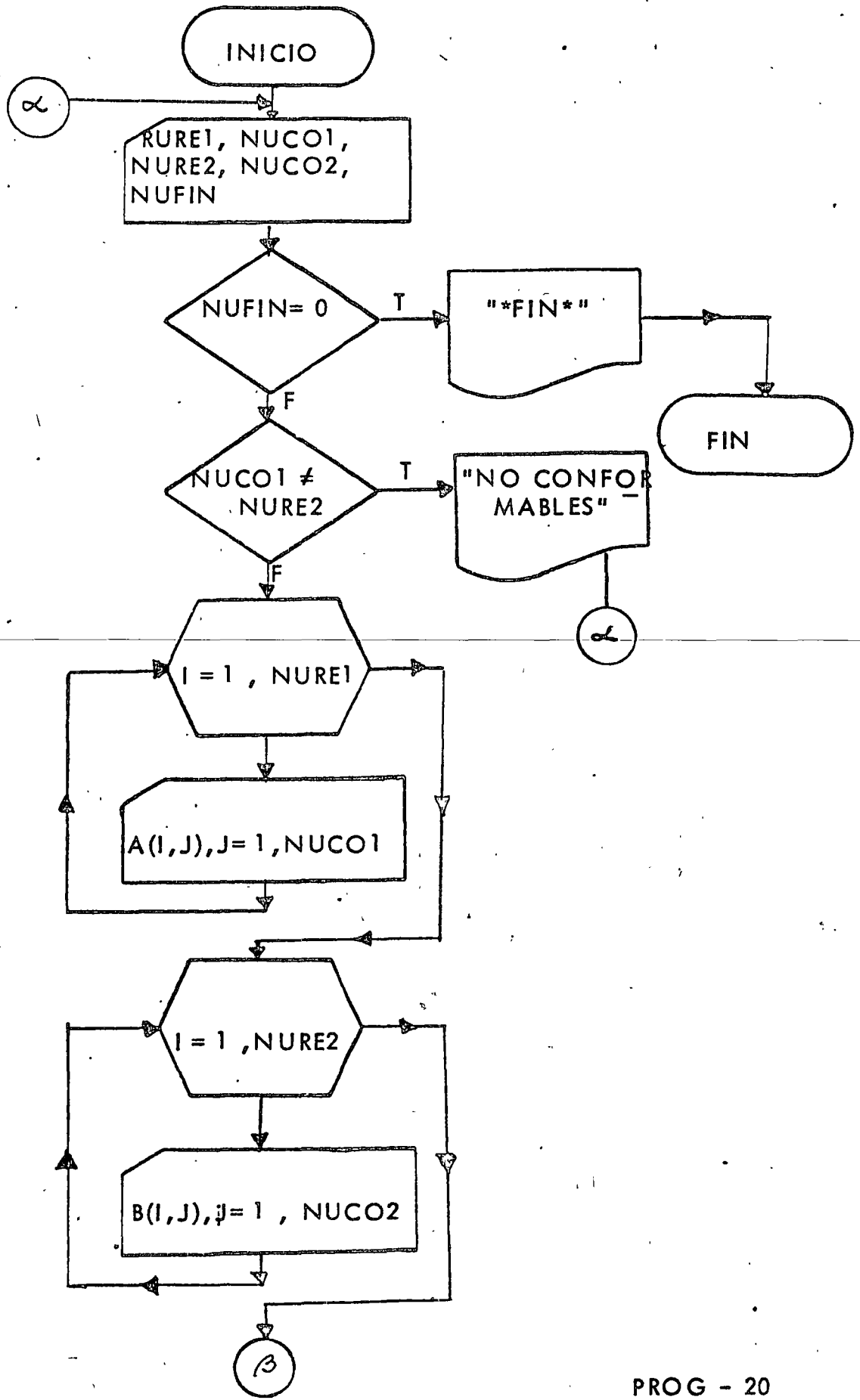
A= 3.00000 B= 20.00000 C= 1.00000

EL DISCRIMINANTE ES POSITIVO, POR TANTO RAICES REALES

X1= -0.05038 X2= -6.61629

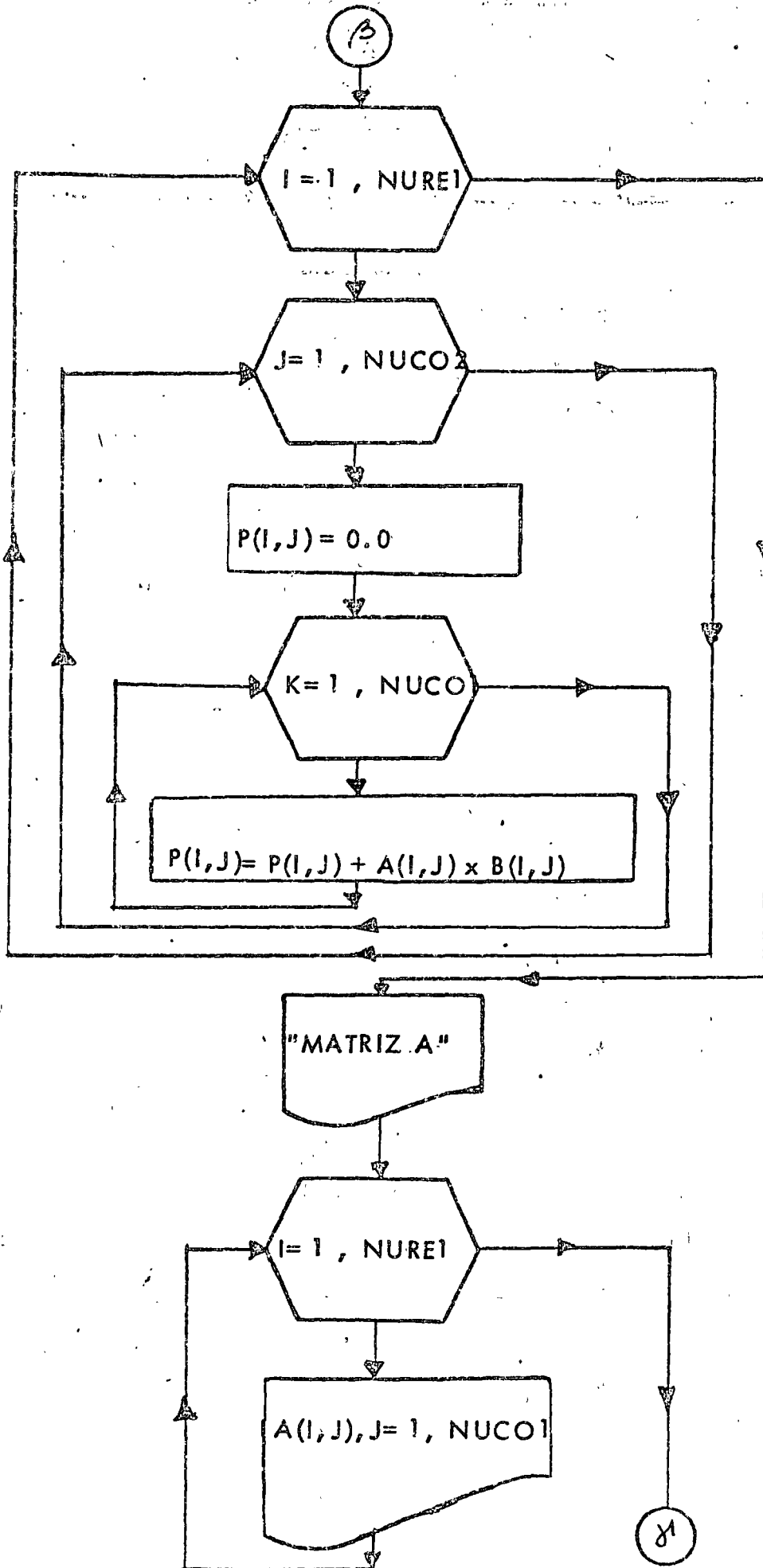
" PRODUCTO DE DOS MATRICES "

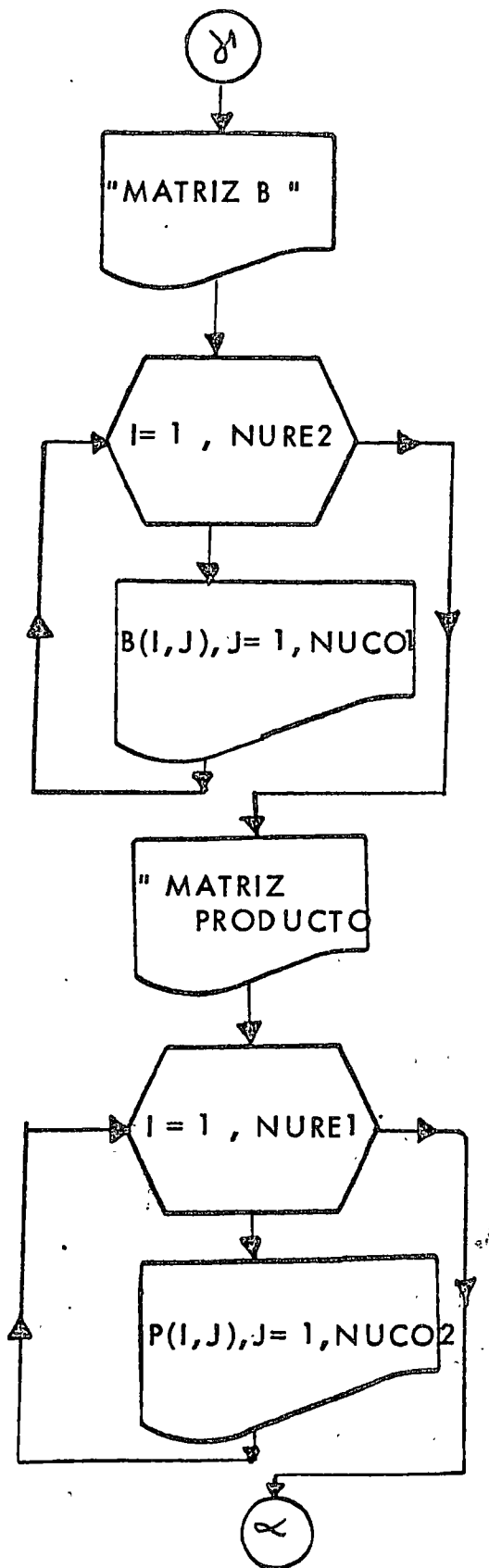
1a.



" PRODUCTO DE DOS MATRICES "

2a.





```

// JOB: T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----V E I N T E -----
C   EL PROGRAMA REALIZA EL PRODUCTO DE DOS MATRICES DE 10 X 10 MAXIMO.
C   UNA ES LA MATRIZ A(NURE1,NUCO1).
C   LA OTRA ES LA MATRIZ B(NURE2,NUCO2).
C   SE RESERVAN LUGARES EN LA MEMORIA PARA LAS MATRICES QUE SE VAN A
C   MULTIPLICAR Y PARA LA MATRIZ PRODUCTO.
C   DIMENSION A(10,10),B(10,10),P(10,10)
C   FILES
      LEE=2
      IMP=3
C   FORMATOS
100  FORMAT(5I2)
101  FORMAT(10F8.3)
102  FORMAT(///,5X,9HMATRIZ A:,//)
103  FORMAT(5X,10(F8.3,2X),/)
104  FORMAT(///,5X,9HMATRIZ B:,//)
105  FORMAT(///,5X,22HLA MATRIZ PRODUCTO ES:,//)
106  FORMAT(5X,10E15.7,/)
107  FORMAT(///,5X,76HEL PRODUCTO NO SE PUEDE LLEVAR A CABO YA QUE LAS
      1MATRICES NO SON CONFORMABLES,///)
108  FORMAT(///)
109  FORMAT(53X,11H***** )
110  FORMAT(53X,11H*   FIN   *)
C   LECTURA DE LOS NUMEROS DE RENGLONES Y DE COLUMNAS DE CADA MATRIZ
C   Y DEL DETECTOR NUFIN.
199  READ(LEE,100)NURE1,NUCO1,NURE2,NUCO2,NUFIN
C   ANALISIS DEL VALOR DE NUFIN. SI VALE CERO EL PROGRAMA NO SE LLEVA
C   A CABO, DE LO CONTRARIO SI.
      IF(NUFIN.EQ.0)GO TO 1000
C   SE VE SI LAS MATRICES SON CONFORMABLES.
      IF(NUCO1.NE.NURE2)GO TO 900
C   LECTURA POR RENGLONES DE LA MATRIZ A.
      DO 200 I=1,NURE1
        READ(LEE,101)(A(I,J),J=1,NUCO1)
C 200  CONTINUE
C   LECTURA POR RENGLONES DE LA MATRIZ B.
      DO 201 I=1,NURE2
        READ(LEE,101)(B(I,J),J=1,NUCO2)
C 201  CONTINUE
C   SE REALIZA EL PRODUCTO
      DO 204 I=1,NURE1
        DO 203 J=1,NUCO2
          P(I,J)=0.0
          DO 202 K=1,NUCO1
            P(I,J)=P(I,J)+A(I,K)*B(K,J)
C 202  CONTINUE
C 203  CONTINUE
C 204  CONTINUE
C   IMPRESION DE LA MATRIZ A POR RENGLONES
      WRITE(IMP,102)
      DO 205 I=1,NURE1
        WRITE(IMP,103)(A(I,J),J=1,NUCO1)
C 205  CONTINUE
C   IMPRESION DE LA MATRIZ B POR RENGLONES.
      WRITE(IMP,104)
      DO 206 I=1,NURE2
        WRITE(IMP,103)(B(I,J),J=1,NUCO1)
C 206  CONTINUE

```

```

C      IMPRESION DE LA MATRIZ P POR RENGLONES.
      WRITE(IMP,105)
      DO 207 I=1,NURE1
        WRITE(IMP,106)(P(I,J),J=1,NUC02)
207    CONTINUE
      GO TO 199
900    CONTINUE
      WRITE(IMP,107)
      GO TO 199
1000  CONTINUE
      WRITE(IMP,108)
      WRITE(IMP,109)
      WRITE(IMP,110)
      WRITE(IMP,109)
      CALL EXIT
      END

```

```

// XEQ
2 2 2 210
  15.54 -42.07
  -1.22  0.0
1950.75 -12.0
  0.001  5.0
2 2 2 210
  0.2    98.75
 -12.5   32.52
1000.01  0.0
  -1.52  15.51
2 3 2 210
2 2 2 200
/*

```

RESULTADOS

MATRIZ A:

15.540 -2.070
0.000 -1.220

MATRIZ B:

1950.750 -12.000
0.001 5.000

LA MATRIZ PRODUCTO ES:

.3031461E+05 -.3968300E+03
-.1220000E-02 -.6100000E+01

MATRIZ A:

0.200 98.750
0.000 -12.500

MATRIZ B:

1000.010 0.000
-1.520 15.510

LA MATRIZ PRODUCTO ES:

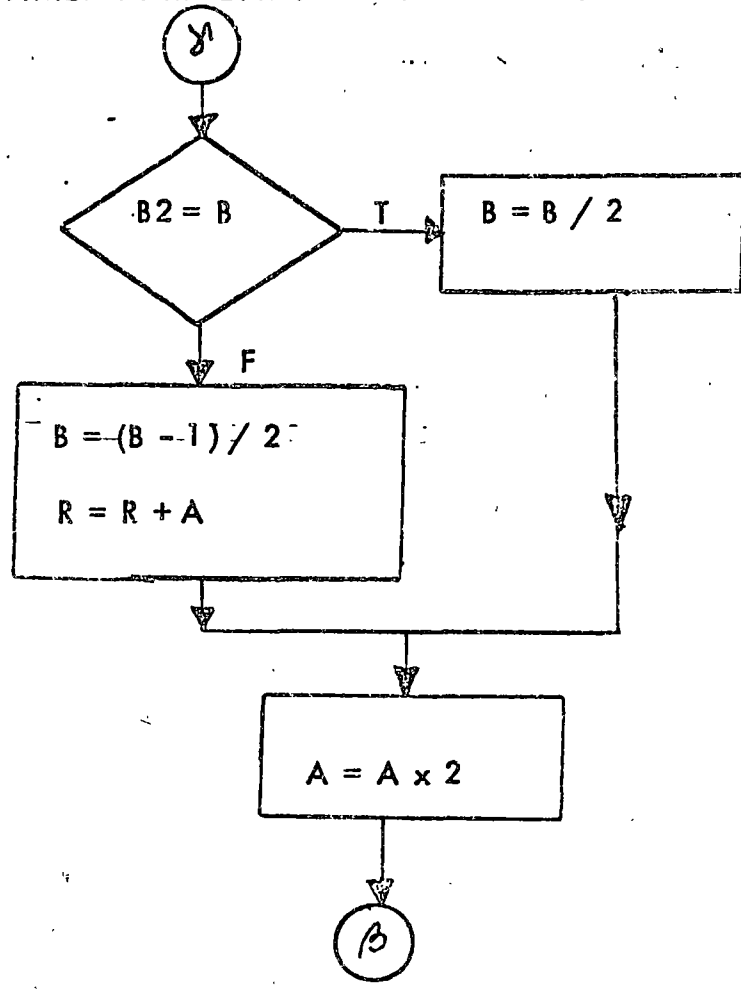
.4990200E+02 .1531613E+04
.1900000E+02 -.1938750E+03

EL PRODUCTO NO SE PUEDE LLEVAR A CABO YA QUE LAS MATRICES NO SON CONFORMABLES

* FIN *

" MULTIPLICACION DE DOS NUMEROS UTILIZANDO EXCLUSIVAMENTE MULTIP. Y DIVISION POR 2 "

2a.



```

// JOB T
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1132 PRINTER)
C-----V E I N T I U N O -----
C      MULTIPLICACION DE DOS NUMEROS
C      UTILIZANDO SOLO MULTIPLICACIONES Y
C      DIVISIONES POR 2
C      INTEGER A,B,C,R,AA,BB
C      WRITE (3,101)
101. FORMAT (1H1)
C      WRITE (3,102)
102 FORMAT (9X,1HA,3X,1HX,4X,1HB,4X,1H=,4X,1HC)
200 READ (2,100,END=260)A,B
100 FORMAT (2I3)
C      R=0
C      AA=A
C      BB=B
210 IF(B.EQ.1) GO TO 240
C      B1=B/2
C      B2=B1*2
C      IF(B2.EQ.B) GO TO 220
C      ES IMPAR
C      B=(B-1)/2
C      R=R+A
C      GO TO 230
220 CONTINUE
C      ES PAR
C      B=B/2
230 CONTINUE
C      A=A*2
C      GO TO 210
240 CONTINUE
C      C=A+R
C      WRITE (3,103) AA,BB,C
103 FORMAT (3I10)
C      GO TO 200
260 CALL EXIT
END

```

```

// XEQ
60 80
19 17
68 35
40 11
77 99
/*

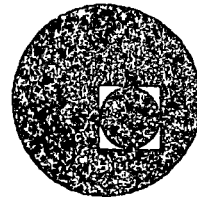
```

RESULTADOS

A	X	B	=	C
60	80	80		4800
19	17	17		323
68	35	35		2380
40	11	11		440
77	99	99		7623



centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



METODOS NUMERICOS Y APLICACIONES CON LA COMPUTADORA
DIGITAL

TEMA 2 : ALGEBRA MATRICIAL

-ABRIL, 1978.

2.1 MATRICES

2.1.1 DEFINICIONES

DEFINICION 2.1 Una matriz $m \times n$, es un arreglo rectangular de números reales, llamados los elementos de la matriz, los cuales estén arreglados en m renglones y n columnas en la siguiente forma:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

NOTAS:

- i) Una representación simplificada del arreglo anterior es $[a_{ij}]_{mn}$.
El símbolo a_{ij} representa al elemento que está en el i -ésimo renglón y en la j -ésima columna del arreglo.
- ii) Las matrices se representarán por letras mayúsculas gruesas, entonces si A es una matriz $m \times n$, significa que $A = [a_{ij}]_{mn}$.
- iii) Si $m = n$, entonces se dice que se tiene una matriz cuadrada de orden m .
- iv) Si A es $m \times 1$, entonces se dice que A es una matriz columna.
- v) Si A es $1 \times n$, entonces se dice que A es una matriz renglón.

EJEMPLOS. Las matrices A , B , C y D mostradas a continuación.

$$A = \begin{bmatrix} 1 & 12 & -1 & 2 \\ 0 & 1 & 2 & 3 \\ 1 & 4 & 2 & 1 \end{bmatrix} \quad B = [1 \ 0 \ 2 \ 1] \quad , \quad C = \begin{bmatrix} 2 & 2 \\ 1 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

tienen las siguientes características: A es 3×4 , B es 1×4 , (es una matriz renglón), C es 2×2 (es una matriz cuadrada), D es 3×1 (es una matriz columna)

NOTACION 1.

- i) Sea $A = [a_{ij}]$ una matriz $m \times n$. El i -ésimo renglón de A , se indicará por $A_{i.}$, i.e.
 $A_{i.} = [a_{i1} \ a_{i2} \ \cdots \ a_{in}]$.

ii) Sea $A = [a_{ij}]$ una matriz $m \times n$. La j -ésima columna de A , se indicará por $A_{.j}$, i.e.

$$A_{.j} = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{bmatrix}$$

EJEMPLOS. Para la matriz $A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 2 & 2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$ encuentre

- i) el primer renglón de A
- ii) el tercer renglón de A
- iii) la primera columna de A
- iv) la cuarta columna de A

Solución

i) $A_{1.} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \end{bmatrix}$
 ii) $A_{3.} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \end{bmatrix}$

iii) $A_{.1} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$

iv) $A_{.4} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

DEFINICION 2.2. Se dice que las matrices $A = [a_{ij}]_{mn}$ y $B = [b_{ij}]_{mn}$ son iguales si y solo si

$$a_{ij} = b_{ij} \quad \forall i = 1, 2, \dots, m \text{ y } \forall j = 1, 2, \dots, n$$

NOTA: De la definición anterior se observa una condición para que dos matrices -- sean iguales es que ambas tengan el mismo número de renglones y el mismo número de columnas

EJEMPLO. En las siguientes matrices

$$A = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \\ 2 & 0 \end{bmatrix}$$

se observa que :

$A = B$ porque $a_{ij} = b_{ij} \quad \forall i = 1, 2; \quad j = 1, 2.$

$A \neq C$ porque para $i = 2$ y $j = 2$ se tiene que $a_{22} = 1 \neq c_{22} = 0$

DEFINICION 2.3. Las matrices $A = [a_{ij}]_{mn}$ y $B = [b_{ij}]_{mn}$ son iguales si y solo si

$$A_{i.} = B_{i.} \quad \forall i = 1, 2, \dots, m$$

DEFINICION 2.4. Las matrices $A = [a_{ij}]_{mn}$ y $B = [b_{ij}]_{mn}$ son iguales si y solo si

$$A_{.j} = B_{.j} \quad \forall j = 1, 2, \dots, n$$

TEOREMA 2.4. Las definiciones 2.2, 2.3 y 2.4 son equivalentes, ie.

DEFINICION 2.2 \Leftrightarrow DEFINICION 2.3

DEFINICION 2.2 \Leftrightarrow DEFINICION 2.4

DEFINICION 2.3 \Leftrightarrow DEFINICION 2.4

DEMOSTRACION. La demostración es simple, solo considere la definición de igualdad de matrices (cualquiera de ellas) y la notación 1. Los detalles se piden en la tarea número 1.

DEFINICION 2.5. La suma de dos matrices $A = [a_{ij}]_{mn}$ y $B = [b_{ij}]_{mn}$, indicada por $A + B$, es una matriz $C = [c_{ij}]$ definida por

$$c_{ij} = a_{ij} + b_{ij} \quad \forall i = 1, 2, \dots, m, j = 1, 2, \dots, n$$

NOTA.

i) La definición anterior expresada en otros términos es

$$C = [c_{ij}] = [a_{ij} + b_{ij}] \stackrel{d}{=} A + B$$

ii) De la definición 2.5, se observa que una condición necesaria para la adición de matrices es que ambas tengan igual número de renglones y de columnas.

EJEMPLO. Para las siguientes matrices

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix}; \quad B = \begin{bmatrix} 1 & 4 & 5 \\ 7 & 8 & 9 \end{bmatrix}; \quad C = \begin{bmatrix} 2 & 3 & 5 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

se tiene que

$$A + B = \begin{bmatrix} 1+1 & 2+4 & 3+5 \\ 0+7 & 0+8 & 1+9 \end{bmatrix} = \begin{bmatrix} 2 & 6 & 8 \\ 7 & 8 & 10 \end{bmatrix}$$

$A + C$ no se define porque tienen distinto número de columnas

$B + C$ no se define por la misma razón

TEOREMA 2.6. (PROPIEDADES DE LA ADICION DE MATRICES)

i) $A + B = B + A$ (la adición es conmutativa)

ii) $(A + B) + C = A + (B + C)$ (La adición es asociativa).

DEMOSTRACION.

i) $A + B \stackrel{d}{=} [a_{ij} + b_{ij}]$

Por otro lado, conocemos de la teoría de los números reales que la adición de los reales es conmutativa, i.e., $a_{ij} + b_{ij} = b_{ij} + a_{ij}$, por lo tanto,

$$A + B \stackrel{d}{=} [a_{ij} + b_{ij}] \Rightarrow A + B = [b_{ij} + a_{ij}] \stackrel{d}{=} B + A \quad \square$$
$$\Rightarrow A + B = B + A$$

$$\begin{aligned} \text{ii)} \quad (A+B) + C &= ([a_{ij}] + [b_{ij}]) + [c_{ij}] \\ &= [a_{ij} + b_{ij}] + [c_{ij}] \\ &= [(a_{ij} + b_{ij}) + c_{ij}] \end{aligned}$$

Pero también conocemos de teoría de los números que en los números reales la adición es asociativa, i.e.,

$$(a_{ij} + b_{ij}) + c_{ij} = a_{ij} + (b_{ij} + c_{ij})$$

Por lo tanto,

$$\begin{aligned} (A+B) + C &= [a_{ij} + (b_{ij} + c_{ij})] \\ &= [a_{ij}] + [b_{ij} + c_{ij}] \\ &= A + (B+C) \quad \square \end{aligned}$$

DEFINICION 2.7. Sea $A = [a_{ij}]$ una matriz $m \times n$ y sea k un número real. La multiplicación de una matriz A por un número real k , indicado por kA , es una matriz $m \times n$ definida por

$$kA = [ka_{ij}]$$

NOTA: A los números reales también se les llama escalares, por lo que a la multiplicación de un real por una matriz también se le llama multiplicación escalar.

EJEMPLO. Si $k = -4$ y $A = \begin{bmatrix} -2 & 3 \\ 1 & 2 \end{bmatrix}$ entonces

$$kA = \begin{bmatrix} 8 & -12 \\ -4 & -8 \end{bmatrix}$$

TEOREMA 2.8 (PROPIEDADES DEL PRODUCTO DE UN REAL POR UNA MATRIZ)

- i) $1 A = A$
- ii) $k(A+B) = kA + kB$
- iii) $(k_1 + k_2) A = k_1 A + k_2 A$
- iv) $(k_1 k_2) A = k_1 (k_2 A)$

DEMOSTRACION. Es trivial, solo aplique la definici3n 2.7, propiedades de n3meros reales y definiciones o propiedades de matrices presentadas anteriormente. Intente hacerlo.

DEFINICION 2.9. Sea $A = [a_{ij}]$ $m \times n$ y sea $B = [b_{ij}]$ $n \times p$. La multiplicaci3n de A por B, indicada por AB , es una matriz de elementos c_{ij} definida por

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

NOTA. Una matriz A y B se pueden multiplicar si y solo si el n3mero de columnas de A es igual al n3mero de renglones de B.

EJEMPLO Si

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \quad B = \begin{bmatrix} 7 & 4 \\ 3 & 1 \end{bmatrix}$$

entonces

$$AB = \begin{bmatrix} 1 \times 7 + 2 \times 3 & 1 \times 4 + 2 \times 1 \\ 3 \times 7 + 4 \times 3 & 3 \times 4 + 4 \times 1 \\ 5 \times 7 + 6 \times 3 & 5 \times 4 + 6 \times 1 \end{bmatrix} = \begin{bmatrix} 13 & 6 \\ 33 & 16 \\ 53 & 26 \end{bmatrix}$$

EJEMPLO.

$$A = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix}; \quad B = [3 \quad 4 \quad 1 \quad 5]$$

$$AB = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix} [3 \quad 4 \quad 1 \quad 5] = \begin{bmatrix} 3 & 4 & 1 & 5 \\ 6 & 8 & 2 & 10 \\ 0 & 0 & 0 & 0 \\ 3 & 4 & 1 & 5 \end{bmatrix}$$

$$BA = [3 \quad 4 \quad 1 \quad 5] \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix} = 3 + 8 + 5 = 16$$

Este ejemplo muestra que la multiplicaci3n de dos matrices no es conmutativa, ie. $AB \neq BA$.

PROPOSICION 2.10 (REPRESENTACIONES MATRICIALES DE ALGUNAS EXPRESIONES ALGEBRAICAS O NUMERICAS).

1. Si $x = [x_1 \ x_2 \ \dots \ x_n]$ $y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \Rightarrow \sum_{k=1}^n x_k y_k = xy$

2. Si $x = [x_1 \ x_2 \ \dots \ x_n] \Rightarrow \sum_{k=1}^n x_k^2 = xx^t$ donde $x^t = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

3. Sea $u = [1 \ 1 \ \dots \ 1]$ un vector de n componentes, los cuales son todos unos.

i) Si $x = [x_1 \ x_2 \ \dots \ x_n] \Rightarrow \sum_{k=1}^n x_k = xu^t$ donde $u^t = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$

ii) $n = uu^t$

El vector $u = [1 \ 1 \ \dots \ 1]$ es llamado el vector suma debido a la propiedad de poder representar matricialmente una suma de números x_i (ver propiedad 3 i)).

DEMOSTRACION. Es simple, sólo use definición de producto de matrices para probar que el lado izquierdo de cada igualdad dada es igual al lado derecho de la misma.

DEFINICION 2.11 Si A es $m \times n$, B es $n \times p$, entonces el producto de A por B , indicado por AB , es una matriz de elementos c_{ij} definidos por

$$c_{ij} = A_i \cdot B_{\cdot j}$$

PROPOSICION 2.12 Las definiciones 2.9 y 2.11 son equivalentes.

DEMOSTRACION. Debemos demostrar que: Definición 2.9 \iff Definición 2.11.

Demostración de la implicación (\implies):

Si $AB = c_{ij} \implies c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$ por la definición 2.9

$\implies c_{ij} = \begin{bmatrix} a_{i1} & a_{i2} & \dots & a_{in} \end{bmatrix} \begin{bmatrix} b_{1j} \\ b_{2j} \\ \vdots \\ b_{nj} \end{bmatrix}$ por Proposición 2.10 parte 1.

$$\implies c_{ij} = A_{i.} B_{.j} \quad \text{por la notación 1}$$

\implies Definición 2.11

Demostración de la implicación (\Leftarrow). Como las implicaciones de la demostración anterior (\implies) son reversibles en cada paso, entonces queda demostrado que la implicación 2.11 implica la definición 2.9 \square

PROPOSICION 2.13.

- i) $(AB)_{ij} = A_{i.} B_{.j}$
- ii) $(AB)_{i.} = A_{i.} B$
- iii) $(AB)_{.j} = AB_{.j}$
- iv) $(ABC)_{ij} = A_{i.} BC_{.j}$

DEMOSTRACION.

DEMOSTRACION DE i). Este resultado es solo un restablecimiento de la definición 2.11.

$$AB = [c_{ij}] \implies (AB)_{ij} = c_{ij} \implies (AB)_{ij} = A_{i.} B_{.j} \quad \square \quad \text{porque } c_{ij} = A_{i.} B_{.j} \text{ de acuerdo con definición 2.11}$$

DEMOSTRACION DE ii). Mostraremos primero dos resultados

$$y \quad (A_{i.} B)_{lj} = (AB)_{ij} \quad (*)$$

$$(A_{i.} B)_{lj} = (AB)_{ij} \quad (**)$$

para luego concluir que el lado izquierdo de (*) es igual al lado izquierdo de (**), y por último mostrar que los elementos del renglón $(AB)_{i.}$ son de la forma (*) y que los elementos del renglón $A_{i.} B$ son de la forma (**), y así probar que $(AB)_{i.} = A_{i.} B$

Demostración de (-):

$$(A_{i.} B)_{lj} = (A_{i.})_{l.} B_{.j} \quad \text{por la parte i) de esta proposición.}$$

$$\implies (A_{i.} B)_{lj} = A_{i.} B_{.j} \quad \text{porque } A_{i.} \text{ es una matriz con un sólo renglón, por lo tanto el primer renglón de } A_{i.} \text{ es el mismo } A_{i.}.$$

$$\implies (A_{i.} B)_{lj} = (AB)_{ij} \quad \text{por la parte i) de esta proposición.}$$

Demostración de (**):

$(AB)_{i.}$ es el i -ésimo renglón de AB , lo cual implica que $(AB)_{i.}$ es una matriz de un solo renglón. Entonces el primer renglón de $(AB)_{i.}$ es la misma matriz $(AB)_{i.}$, expresando esta conclusión simbólicamente se tiene que

$$((AB)_{i.})_{1.} = (AB)_{i.}$$

Por lo tanto, si tomamos la j -ésima columna de $((AB)_{i.})_{1.}$, equivale a tomar la j -ésima columna de $(AB)_{i.}$, simbólicamente

$$((AB)_{i.})_{1j} = (AB)_{ij} \quad \square$$

Una vez demostradas (*) y (**) se tiene que

$$(*) \text{ y } (**) \Rightarrow (A_{i.}B)_{1j} = ((AB)_{i.})_{1j} \quad \forall j \quad (***)$$

Falta ahora demostrar que los elementos del renglón $(AB)_{i.}$ son de la forma $(A_{i.}B)_{1j}$, y que los elementos del renglón $A_{i.}B$ son de la forma $((AB)_{i.})_{1j}$ y por (***) concluir que $(AB)_{i.} = A_{i.}B$:

$$\begin{aligned} (AB)_{i.} &= [(AB)_{i1} \quad (AB)_{i2} \quad \dots \quad (AB)_{in}] \\ &= [((AB)_{i.})_{11} \quad ((AB)_{i.})_{12} \quad \dots \quad ((AB)_{i.})_{1n}] && \text{por (**)} \\ &= [(A_{i.}B)_{11} \quad (A_{i.}B)_{12} \quad \dots \quad (A_{i.}B)_{1n}] && \text{por (***)} \\ &= A_{i.}B \quad \square && \text{por definición de } A_{i.}B \end{aligned}$$

DEMOSTRACION DE iii). Es similar a la anterior. Inténtela.

DEMOSTRACION DE iv).

$$(ABC)_{ij} = ((AB)C)_{ij}$$

$$= (AB)_{i.}C_{.j}$$

$$= A_{i.}BC_{.j} \quad \square$$

asociatividad en la multiplicación de matrices

parte i) de esta proposición.

propiedad ii) de esta proposición.

TEOREMA 2.14 (PROPIEDADES DE LA MULTIPLICACION DE MATRICES)

- i) $A(B + C) = AB + AC$
- ii) $(A + B)C = AC + BC$
- iii) $A(BC) = (AB)C$

DEMOSTRACION.

i) Debemos demostrar que el elemento (i, j) de $A(B+C)$ es igual al elemento (i, j) de $AB+AC$, para todo par (i, j) :

$$\{A(B+C)\}_{ij} = \sum_k a_{ik}(B+C)_{kj} \quad \text{definición de producto de matrices}$$

$$= \sum_k a_{ik}(b_{kj} + c_{kj}) \quad \text{definición de edición de matrices}$$

$$= \sum_k a_{ik}b_{kj} + \sum_k a_{ik}c_{kj} \quad \text{propiedad distributiva de la multiplicación con respecto a la edición en los números reales}$$

$$= (AB)_{ij} + (AC)_{ij} \quad \forall (i, j)$$

$$= (AB + AC)_{ij} \quad \square$$

ii) Es similar a parte i).

$$\text{iii) } \{A(BC)\}_{ij} = \sum_k a_{ik}(BC)_{kj} \quad \text{definición de multiplicación de matrices}$$

$$= \sum_k a_{ik} \left(\sum_r b_{kr}c_{rj} \right) \quad \text{definición de multiplicación de matrices}$$

$$= \sum_k \sum_r a_{ik}b_{kr}c_{rj} \quad \text{asociatividad de la multiplicación en los números reales} \quad (1)$$

Por otro lado,

$$\{(AB)C\}_{ij} = \sum_r (AB)_{ir}c_{rj} \quad \text{definición de multiplicación de matrices}$$

$$= \sum_r \left(\sum_k a_{ik}b_{kr} \right) c_{rj} \quad \text{definición de multiplicación de matrices}$$

$$= \sum_r \sum_k a_{ik}b_{kr}c_{rj} \quad \text{asociatividad de la multiplicación en los números reales.} \quad (2)$$

Por lo tanto, comparando los lados derechos de (1) y (2), se demuestra que $(AB)C=A(BC)$. \square

DEFINICION 2.15. La matriz identidad $n \times n$, indicada por I_n , es una matriz cuadrada cuyos elementos sobre la diagonal principal son todos 1 y los elementos fuera de la diagonal principal son todos cero; ie.

$$I_n = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \left. \vphantom{\begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}} \right\} \begin{array}{l} n \text{ renglones} \\ n \text{ columnas} \end{array}$$

NOTA: La matriz identidad I_n , también se puede definir en términos de la delta de Kronecker, la cual se define a continuación. La delta de Kronecker, indicada por δ_{ij} , se define por

$$\delta_{ij} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$$

La matriz identidad I_n se define en términos de δ_{ij} , por

$$I_n = [\delta_{ij}]_{n \times n} = \begin{bmatrix} \delta_{11} & \delta_{12} & \dots & \delta_{1n} \\ \delta_{21} & \delta_{22} & \dots & \delta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{n1} & \delta_{n2} & \dots & \delta_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

TEOREMA 2.16. Sea $A = (a_{ij})_{m \times n}$. Si I_n y I_m son matrices identidad $n \times n$ y $m \times m$ respectivamente, entonces

- i) $I_n A = A$
- ii) $A I_m = A$

DEMOSTRACION:

i) Si $I_n A = [C_{ij}] \Rightarrow C_{ij} = \sum_{k=1}^n \delta_{ik} a_{kj}$ por definición de multiplicación de matrices.

$$\Rightarrow C_{ij} = \delta_{i1} a_{1j} + \dots + \delta_{i, i-1} a_{i-1, j} + \delta_{ii} a_{ij} + \dots + \delta_{i, i+1} a_{i+1, j} + \dots + \delta_{in} a_{nj}$$

$$\Rightarrow C_{ij} = a_{ij}$$

$$\Rightarrow I_n A = [C_{ij}] = [a_{ij}] = A \quad \square$$

ii) Es similar.

EJEMPLO. Si $A = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 2 \end{bmatrix}$ entonces

$$I_3 A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 4 & 2 \end{bmatrix} = A$$

$$A I_2 = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 4 & 2 \end{bmatrix} = A$$

DEFINICION 2.17. Sea $A = [a_{ij}]_{mn}$. La transpuesta de A , indicada por A^t , es una matriz de elementos b_{ij} definida por

$$b_{ij} = a_{ji} \quad \forall i = 1, 2, \dots, m, j = 1, 2, \dots, n.$$

ie, si

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

entonces la transpuesta de A , se define por

$$A^t = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \dots & \dots & \dots & \dots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix}$$

NOTA: Dada una matriz A , la transpuesta de A , se obtiene intercambiando los renglones de A para que lleguen a ser las columnas de A^t , ie.

La primera columna de A^t es el primer renglón de A

La segunda columna de A^t es el segundo renglón de A , etc.

EJEMPLOS. Si

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}; \quad B = \begin{bmatrix} 1 & 2 & 3 & 1 \end{bmatrix}; \quad C = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

entonces

$$A^t = \begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{bmatrix}; \quad B^t = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix}; \quad C^t = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

PROPOSICION 2.18

- i) $(A_{i.})^t = (A^t)_{.i}$
 ii) $(A_{.j})^t = (A^t)_{j.}$

DEMOSTRACION

- i) Debemos demostrar que el lado izquierdo de la igualdad i), (L I I), es igual al lado derecho de la igualdad i), L D I,

$$\begin{aligned}
 \text{L I I} = (A_{i.})^t &= \begin{bmatrix} a_{i1} & a_{i2} & \dots & a_{in} \end{bmatrix}^t = \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{in} \end{bmatrix} \\
 \text{L D I} = (A^t)_{.i} &= \begin{pmatrix} \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mj} & \dots & a_{mn} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} a_{11} & a_{21} & \dots & a_{j1} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{j2} & \dots & a_{m2} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ a_{1j} & a_{2j} & \dots & a_{jj} & \dots & a_{mj} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{jn} & \dots & a_{mn} \end{bmatrix} \end{pmatrix}_{.i} = \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{in} \end{bmatrix}
 \end{aligned}$$

Por lo tanto, $\text{L I I} = (A_{i.})^t = (A^t)_{.i} = \text{L D I}$, \square

EJEMPLO. Si $A = \begin{bmatrix} -1 & -2 & -3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ encuentre el

primer renglón de la transpuesta de A y la tercera columna de A^t .

SOLUCION

$$(A^t)_{j.} = (A_{.j})^t \quad \text{por PROPOSICION 2.18 ii)}$$

$$= \left(\begin{bmatrix} -1 \\ 4 \\ 7 \end{bmatrix} \right)^t = (-1 \quad 4 \quad 7)$$

$$\begin{aligned} (A^t)_3 &= (A_3)^t && \text{por PROPOSICION 2.18 i)} \\ &= (7 \quad 8 \quad 9)^t = \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} \end{aligned}$$

TEOREMA 2.19 (PROPIEDADES DE LA MATRIZ TRANSPUESTA)

- i) $(A^t)^t = A$
- ii) $(A + B)^t = A^t + B^t$
- iii) $(kA)^t = kA^t$
- iv) $(AB)^t = B^t A^t$

DEMOSTRACION:

i) Sea $B = [b_{ij}] = A^t$ y sea $C = [c_{ij}] = B^t = (A^t)^t$, debemos demostrar que $C = A$.

Demostración:

$$C = B^t \implies c_{ij} \stackrel{d}{=} b_{ji} \quad (1) \quad \text{por definición de transpuesta}$$

Por otro lado,

$$\begin{aligned} B = A^t &\implies b_{ij} \stackrel{d}{=} a_{ji} && (2) \quad \text{por definición de transpuesta} \\ &\implies b_{ji} = a_{ij} \end{aligned}$$

$$\begin{aligned} (2) \text{ en } (1) &\implies c_{ij} = b_{ji} = a_{ij} \\ &\implies c_{ij} = a_{ij} \quad \forall i = 1, 2, \dots, m, j = 1, 2, \dots, n \\ &\implies C = A \\ &\implies (A^t)^t = A \quad \square \end{aligned}$$

2.2 PARTICION DE MATRICES Y OPERACIONES CON MATRICES PARTICIONADAS.

DEFINICION. 2.2.1 Sea A $m \times n$. Se dice que la matriz A es una matriz particionada de acuerdo a un criterio dado, si la matriz ha sido dividida por rayas verticales y horizontales de acuerdo a dicho criterio. Si B es una notación para indicar el criterio con el cual la matriz A ha sido dividida, entonces la matriz particionada A (o la partición de A según B) se indicará por A_B .

EJEMPLO Sea

$$A = \begin{bmatrix} 1 & 2 & 1 & 3 \\ 2 & 3 & 2 & 1 \\ 1 & 4 & 2 & 1 \end{bmatrix}$$

Si el criterio B para particionar A consiste en dividir A por una raya vertical entre la primera y segunda columna y por una raya horizontal entre el segundo y el tercer renglón, entonces:

$$A_B = \begin{bmatrix} 1 & | & 2 & 1 & 3 \\ 2 & | & 3 & 2 & 1 \\ \hline 1 & | & 4 & 2 & 1 \end{bmatrix}$$

Otras particiones de A generadas por otros criterios de partición podrían ser:

$$A_D = \begin{bmatrix} 1 & 2 & 1 & 3 \\ 2 & 3 & 2 & 1 \\ \hline 1 & 4 & 2 & 1 \end{bmatrix}; \quad A_E = \begin{bmatrix} 1 & | & 2 & | & 1 & | & 3 \\ 2 & | & 3 & | & 2 & | & 1 \\ \hline 1 & | & 4 & | & 2 & | & 1 \end{bmatrix}$$

DEFINICION. 2.2.2 A las matrices que se generan por la --
partición de una matriz con un criterio B dado se les llama submatrices de A generadas por la partición de A según el criterio B

DEFINICION. 2.2.3 Sea A una matriz y B una partición sobre A . Las submatrices de A generadas por la partición B , se llaman los elementos de A_B .

NOTA. Una matriz particionada A_B se puede representar a través de sus elementos (ie. sus submatrices).

EJEMPLO: Si

$$A = \begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{bmatrix} \quad \text{y} \quad A_B = \begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{bmatrix}$$

entonces las submatrices de A generadas por B , son:

$$A_B^{11} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}; \quad A_B^{12} = \begin{bmatrix} 5 & 9 \\ 6 & 10 \\ 7 & 11 \end{bmatrix}; \quad A_B^{21} = [4]; \quad A_B^{22} = [8 \quad 12]$$

y A_B puede ser representada en términos de las submatrices en la siguiente forma:

$$A_B = \begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{bmatrix} = \begin{bmatrix} A_B^{11} & A_B^{12} \\ A_B^{21} & A_B^{22} \end{bmatrix}$$

DEFINICION. 2.2.4 Si Γ es una matriz particionada, entonces el símbolo $\{\Gamma\}$ indica la matriz obtenida de Γ eliminando sus particiones. Por lo tanto,

$$\{A_B\} = A$$

DEFINICION. 2.2.5 Sea A $m \times n$ y B $m \times n$. Sea S el mismo criterio de partici3n para A y B , ie.

$$A_S = \begin{bmatrix} 11 & 12 & \dots & 1q \\ A_S & A_S & \dots & A_S \\ 21 & 22 & \dots & 2q \\ A_S & A_S & \dots & A_S \\ \vdots & \vdots & \vdots & \vdots \\ p1 & p2 & \dots & pq \\ A_S & A_S & \dots & A_S \end{bmatrix} \quad B = \begin{bmatrix} 11 & 12 & \dots & 1q \\ B_S & B_S & \dots & B_S \\ 21 & 22 & \dots & 2q \\ B_S & B_S & \dots & B_S \\ \vdots & \vdots & \vdots & \vdots \\ p1 & p2 & \dots & pq \\ B_S & B_S & \dots & B_S \end{bmatrix}$$

La adici3n de las matrices particionadas A_S y B_S se define por

$$A_S + B_S = \begin{bmatrix} 11 & 11 & 12 & 12 & \dots & 1q & 1q \\ A_S + B_S & A_S + B_S & \dots & A_S + B_S & \dots & A_S + B_S & A_S + B_S \\ 21 & 21 & 22 & 22 & \dots & 2q & 2q \\ A_S + B_S & A_S + B_S & \dots & A_S + B_S & \dots & A_S + B_S & A_S + B_S \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p1 & p2 & p2 & p2 & \dots & pq & pq \\ A_S + B_S & A_S + B_S & \dots & A_S + B_S & \dots & A_S + B_S & A_S + B_S \end{bmatrix}$$

EJEMPLO Si

$$A_S = \left[\begin{array}{cc|cc} 2 & 1 & 4 & 0 \\ 0 & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 1 \end{array} \right] \quad B_S = \left[\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 4 & 2 & 2 & 1 \\ \hline 0 & 1 & 0 & 0 \end{array} \right]$$

entonces

$$\begin{aligned}
 B + B_B &= \begin{bmatrix} \begin{bmatrix} 2 & 1 \end{bmatrix} & + & \begin{bmatrix} 0 & 0 \end{bmatrix} & \begin{bmatrix} 4 & 0 \end{bmatrix} & + & \begin{bmatrix} 1 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & + & \begin{bmatrix} 4 & 2 \end{bmatrix} & \begin{bmatrix} 1 & 2 \end{bmatrix} & + & \begin{bmatrix} 2 & 1 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & + & \begin{bmatrix} 0 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 \end{bmatrix} & + & \begin{bmatrix} 0 & 0 \end{bmatrix} \end{bmatrix} \\
 &= \begin{bmatrix} \begin{bmatrix} 2 & 1 \end{bmatrix} & \begin{bmatrix} 5 & 0 \end{bmatrix} \\ \begin{bmatrix} 4 & 2 \end{bmatrix} & \begin{bmatrix} 3 & 3 \end{bmatrix} \\ \begin{bmatrix} 0 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 2 & 1 & | & 5 & 0 \\ 4 & 2 & | & 3 & 3 \\ 0 & 1 & | & 1 & 1 \end{bmatrix}
 \end{aligned}$$

TEOREMA 2.2.6 (PROPIEDADES DE LA ADICION DE MATRICES - PARTICIONADAS).

- i) $A_B + B_B = (A + B)_B$
 ii) $A_B + B_B = B_B + A_B$
 iii) $A_B + (B_B + C_B) = (A_B + B_B) + C_B$

DEFINICION. 2.2.7 Sea A $m \times n$ y B $n \times t$. Sean B y D particiones aplicadas a A y B respectivamente, ie.

$$A_B = \begin{bmatrix} 11 & 12 & \dots & 1q \\ A_B & A_B & \dots & A_B \\ 21 & 22 & \dots & 2q \\ A_B & A_B & \dots & A_B \\ \vdots & \vdots & \vdots & \vdots \\ p1 & p2 & \dots & pq \\ A_B & A_B & \dots & A_B \end{bmatrix}; B_D = \begin{bmatrix} 11 & 12 & \dots & 1s \\ B_D & B_D & \dots & B_D \\ 21 & 22 & \dots & 2s \\ B_D & B_D & \dots & B_D \\ \vdots & \vdots & \vdots & \vdots \\ r1 & r2 & \dots & rs \\ B_D & B_D & \dots & B_D \end{bmatrix}$$

Si $q = r$ y si cada producto

$A_B^{ik} \cdot B_D^{kj}$ ($i = 1, \dots, p$; $j = 1, \dots, s$; $k = 1, \dots, q$)
 está definido, entonces el producto $A_B \cdot B_D$ se define como una matriz particionada de p renglones y s columnas

cuyo elemento (i, j) es

$$\sum_{k=1}^q A^{ik} B^{kj} \quad (i = 1, \dots, p; j = 1, \dots, s)$$

ie

$$A_B B_D = \begin{bmatrix} 11 & 12 & \dots & 1q \\ A_B & A_B \dots & A_B & \\ \vdots & \vdots & \vdots & \vdots \\ 21 & 22 & \dots & 2q \\ A_B & A_B \dots & A_B & \\ \vdots & \vdots & \vdots & \vdots \\ p1 & p2 & \dots & pq \\ A_B & A_B \dots & A_B & \end{bmatrix} \begin{bmatrix} 11 & 12 & \dots & 1s \\ B_D & B_D \dots & B_D & \\ \vdots & \vdots & \vdots & \vdots \\ 21 & 22 & \dots & 2s \\ B_D & B_D \dots & B_D & \\ \vdots & \vdots & \vdots & \vdots \\ r1 & r2 & \dots & rs \\ B_D & B_D \dots & B_D & \end{bmatrix}$$

$$= \begin{bmatrix} 11 & 11 + \dots + 1q & r1 & 11 & 12 + \dots + 1q & r2 & 11 & 1s + \dots + 1q & rs \\ (A_B & B_D + \dots + A_B & B_D) & (A_B & B_D + \dots + A_B & B_D) & (A_B & B_D + \dots + A_B & B_D) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 21 & 11 + \dots + 1q & r1 & 21 & 12 + \dots + 2q & r2 & 21 & 1s + \dots + 2q & rs \\ (A_B & B_D + \dots + A_B & B_D) & (A_B & B_D + \dots + A_B & B_D) & (A_B & B_D + \dots + A_B & B_D) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p1 & 11 + \dots + pq & r1 & p1 & 12 + \dots + pq & r2 & p1 & 1s + \dots + pq & rs \\ (A_B & B_D + \dots + A_B & B_D) & (A_B & B_D + \dots + A_B & B_D) & (A_B & B_D + \dots + A_B & B_D) \end{bmatrix}$$

EJEMPLO. Si

$$A_B = \begin{bmatrix} 11 & 12 \\ A_B & A_B \\ 21 & 22 \\ A_B & A_B \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 1 & 1 & 2 & 1 \\ 2 & 2 & 1 & 3 \end{bmatrix}; B_D = \begin{bmatrix} 11 \\ B_D \\ 12 \\ B_D \\ D \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 2 & 1 \end{bmatrix}$$

entonces

$$A_B B_D = \begin{bmatrix} 11 & 11 & 12 & 21 \\ A_B & B_D & A_B & B_D \\ 21 & 11 & 22 & 21 \\ A_B & B_D & A_B & B_D \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 6 & 7 \\ 1 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 4 \\ 8 \\ 1 \\ 3 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 6 & 6 \\ 16 & 16 \end{bmatrix} + \begin{bmatrix} 8 & 4 \\ 16 & 8 \end{bmatrix} = \begin{bmatrix} 14 & 10 \\ 32 & 24 \end{bmatrix} = \begin{bmatrix} 14 & 10 \\ 32 & 24 \\ 6 & 5 \\ 11 & 8 \end{bmatrix}$$

EJEMPLO

$$A_B \quad B_D = \left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 \end{array} \right] \quad \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right]$$

$$= \left[\begin{array}{cc|cc|cc} \left[\begin{array}{cc} 1 & 0 \\ 0 & 2 \end{array} \right] & \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right] & & & \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array} \right] & & & \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array} \right] & & & & \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right] & \left[\begin{array}{cc} 1 & 0 \\ 0 & 3 \end{array} \right] & & \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array} \right] & & & & \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array} \right] & & & \end{array} \right]$$

$$= \left[\begin{array}{cc|cc} \left[\begin{array}{cc} 1 & 0 \\ 0 & 2 \end{array} \right] & \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array} \right] & & \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array} \right] & \left[\begin{array}{cc} 1 & 0 \\ 0 & 3 \end{array} \right] & & \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array} \right] & \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array} \right] & & \end{array} \right] = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right]$$

2.4 INVERSA DE UNA MATRIZ.

DEFINICIONES Y PROPIEDADES.

DEFINICION 2.4.1 Sea A una matriz cuadrada $n \times n$. Si existe una matriz B $n \times n$ tal que

$$AB = I_{nn}$$
$$\text{y } BA = I_{nn}$$

entonces B es llamada la inversa de A . A la matriz B se le indicará por A^{-1} , ie, la inversa de A , indicada por A^{-1} , es una matriz tal que

$$AA^{-1} = A^{-1}A = I_{nn}$$

NOTA. Si para una matriz A no es posible encontrar una matriz A^{-1} que satisfaga la definición anterior, entonces se dice que A no tiene inversa o que su inversa no existe.

TEOREMA 2.4.2 (PROPIEDADES DE LA MATRIZ INVERSA). Sean A y B matrices $n \times n$ y sean A^{-1} y B^{-1} sus respectivas inversas. Se afirma que :

i) $(AB)^{-1} = B^{-1}A^{-1}$

ii) $(A^{-1})^{-1} = A$

iii) $(A^t)^{-1} = (A^{-1})^t$

DEMOSTRACION DE i). Para que $B^{-1}A^{-1}$ sea la inversa de AB , debe satisfacer que

$$(B^{-1}A^{-1})(AB) = I_{nn}$$
$$\text{y } (AB)(B^{-1}A^{-1}) = I_{nn}$$

Las demostraciones de estas dos condiciones se presenta en las siguientes demandas

DEMANDA 1. $(B^{-1}A^{-1})(AB) = I_{nn}$

DEMOSTRACION

$$(B^{-1}A^{-1})(AB) = (B^{-1}A^{-1}A)B \quad \text{por asociatividad en la multiplicación de matrices.}$$
$$= B^{-1}(A^{-1}A)B \quad \text{por asociatividad de en la multiplicación de matrices.}$$
$$= B^{-1}I_{nn}B$$
$$= B^{-1}B = I_{nn} \quad \square$$

DEMANDA 2. $(AB)(B^{-1}A^{-1}) = I_{nn}$

DEMOSTRACION. Es similar, a la anterior, ie.

$$(AB)(B^{-1}A^{-1}) = (ABB^{-1})A^{-1} = A(BB^{-1})A^{-1} = AI_{nn}A^{-1}$$
$$= AA^{-1} = I_{nn} \quad \square$$

DEMOSTRACION DE ii) Debemos demostrar que

$$A^{-1}(A^{-1})^{-1} = I_{nn}$$
$$\text{y } (A^{-1})^{-1}A^{-1} = I_{nn}$$

La demostración es trivial, ya que conocemos que para cualquier matriz B con inversa, se debe satisfacer que

$$BB^{-1} = B^{-1}B = I_{nn}$$

Por lo tanto, si hacemos $B = A^{-1}$,

$$(A^{-1})(A^{-1})^{-1} = (A^{-1})^{-1}(A^{-1}) = I_{nn} \quad \square$$

DEMOSTRACION DE iii).

$$AA^{-1} = A^{-1}A = I_{nn} \rightarrow (AA^{-1})^t = (A^{-1}A)^t = I_{nn}^t$$

$$\rightarrow (A^{-1})^t A^t = A^t (A^{-1})^t = I_{nn} \quad \text{Teorema 2.19 iv) (pág. 13).}$$

$$\text{Si } B = (A^{-1})^t \rightarrow B A^t = A^t B = I_{nn}$$

Por lo tanto la inversa de A^t es B por definición de inversa. Pero $B = (A^{-1})^t$ entonces la inversa de A^t es $(A^{-1})^t$, ie $(A^t)^{-1} = (A^{-1})^t \quad \square$.

TEOREMA 2.4.3 (INVERSA DE UNA MATRIZ 2 X 2).

Si

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \text{ entonces } A^{-1} = \frac{1}{|A|} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$$

$$\text{donde } |A| = a_{11} a_{22} - a_{12} a_{21}$$

DEMOSTRACION. Suponga que

$$B = \begin{bmatrix} w & x \\ y & z \end{bmatrix}$$

es la inversa de A , entonces $AB = I$, ó sea

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} a_{11}w + a_{12}y & a_{11}x + a_{12}z \\ a_{21}w + a_{22}y & a_{21}x + a_{22}z \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$a_{11}w + a_{12}y = 1$$

$$a_{21}w + a_{22}y = 0$$

$$a_{11}x + a_{12}z = 0$$

$$a_{21}x + a_{22}z = 1$$

Resolviendo en las variables w, x, y, z , se tiene que

$$B = \begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} \frac{a_{22}}{a_{22} a_{11} - a_{12} a_{21}} & \frac{-a_{12}}{a_{22} a_{11} - a_{12} a_{21}} \\ \frac{-a_{21}}{a_{22} a_{11} - a_{12} a_{21}} & \frac{a_{11}}{a_{22} a_{11} - a_{12} a_{21}} \end{bmatrix}$$

$$B = \frac{1}{a_{22} a_{11} - a_{12} a_{21}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} = \frac{1}{|A|} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \quad \square$$

EJEMPLO. Si

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix} \text{ entonces } A^{-1} = \frac{1}{-5} \begin{bmatrix} 3 & -4 \\ -2 & 1 \end{bmatrix} \quad A^{-1} = \begin{bmatrix} -\frac{3}{5} & \frac{4}{5} \\ \frac{2}{5} & -\frac{1}{5} \end{bmatrix}$$

TEOREMA 2.4.4 (INVERSION DE MATRICES POR PARTICIONES). Si A es una matriz n x n, particionada de acuerdo al siguiente criterio

$$A = \begin{bmatrix} a_{11} \dots a_{1p} & a_{1,p+1} \dots a_{1n} \\ \vdots & \vdots \\ a_{p1} & a_{pp} & a_{p,p+1} \dots a_{pn} \\ \hline a_{p+1,1} \dots a_{p+1,p} & a_{p+1,p+1} \dots a_{p+1,n} \\ \vdots & \vdots \\ a_{n1} \dots a_{n,p} & a_{n,p+1} \dots a_{n,n} \end{bmatrix} = \begin{bmatrix} M & R \\ L & N \end{bmatrix}; \quad \begin{array}{l} M \text{ es } p \times p \\ L \text{ es } q \times p \\ R \text{ es } p \times q \\ N \text{ es } q \times q \\ p + q = n \end{array}$$

y N^{-1} existe entonces

$$A^{-1} = \begin{bmatrix} \mu & \rho \\ \lambda & \nu \end{bmatrix} \quad \text{donde} \quad \begin{array}{ll} \mu = (M - R N^{-1} L)^{-1} & \text{es } p \times p \\ \lambda = - N^{-1} L \mu & \text{es } q \times p \\ \rho = - \mu R N^{-1} & \text{es } p \times q \\ \nu = N^{-1} - N^{-1} L \rho & \text{es } q \times q \end{array}$$

DEMOSTRACION

Suponga que la inversa de A es una matriz particionada de la forma

$$A^{-1} = \begin{bmatrix} \mu & \rho \\ \lambda & \nu \end{bmatrix}$$

donde μ , ρ , λ y ν son submatrices de A^{-1} que tienen las siguientes dimensiones.

- μ es $p \times p$
- λ es $q \times p$
- ρ es $p \times q$
- ν es $q \times q$

Por definición de inversa, se tiene que

$$A A^{-1} = I_{nn}$$

Esta igualdad expresada en forma particionada presenta la siguiente forma

$$\begin{bmatrix} M & R \\ L & N \end{bmatrix} \begin{bmatrix} \mu & \rho \\ \lambda & \nu \end{bmatrix} = \begin{bmatrix} I_{pp} & | & O_{pq} \\ \hline O_{qp} & | & I_{qq} \end{bmatrix}$$

Realizando el producto de las matrices particionadas del lado izquierdo de la igualdad anterior se tiene que

$$\begin{bmatrix} M\mu + R\lambda & M\rho + R\nu \\ L\mu + N\lambda & L\rho + N\nu \end{bmatrix} = \begin{bmatrix} I_{pp} & | & O_{p \times q} \\ \hline O_{q \times p} & | & I_{qq} \end{bmatrix}$$

$$M\mu + R\lambda = I_{pp} \quad (2.4.1)$$

$$L\mu + N\lambda = O_{qp} \quad (2.4.2)$$

$$M\rho + R\nu = O_{pq} \quad (2.4.3)$$

$$L\rho + N\nu = I_{qq} \quad (2.4.4)$$

$$(2.4.2) \rightarrow \lambda = -N^{-1}L\mu \quad \square \quad (2.4.5)$$

$$(2.4.5) \text{ en } (2.4.1) \rightarrow M\mu + R(-N^{-1}L\mu) = I_{pp}$$

$$\rightarrow (M - RN^{-1}L)\mu = I_{pp}$$

$$\rightarrow \mu = (M - RN^{-1}L)^{-1} I_{pp}$$

$$\rightarrow \mu = (M - RN^{-1}L)^{-1} \quad \square \quad (2.4.6)$$

$$(2.4.4) \rightarrow \nu = N^{-1} - N^{-1}L\rho \quad \square \quad (2.4.7)$$

$$(2.4.7) \text{ en } (2.4.3) \rightarrow M\rho + R[N^{-1} - N^{-1}L\rho] = O_{pq}$$

$$\rightarrow [M - RN^{-1}L]\rho = -RN^{-1}$$

$$\rightarrow \rho = -[M - RN^{-1}L]^{-1} RN^{-1}$$

$$\rightarrow \rho = -\mu RN^{-1} \quad \square$$

EJEMPLO. Si $A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 6 \end{bmatrix}$

encuentre su inversa por particiones.

SOLUCION. Una posible partición de A es

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 6 \end{bmatrix} = \begin{bmatrix} M & R \\ L & N \end{bmatrix}$$

Por lo tanto,

$$A^{-1} = \begin{bmatrix} \mu & \rho \\ \lambda & \nu \end{bmatrix}$$

$$N = \begin{bmatrix} 3 & 4 \\ 4 & 6 \end{bmatrix} \rightarrow N^{-1} = \frac{1}{2} \begin{bmatrix} 6 & -4 \\ -4 & 3 \end{bmatrix} = \begin{bmatrix} 3 & -2 \\ -2 & 3/2 \end{bmatrix}$$

Ya que N^{-1} existe entonces la partición elegida es apropiada y podemos aplicar el método por particiones de acuerdo al teorema anterior.

$$\mu = (M - RN^{-1}L)^{-1} = (1 - \begin{bmatrix} 2 & 3 \end{bmatrix} \begin{bmatrix} 3 & -2 \\ -2 & 3/2 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix})^{-1}$$

$$\mu = (1 - \begin{bmatrix} 0 & 1/2 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix})^{-1} = (1 - 3/2)^{-1} = (-1/2)^{-1}$$

$$\mu = -2$$

$$\lambda = -N^{-1}L\mu = -\begin{bmatrix} 3 & -2 \\ -2 & 3/2 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} (-2) = 2 \begin{bmatrix} 0 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\rho = -\mu RN^{-1} = -(-2) \begin{bmatrix} 2 & 3 \end{bmatrix} \begin{bmatrix} 3 & -2 \\ -2 & 3/2 \end{bmatrix} = 2 \begin{bmatrix} 0 & 1/2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$\nu = N^{-1} - N^{-1}L\rho = \begin{bmatrix} 3 & -2 \\ -2 & 3/2 \end{bmatrix} - \begin{bmatrix} 3 & -2 \\ -2 & 3/2 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & -2 \\ -2 & 3/2 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & 1/2 \end{bmatrix}$$

$$\nu = \begin{bmatrix} 3 & -2 \\ -2 & 1 \end{bmatrix}$$

Por lo tanto

$$A^{-1} = \begin{bmatrix} \mu & \rho \\ \lambda & \nu \end{bmatrix} = \begin{bmatrix} -2 & 0 & 1 \\ 0 & 3 & -2 \\ 1 & -2 & 1 \end{bmatrix} = \begin{bmatrix} -2 & 0 & 1 \\ 0 & 3 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Otra posible solución es eligiendo la siguiente partición

$$A = \left[\begin{array}{cc|c} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 6 \end{array} \right] = \begin{bmatrix} M & R \\ L & N \end{bmatrix}$$

$$N = 6 \rightarrow N^{-1} = 1/6$$

Ya que N^{-1} existe, entonces de acuerdo al teorema anterior se tiene

$$\mu = (M - RN^{-1}L)^{-1} = \left(\begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} - \begin{bmatrix} 3 \\ 4 \end{bmatrix} (1/6) \begin{bmatrix} 3 & 4 \end{bmatrix} \right)^{-1}$$

$$\mu = \begin{bmatrix} -1/2 & 0 \\ 0 & 1/3 \end{bmatrix}^{-1} = \begin{bmatrix} -2 & 0 \\ 0 & 3 \end{bmatrix}$$

$$\lambda = -N^{-1}L\mu = -\frac{1}{6} \begin{bmatrix} 3 & 4 \end{bmatrix} \begin{bmatrix} -2 & 0 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & -2 \end{bmatrix}$$

2. ALGEBRA MATRICIAL

2.1 Introducción

Una matriz es un arreglo rectangular de elementos distribuidos en "m" renglones y "n" columnas, si a la matriz se le denota por la letra A, entonces al elemento del "i-ésimo" renglón y de la "j-ésima" columna se le representará por el símbolo a_{ij} . Generalmente una matriz se representa mediante paréntesis cuadrados como se muestra a continuación:

$$\underline{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad (2.1)$$

Los elementos que componen una matriz pueden ser de diversos tipos: números reales, números complejos, funciones en el dominio del tiempo, etc..

Al ser una matriz un arreglo ordenado de elementos, permite que al aplicar cierta metodología a dicho arreglo se obtenga una serie de resultados que responden a las interrogantes por las que se originó el arreglo; entre algunos de los procesos en los que se utilizan arreglos matriciales se tiene: jerarquización de actividades, almacenamiento de datos, inventarios, representación de sistemas dinámicos, sistemas de ecuaciones, etc..

Existen ciertas distribuciones típicas de los elementos de una matriz y de acuerdo a ellas se clasifica a las matrices en diferentes tipos, entre los que se tienen:

Matriz Cuadrada

Es una matriz en la que el número de renglones es igual al número de columnas, es decir, $m=n$. Por ejemplo:

$$\underline{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad (2.2)$$

Matriz Nula

Es una matriz de orden cualquiera, en la que todos los elementos son nulos; por ejemplo:

$$\underline{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (2.3)$$

Se acostumbra denotarla por el símbolo $\underline{0}$.

Matriz Identidad

Es una matriz cuadrada en la cual los elementos de la diagonal principal son unitarios y el resto son nulos, es decir:

$$\delta_{ij} = \begin{cases} 0 & , i \neq j \\ 1 & , i = j \end{cases}$$

Se suele denotarla como \underline{I}_n donde "n" indica el orden de la matriz y al símbolo δ_{ij} se le conoce como delta de Kronecker. Por ejemplo:

$$\underline{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Matriz Diagonal

Es una matriz cuadrada en la que los elementos que no pertenecen a la diagonal principal son nulos, es decir:

$$a_{ij} = 0 \quad \text{si } i \neq j \quad (2.5)$$

Un ejemplo de este tipo de matriz sería:

$$\underline{A} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & \text{sent} \end{bmatrix} \quad (2.6)$$

Matriz Transpuesta

Es una matriz cuadrada que se obtiene a partir de una matriz dada \underline{A} intercambiando renglones con columnas. Se le denota con el símbolo \underline{A}^T y se cumple que:

$$a_{ij}^T = a_{ji} \quad (2.7)$$

Matriz Simétrica

Es una matriz cuadrada \underline{B} para la que se cumple:

$$\underline{B} = \underline{B}^T \quad (2.8)$$

o equivalentemente:

$$b_{ij} = b_{ji} \quad (2.9)$$

Entre las matrices se definen dos operaciones básicas:

- suma o resta de matrices,
- multiplicación matricial.

2.2 Suma Matricial

2.2.1 Objeto

Obtener la suma de dos matrices de igual orden, o sea:

$$\underline{C} = \underline{A} + \underline{B} \quad (2.10)$$

2.2.2 Método

Para poder efectuar la suma de dos matrices ($\underline{A} + \underline{B}$) se requiere que sean conformables para la suma, lo cual implica que el orden de las dos matrices es igual. En otras palabras:

si \underline{A} es de orden $(m \times n)$

y \underline{B} es de orden $(r \times s)$

la suma $\underline{C} = \underline{A} + \underline{B}$ es posible solo si $m=r$ y $n=s$.

Los elementos de la matriz suma están dados por la siguiente relación:

$$c_{ij} = a_{ij} + b_{ij} \quad (2.11)$$

El restar dos matrices equivale a cambiar el signo de todos los elementos de una de ellas y efectuar la suma, esto es:

$$\underline{W} = \underline{X} - \underline{Y} = \underline{X} + (-\underline{Y}) \quad (2.12)$$

2.2.3 Descripción del Programa

a) Subrutinas requeridas:

SUBROUTINE SUMAT(A, B, C, N, M), esta subrutina efectúa la suma matricial, el programa principal lee e imprime resultados.

b) Descripción de las variables:

Para la subrutina SUMAT:

N cantidad de renglones de cada una de las matrices que se desea sumar.
 M cantidad de columnas de cada una de las matrices que se desea sumar.
 A(I, J) matriz sumando de orden NxM
 B(I, J) matriz sumando de orden NxM
 C(I, J) matriz suma

Para el programa principal:

N cantidad de renglones de las matrices
 M cantidad de columnas de las matrices
 A(I, J) matriz sumando de orden NxM
 B(I, J) matriz sumando de orden NxM
 C(I, J) matriz suma

c) Dimensiones:

La proposición DIMENSION deberá ser modificada tanto en el programa principal como en la subrutina cuando:

$N > 10$ y/o $M \geq 10$

d) Formatos para los datos de entrada:

SEC. TARJETAS	FORMATO	INFORMACION
1	(2I5)	N, M
2	(8F10.0)	A(I, J), se dan los elementos de la matriz renglón por renglón. Emplear tantas tarjetas como se requiera.
3	(8F10.0)	B(I, J), igual que para la matriz <u>A</u> .

 otros paquetes de datos (opcional)

n TARJETA EN BLANCO, al finalizar toda la información.

e) Diagrama de bloques

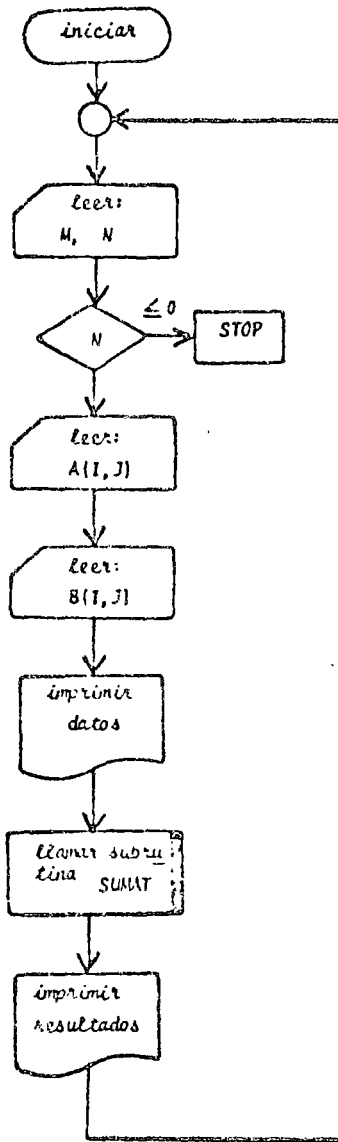


Fig. 2.1 Diagrama de bloques para el programa principal

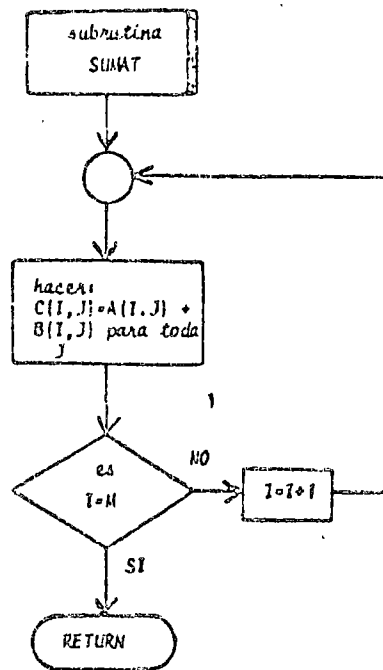


Fig.2.2 Diagrama de bloques para la subrutina SUMAT-

2.2.4 Ejemplo

En una tienda de artículos eléctricos se venden resistencias eléctricas de 1/4, 1/2 y 1 Watt de potencia en seis diferentes valores resistivos.

Si las existencias un viernes por la tarde son:

	1/4	1/2	1
100 Ω	200	380	275
150 Ω	400	250	275
1.0 K	500	175	325
1.5 K	800	225	150
10.0 K	600	380	180
15.0 K	550	250	220

y el sábado se recibe una remesa con las siguientes características:

	1/4	1/2	1
100 Ω	80	90	50
150 Ω	90	100	55
1.0 K	75	90	60
1.5 K	65	95	55
10.0 K	80	100	60
15.0 K	75	110	60

Determine las resistencias que tendrá en inventario el establecimiento el lunes por la mañana dado que ni el sábado ni el domingo hubo ventas.

* SOLUCION

TABLA 2.1 Datos para el problema del ejemplo 2.2.4

$N=6$

$M=3$

A =	200	380	275
	400	250	275
	500	175	325
	800	225	150
	600	380	180
	550	250	220

$$B = \begin{bmatrix} 80 & 90 & 50 \\ 90 & 100 & 55 \\ 75 & 90 & 60 \\ 65 & 95 & 55 \\ 80 & 100 & 60 \\ 75 & 110 & 60 \end{bmatrix}$$

TABLA 2.2 Resultados del problema del ejemplo 2.2.4

LAS MATRICES POR SUMAR SON		
MATRIZ A		
.200E+03	.380E+03	.275E+03
.400E+03	.250E+03	.275E+03
.500E+03	.175E+03	.325E+03
.800E+03	.225E+03	.150E+03
.600E+03	.380E+03	.180E+03
.550E+03	.250E+03	.220E+03
MATRIZ B		
.800E+02	.900E+02	.500E+02
.900E+02	.100E+03	.550E+02
.750E+02	.900E+02	.600E+02
.650E+02	.950E+02	.550E+02
.800E+02	.100E+03	.600E+02
.750E+02	.110E+03	.600E+02
MATRIZ SUMA		
.280E+03	.470E+03	.325E+03
.490E+03	.350E+03	.330E+03
.575E+03	.265E+03	.385E+03
.865E+03	.320E+03	.205E+03
.680E+03	.480E+03	.240E+03
.625E+03	.360E+03	.280E+03

2.3 Multiplicación Matricial

2.3.1 Objeto

Dadas dos matrices \underline{A} y \underline{B} , obtener el producto matricial \underline{C} de la forma:

$$\underline{C} = \underline{A} \times \underline{B} \quad (2.13)$$

2.3.2 Método

Para efectuar el producto entre dos matrices ($\underline{A} \times \underline{B}$) se requiere que las matrices sean conformables para la multiplicación, lo que equivale a que el número de columnas de la matriz premultiplicadora (\underline{A}) sea igual al número de renglones de la postmultiplicadora (\underline{B}), es decir:

si \underline{A} es de orden $(m \times n)$

y \underline{B} es de orden $(n \times s)$

el producto matricial \underline{AB} será posible solo si $n=n$ y el orden de la matriz producto será $(m \times s)$.

Si la matriz \underline{C} representa la matriz resultante del producto matricial \underline{AB} , entonces el elemento c_{ij} está dado por:

$$c_{ij} = \sum_{l=1}^n a_{il} b_{lj}, \quad \begin{matrix} i=1, \dots, m \\ j=1, \dots, s \end{matrix} \quad (2.14)$$

Es importante hacer notar que el producto matricial no es conmutativo, esto es:

$$\underline{A} \times \underline{B} \neq \underline{B} \times \underline{A}$$

2.3.3 Descripción del Programa

a) Subrutinas requeridas:

SUBROUTINE MULTMA(A,B,N,M,L,X), esta subrutina efectúa el producto matricial $\underline{A} \times \underline{B}$. El programa principal se emplea para la lectura de datos e impresión de resultados.

b) Descripción de las variables:

Para la subrutina MULTMA:

A(I,J) matriz premultiplicadora de orden $N \times M$

$B(I, J)$ matriz postmultiplicadora de orden $M \times L$

$X(I, J)$ matriz producto de orden $N \times L$

Para el programa principal:

$A(I, J)$ matriz premultiplicadora de orden $N \times M$

$B(I, J)$ matriz postmultiplicadora de orden $M \times L$

$X(I, J)$ matriz producto de orden $N \times L$

c) Dimensiones:

La proposición DIMENSION deberá ser modificada tanto en el programa principal como en la subrutina cuando:

$N > 10$ y/o $M > 10$ y/o $L > 10$

d) Formatos para los datos de entrada:

SEC. TARJETAS	FORMATO	INFORMACION
1	(3I5)	N, M, L
2	(8F10.0)	$A(I, J)$, los elementos de la matriz se dan renglón por renglón. Emplear la cantidad de tarjetas que sea necesaria.
3	(8F10.0)	$B(I, J)$, igual que en el caso anterior.

 otros paquetes de datos (opcional)

n TARJETA EN BLANCO, al finalizar toda la información

e) Diagrama de bloques:

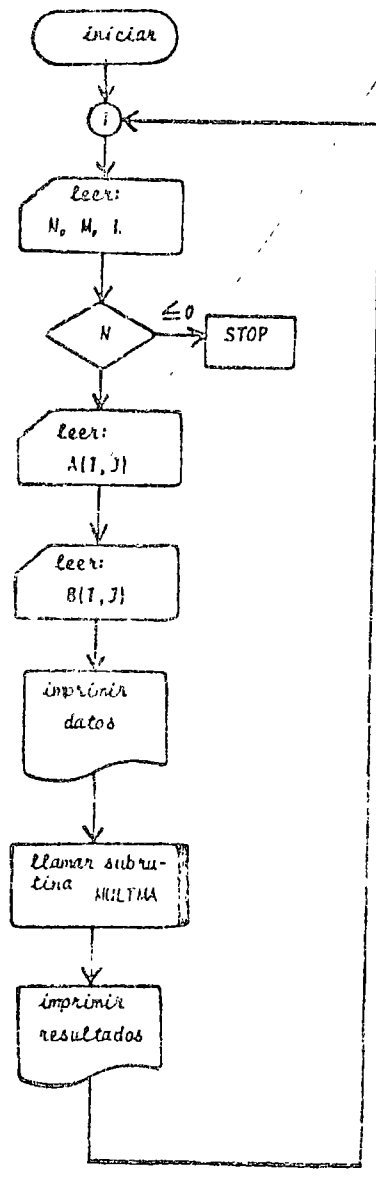


Fig. 2.5 Diagrama de bloques para el programa principal

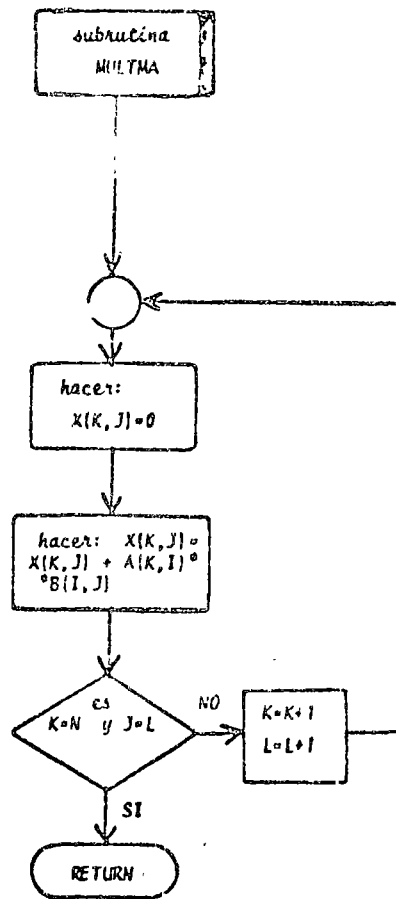


Fig. 2.6 Diagrama de bloques para la subrutina MULTMA

6) Listado:

```

C   PROGRAMA PARA EFECTUAR PRODUCTOS MATRICIALES
C   SIGNIFICAC DE LAS VARIABLES EMPLEADAS
C   A=MATRIZ PREMULTIPLICADORA DE ORDEN (N*M)
C   B=MATRIZ POSTMULTIPLICADORA DE ORDEN (M*L)
C   X=MATRIZ PRODUCTO DE ORDEN (N*L)

      DIMENSION A(10,10),B(10,10),X(10,10)
      IR=5
      J=5
C   LECTURA DE DATOS
1   READ(I,10) N,M,L
      IF(N) 2,2,3
2   CALL EXIT
3   JC 4 I,1,M
4   READ(I,11) (A(I,J),J=1,M)
      UC 5 I,1,M
5   READ(I,11) (B(I,J),J=1,L)
C   IMPRESION DE DATOS
      WRITE(I,12)
      CO 6 I,1,M
6   WRITE(I,13) (A(I,J),J=1,M)
      WRITE(I,14)
      CO 7 I,1,M
7   WRITE(I,13) (B(I,J),J=1,L)
      WRITE(I,15)
C   LLAMADA DE SUBROUTINA PARA EFECTUAR PRODUCTO MATRICIAL
      CALL MULTMA(A,B,N,M,L,X)
C   IMPRESION DE RESULTADOS
      CO 8 I,1,M
8   WRITE(I,13) (X(I,J),J=1,L)
      CO 9 I
C   FORMATS DE LECTURA E IMPRESION
10  FORMAT (3I5)
11  FORMAT (F10.0)
12  FORMAT (A(//),5X,'MATRIZ A',//)
13  FORMAT (//2X,10(E10.3,1X))
14  FORMAT (A(//),5X,'MATRIZ B',//)
15  FORMAT (A(//),5X,'MATRIZ PRODUCTO',//)
      END

```

Fig. 2.7 Listado del programa principal

```

C   SUBROUTINE MULTMA(A,N,M,L,X)
C   SUBROUTINA PARA MULTIPLICAR DOS MATRICES
C   LL SIGNIFICAC DE LAS VARIABLES EMPLEADAS ES
C   A=MATRIZ PREMULTIPLICADORA DE ORDEN (N*M)
C   B=MATRIZ POSTMULTIPLICADORA DE ORDEN (M*L)
C   X=MATRIZ PRODUCTO
C   DIMENSION A(10,10),B(10,10),X(10,10)
      CO 1 J=1,L
      CO 1 I=1,N
      X(N,J)=0.0
      CO 1 I=1,M
1   X(N,J)=X(N,J) + A(N,I)*B(I,J)
      RETURN
      END

```

Fig. 2.8 Listado de la subrutina MULTMA

2.3.4 Ejemplo

Cuatro componentes de un automóvil requieren como materia prima de hule, aluminio y acero. Las unidades que se requieren de cada material para formar una unidad de cada componente del automóvil se proporcionan a continuación:

	hule	aluminio	acero
comp. 1	8	5	3
comp. 2	3	4	5
comp. 3	20	2	4
comp. 4	1	8	10

si los costos unitarios de cada material son:

	\$
hule	25.00
aluminio	30.00
acero	40.00

Determine el costo total de cada componente debido a la materia prima de que está compuesto.

*SOLUCION

TABLA 2.3 Datos para el problema del ejemplo 2.3.4

$$N=4$$

$$M=3$$

$$L=1$$

$$\underline{A} = \begin{bmatrix} 8 & 5 & 3 \\ 3 & 4 & 5 \\ 20 & 2 & 4 \\ 1 & 8 & 10 \end{bmatrix}$$

$$\underline{B} = \begin{bmatrix} 25 \\ 30 \\ 40 \end{bmatrix}$$

TABLA 2.4 Resultados del problema del ejemplo 2.3.4

MATRIZ A

.800E+01	.500E+01	.300E+01
.300E+01	.400E+01	.500E+01
.200E+02	.300E+01	.400E+01
.100E+01	.800E+01	.100E+02

MATRIZ B

.250E+02
.350E+02
.400E+02

MATRIZ PRODUCTO

.470E+03
.395E+03
.720E+03
.665E+03

2.4 Inversión de Matrices

2.4.1 Objeto

Dada una matriz cuadrada \underline{A} obtener su matriz inversa \underline{A}^{-1} .

2.4.2 Método

La matriz inversa de una matriz cuadrada \underline{A} es otra matriz cuadrada que se representa por \underline{A}^{-1} y que cumple la siguiente propiedad si la matriz \underline{A} es de orden $(n \times n)$:

$$\underline{A} \underline{A}^{-1} = \underline{I}_n = \underline{A}^{-1} \underline{A} \quad (2.15)$$

Se define a la matriz inversa como:

$$\underline{A}^{-1} = \frac{\underline{A}^{\dagger}}{|\underline{A}|} \quad (2.16)$$

donde \underline{A}^{\dagger} se conoce como la matriz adjunta de la matriz \underline{A} y $|\underline{A}|$ representa el determinante de la matriz \underline{A} .

De la ecuación (2.16) se infiere que para que exista la inversa de una matriz se requiere que $|\underline{A}| \neq 0$, es decir, que la matriz sea no singular.

Para la obtención numérica de la matriz inversa es necesario acudir al método de Gauss-Jordan modificado. Esto se hace debido a que para obtener \underline{A}^{-1} en una computadora digital mediante la ecuación (2.16) se requiere una gran cantidad de operaciones y consecuentemente de tiempo. Para obtener la inversa de una matriz (10×10) se requieren más de 340 millones de operaciones con el método directo.

El método de Gauss-Jordan es un método de eliminación sistemática mediante el cual se transforma la matriz original \underline{A} en una matriz identidad \underline{I}_n y al mismo tiempo esta última se transforma en la matriz inversa \underline{A}^{-1} , es decir, partiendo del arreglo:

$$\left[\underline{A}_n \ ; \ \underline{I}_n \right] \quad (2.17)$$

y aplicando algunas de las siguientes transformaciones al arreglo (2.17):

- intercambio de renglones,
 - multiplicación de un renglón por un escalar $\lambda \neq 0$,
 - suma de equimúltiplos de un renglón a otro renglón.
- se llega al siguiente arreglo:

$$\left[\begin{array}{c|c} I_n & \\ \hline & A^{-1} \end{array} \right] \quad (2.18)$$

El método parte de la suposición de que A es una matriz no singular, lo cual implica que sus columnas son vectores linealmente independientes, en caso de no serlo el método lo puede detectar; en dicha situación se presenta que todos los elementos de un renglón de la matriz A o de sus matrices transformadas, son nulos.

A fin de minimizar los errores de redondeo, la eliminación de elementos se efectúa pivoteando sobre los mayores elementos que quedan en la matriz A o en las matrices obtenidas a partir de esta última por transformación; debe tenerse cuidado de no emplear como pivotes elementos de renglones que ya hayan sido utilizados como pivotes.

2.4.3 Descripción del Programa

a) Subrutinas requeridas:

SUBROUTINE MATINV(A,N, EPS, DET), obtiene la matriz inversa de la matriz A . El programa principal se emplea para la lectura de datos e impresión de resultados.

b) Descripción de las variables:

Para la subrutina MATINV:

A(I, J)	matriz de la que se buscará la inversa, durante el proceso se convierte en la matriz inversa.
N	orden de la matriz A
EPS	criterio para determinar si el determinante de la matriz es nulo
DET	parámetro que indica si el determinante de la matriz es nulo
C(I, J)	matriz identidad que se emplea para obtener la matriz inversa
MVR(I) y MVC(I)	contadores que indican cuáles renglones y cuáles columnas de la matriz A ya fueron empleados como pivotes

RAMAX	mayor elemento de la matriz <u>A</u> o de sus transformaciones que se emplea como elemento pivote
TEMP	variable de localización temporal
Para el programa principal:	
A(I,J)	matriz de la que se busca la inversa, durante el proceso se convierte en la matriz inversa
N	orden de la matriz <u>A</u>
EPS	criterio para determinar si el determinante de la matriz <u>A</u> es nulo.
DET	variable que indica si el determinante de <u>A</u> es o no nulo

c) Dimensiones:

La proposición DIMENSION del programa principal y de la subrutina deberá ser modificada cuando:

$$N > 10$$

d) Formatos para los datos de entrada:

SEC. TARJETAS	FORMATO	INFORMACION
1	(I5)	N
2	(8F10.0)	A(I,J), se proporcionan los elementos de la matriz renglón por renglón. Emplear tantas tarjetas como se requieran.

	otros paquetes de datos (opcional)	

n		TARJETA EN BLANCO, al finalizar toda la información.

e) Diagrama de bloques:

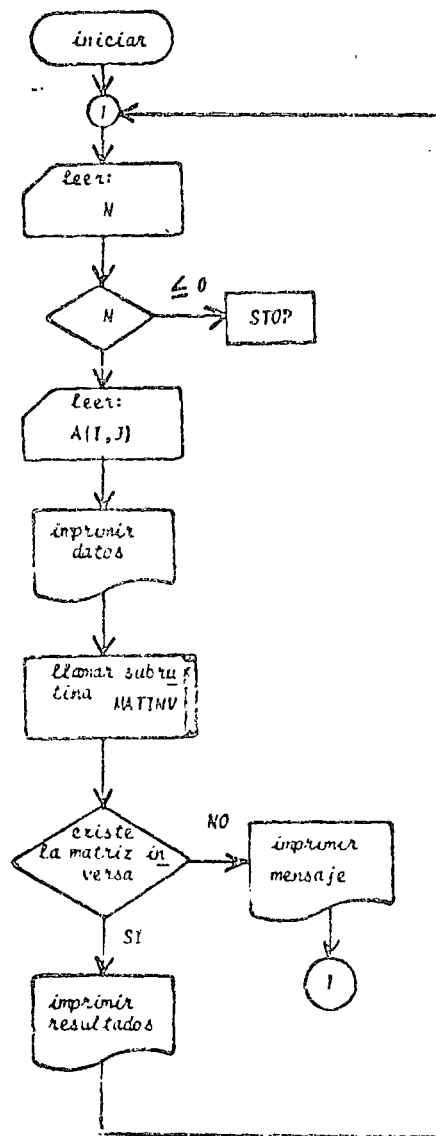


Fig. 2.9 Diagrama de bloques del programa principal

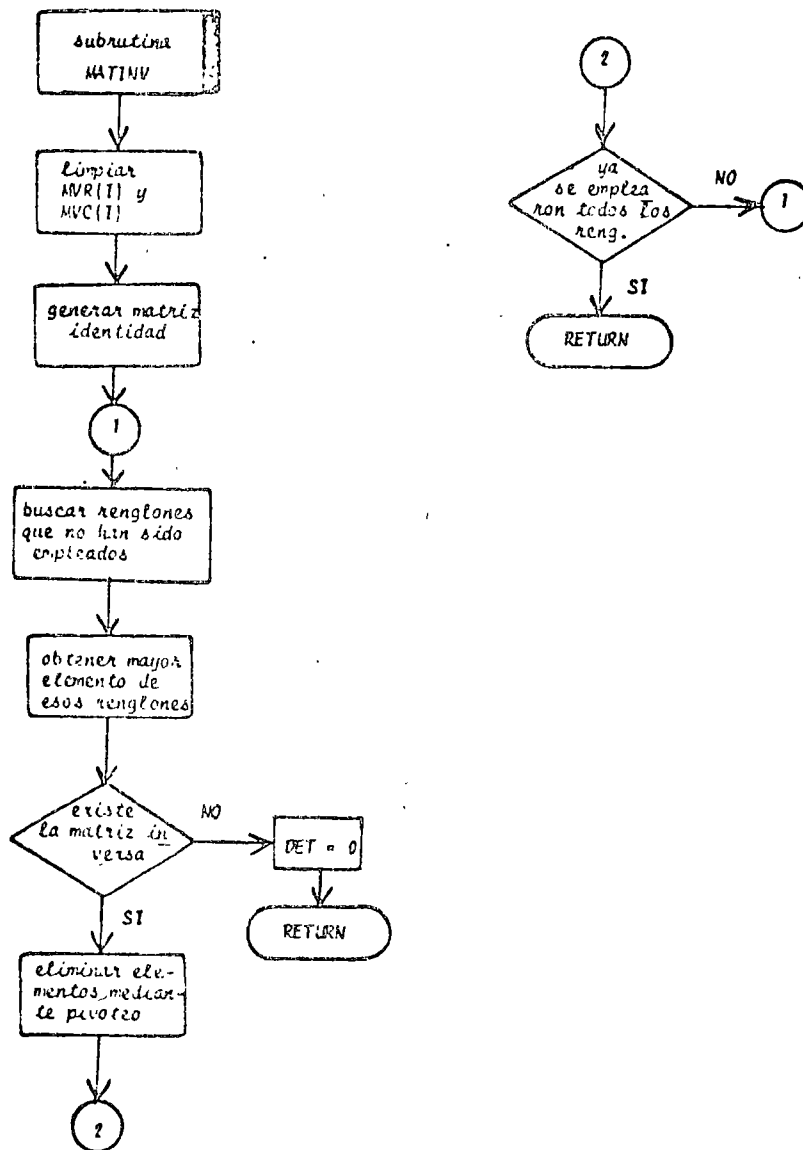


Fig. 2.10 Diagrama de bloques de la subrutina MATINV

6) Listado:

```

I      PROGRAMA PARA INVERTIR MATRICES POR EL METODO DE GAUSS-JORDAN
C      SIGNIFICADO DE LAS VARIABLES EMPLEADAS
C      N=ORDEN DE LA MATRIZ A
C      A=MATRIZ DE LA QUE SE BUSCA SU INVERSA
C      EPS=CRITERIO PARA DETERMINAR SI EXISTE O NO LA INVERSA DE LA MATRIZ
C      DET=PARAMETRO QUE INDICA SI EXISTE O NO LA INVERSA DE LA MATRIZ
-----
DIMENSION A(10,10),C(10,10)
IN=5
INFO
EPS=0.000001
C      LECTURA DE DATOS
1 READ(8,19) N
2 IF(N) 2,2,3
3 CALL EXIT
4 DO 4 I=1,N
5 READ(10,20) (A(I,J),J=1,N)
C      IMPRESION DE DATOS
WRITE(11,21)
DO 5 I=1,N
5 WRITE(11,22) (A(I,J),J=1,N)
C      LLAMADO DE SUBROUTINA PARA OBTENER LA MATRIZ INVERSA
CALL INVER(A,N,EPS,DET)
IF(ABS(DET).GT.EPS) GO TO 7
WRITE(11,23)
GO TO 1
7 WRITE(11,24)
DO 8 I=1,N
8 WRITE(11,22) (A(I,J),J=1,N)
GO TO 1
C      FORMATOS DE LECTURA E IMPRESION
19 FORMAT(15)
20 FORMAT(8F10.0)
21 FORMAT(4(//),5X,'MATRIZ A',//)
22 FORMAT(//,2X,10(C10.3,1X))
23 FORMAT(4(//),5X,'NO EXISTE LA MATRIZ INVERSA')
24 FORMAT(4(//),5X,'INVERSA DE LA MATRIZ A')
END
-----

```

Fig. 2.11 Listado del programa principal

```

SUBROUTINE MATINV(A,N,EPS,CET)
C
C SUBROUTINA PARA OBTENER LA INVERSA DE UNA MATRIZ
C EL SIGNIFICADO DE LAS VARIABLES EMPLEADAS ES
C A=MATRIZ A LA QUE SE BUSCARA SU INVERSA Y QUE DURANTE EL PROCESO
C SE CONVIERTE EN LA MATRIZ INVERSA
C N=ORDEN DE LA MATRIZ
C EPS=CRITERIO PARA DETERMINAR SI EL DETERMINANTE DE LA MATRIZ ES
C NULO
C DET=VALOR ABSOLUTO DEL DETERMINANTE DE LA MATRIZ
C CM=MATRIZ IDENTIDAD QUE SE UTILIZA PARA OBTENER LA MATRIZ INVERSA
C POR EL METODO DE GAUSS-JORDAN MODIFICADO
C MVR Y MVC=CONTADORES QUE INDICAN CUALES RENGLONES Y COLUMNAS YA
C FUERON UTILIZADOS COMO PILOTES
C
C DIMENSION A(10,10),C(10,10),MVR(10),MVC(10)
C
C OBTENCION DE LA MATRIZ IDENTIDAD Y ACTUALIZACION DE VALORES PARA
C INICIAR EL PROCESO
C
C DO 1 I=1,N
MVR(I)=0
1 MVC(I)=0
C DO 4 I=1,N
C DO 3 J=1,N
IF(I.EQ.J) GO TO 2
C(I,J)=0.0
GO TO 3
2 C(I,J)=1.0
3 CONTINUE
4 CONTINUE
C
C OBTENCION DE LA MATRIZ INVERSA
C
C DO 12 K=1,N
RAMAX=0.0
LC=0
LR=0
C DO 6 I=1,N
IF(MVR(I).EQ.1) GO TO 4
C DO 5 J=1,N
IF(MVC(J).EQ.0) GO TO 3
IF(ABS(RAMAX).GE.ABS(A(I,J))) GO TO 5
RAMAX=A(I,J)
LR=I
LC=J
5 CONTINUE
6 CONTINUE
DET=ABS(RAMAX)
IF(DET.LE.EPS) GO TO 14
IF(LR.EQ.LC) GO TO 8
C DO 7 I=1,N
TEMP=A(LR,I)
A(LR,I)=A(LC,I)
A(LC,I)=TEMP
TEMP=C(LR,I)
C(LR,I)=C(LC,I)
7 C(LC,I)=TEMP
C DO 9 I=1,N
A(LC,I)=A(LC,I)/RAMAX
9 C(LC,I)=C(LC,I)/RAMAX
C DO 11 I=1,N
IF(I.EQ.LC) GO TO 11
TEMP=A(I,LC)
C DO 10 J=1,N
A(I,J)=A(I,J) - TEMP*A(LC,J)
10 C(I,J)=C(I,J) - TEMP*C(LC,J)
11 CONTINUE
MVR(LC)=LC
MVC(LC)=LC
12 CONTINUE
C DO 13 I=1,N
C(I,J)=C(I,J)
13 AC(I,J)=C(I,J)
14 RETURN
END

```

Fig. 2.12 Listado de la subrutina MATINV

2.4.4 Ejemplo

Obtener la inversa de la matriz:

$$\underline{A} = \begin{bmatrix} 10 & 2 & 3 & -1 \\ 1 & -20 & -1 & 3 \\ 1 & 1 & -10 & 2 \\ 2 & -1 & -1 & 30 \end{bmatrix}$$

**SOLUCION

TABLA 2.5 Datos para el problema del ejemplo 2.4.4
N=4

$$\underline{A} = \begin{bmatrix} 10 & 2 & 3 & -1 \\ 1 & -20 & -1 & 3 \\ 1 & 1 & -10 & 2 \\ 2 & -1 & -1 & 30 \end{bmatrix}$$

TABLA 2.6 Resultados del problema del ejemplo 2.4.4

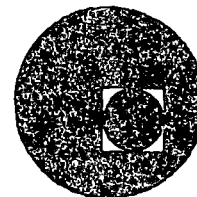
MATRIZ A			
.100E+02	.200E+01	.300E+01	-.100E+01
.100E+01	-.200E+02	-.100E+01	.300E+01
.100E+01	.100E+01	-.100E+02	.200E+01
.200E+01	-.100E+01	-.100E+01	.300E+02
INVERSA DE LA MATRIZ A			
.561E-01	.113E-01	.277E-01	.253E-03
.347E-02	-.400E-01	.553E-02	.471E-02
.570E-02	-.437E-02	-.977E-01	.724E-02
-.600E-02	-.253E-02	-.492E-02	.337E-01

2.5 Bibliografía

1. CARNAHAN B., LUTHER H., WILKES J., "Applied Numerical Methods". New York: John Wiley and Sons Inc., 1969.
pp.210-218, 282-296.
2. HADLEY G., "Algebra Lineal" . Bogotá: Fondo Educativo Interamericano, 1969.
pp.60-131.
3. HAMMING Richard, "Numerical Methods for Scientists and Engineers". New York: Mc Graw Hill Book Co., 1962.
pp.366-367.
4. JOHNSTON J., BAILEY PRICE G., VAN VLECK F., "Linear Equations and Matrices". Reading Mass.: Addison-Wesley Co., 1966.
pp.95-157.
5. KAPLAN Lewis, "Calculus and Linear Algebra Vol.2".
New York: John Wiley and Sons Inc., 1971.
pp.718-803.
6. KUO S. Shan, "Computer Applications of Numerical Methods".
Reading Mass.: Addison-Wesley Co., 1972.
pp.176-179, 189-194.



centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



METODOS NUMERICOS Y APLICACIONES CON LA
COMPUTADORA DIGITAL

TIMA 2. ALGEBRA MATRICIAL
(Complemento)

ING. HORACIO SANDOVAL

-ABRIL, 1978.

ANTECEDENTES HISTORICOS.

Hoy 3 personas que desarrollan (introducen) las matrices al mundo de las Matemáticas

William R. Hamilton (1805-1865)
James J. Sylvester (1814-1897)
Arthur Cayley (1821-1895)

El término *matrice* fue utilizado por primera vez por Sylvester en 1850, para designar un arreglo rectangular de números, a partir del cual se pueden formar determinantes. Cayley sentó las bases de la Teoría de Matrices (1857). Peano axiomatizó el álgebra de vectores e introdujo el concepto de "espacio vectorial definido sobre un campo de números". (1888).

Ref: Lectures on Matrices
J. H. M. Wedderburn
Ed. Ann Arbor, Mich., 1949.

MATRICES Y DETERMINANTES.

Las matrices y los determinantes corresponden a dos categorías matemáticas distintas.

- El determinante es un Número.
- La matriz es un conjunto ordenado

Una matriz es simplemente una forma ordenada de elementos numéricos que organiza cierta información. Esto debe interpretarse en el sentido de que entre los elementos de una matriz dada NO debe efectuarse ninguna operación algebraica.

Nota: La primera referencia a determinantes la hizo el Japonés Seki Kowa en 1683.

APLICACIONES EN LA FÍSICA Y EN LA INGENIERÍA.

La primera aplicación conocida a la física data de 1925, año en que Heisenberg, Born y Jordan aplicaron las matrices al estudio de la mecánica cuántica.

La primera aplicación conocida en Ingeniería fue en 1934, cuando Duncan y Latta lo emplearon en teoría de las vibraciones.

El advenimiento de las computadoras electrónicas ha impulsado fuertemente el uso de las matrices y en general del Álgebra lineal, debido principalmente a la sencillez con que las máquinas almacenan y manipulan la información matricial.

Usualmente el nombre de las matrices se da mediante letras mayúsculas (Latinas o Griegas).

$$A = \begin{bmatrix} 1 & -2 \\ 3 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ -1 \\ 2 \end{bmatrix}$$

$$C = \begin{bmatrix} 4 & -1 & 3 \\ 7 & 2 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 2 \\ 4 \end{bmatrix}$$

CONCEPTO DE MATRIZ

Se llama matriz a un conjunto ordenado de elementos dispuestos en m renglones y n columnas.

(renglones = filas = hileras)

El número de renglones puede ser igual, menor o mayor que el número de columnas.

Ejemplos:

$$\begin{bmatrix} 1 & 0 & 4 \\ 7 & 2 & 0 \\ 6 & 1 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 4 & 2 & 5 & 7 \\ 6 & 2 & 1 & 3 \end{bmatrix}$$

$$\begin{bmatrix} -5 \\ 0 \\ -2 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 4 \end{bmatrix}$$

y los elementos de cada una de ellos con lo mismo: *contas pendiente*

$$A = \begin{bmatrix} 1 & -2 \\ 3 & 4 \end{bmatrix} \Rightarrow a_{11} = 1 ; a_{12} = -2 ; a_{21} = 3 ; a_{22} = 4.$$

5. forma genérica se escribe como

$$a_{ij} = \text{indicatedor de fila } i \text{ ; } b_{ij} = \text{indicatedor de columna } j$$

$$A_{(2,2)} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \end{bmatrix}$$

$$\text{ó } A = [a_{ij}]_{2 \times 5} \quad \text{ó } A_{(2 \times 5)} = [a_{ij}]$$

Si A es de orden (m, n) se tiene

$$A = \begin{bmatrix} a_{11} & \dots & a_{1j} & \dots & a_{1n} \\ \vdots & & \vdots & & \vdots \\ a_{21} & \dots & a_{2j} & \dots & a_{2n} \\ \vdots & & \vdots & & \vdots \\ a_{m1} & \dots & a_{mj} & \dots & a_{mn} \end{bmatrix} = [a_{ij}]_{m \times n} \quad \text{ó } A_{m \times n} = [a_{ij}]$$

Ejemplo:

El Registro Federal de Automóviles está organizado en dos deptos: A) atiende automóviles y B) atiende camionetas. El personal en A está compuesto por 27 hombres y 18 mujeres y el depto B por 32 hombres y 6 mujeres. Dé esta información matricialmente.

Departamento	Personas	
	H	M
A	27	18
B	32	6

$$P = \begin{bmatrix} 27 & 18 \\ 32 & 6 \end{bmatrix}$$

MATRICES ESPECIALES

MATRIZ CUADRADA - Tiene el mismo número de filas que de columnas, es decir, $m=n$

Ejemplos:

$$R = [7]$$

Matriz cuadrada de orden?

$$S = \begin{bmatrix} 3 & 1 \\ -2 & 1 \end{bmatrix}$$

M. Cuadrada de orden = ?

$$T = \begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & -5 \\ 6 & 7 & -8 \end{bmatrix}$$

M. C. de orden = ?

$$U = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \dots & u_{nn} \end{bmatrix}$$

orden = ?

DIAGONAL PRINCIPAL - En una matriz cuadrada A , la diagonal principal es el conjunto de elementos a_{ij} tales que $i=j$

En los ejemplos anteriores se tiene

$$R = 7$$

$$S = 3, 1$$

$$T = 1, 4, -8$$

$$U = u_{11}, u_{22}, u_{33}, \dots, u_{nn}$$

MATRIZ DIAGONAL.

Es una matriz cuadrada en la que los elementos fuera de la diagonal principal son todos nulos.

Ejemplos:

$$E = \begin{bmatrix} 2 & 0 \\ 0 & -3 \end{bmatrix} \quad F = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad G = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad H = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}$$

Nota: No se da ninguna restricción a los elementos de la diagonal Principal.

Simbólicamente se puede expresar como:

A es diagonal si $a_{ij} = 0$ para todo $i \neq j$.

MATRIZ ESCALAR.

Es una matriz diagonal en la que todos los elementos diagonales son iguales.

¿Cuales de las matrices E, F, G y H son escalares?

Existe un caso particular de la matriz escalar que es muy importante, y es cuando todos los elementos son iguales a la Unidad.

$$I = [1] \quad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

Se simboliza con I_n o I solamente.

Así mismo existe la matriz nula donde todos sus elementos son cero. (Este concepto envuelve a las matrices rectangulares también). O .

MATRIZ TRIANGULAR SUPERIOR.

Es aquella en que todos los elementos bajo la diagonal principal son nulos.

Ejemplos.

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} \quad B = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 6 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix}$$

MATRIZ TRIANGULAR INFERIOR.

Es aquella en que todos los elementos arriba de la diagonal principal son nulos.

Ejemplos:

$$D = \begin{bmatrix} a & 0 \\ c & b \end{bmatrix} \quad E = \begin{bmatrix} x & 0 & 0 \\ y & 5 & 0 \\ z & 0 & 2 \end{bmatrix} \quad F = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

MATRIZ SIMETRICA.

Es una matriz cuadrada donde $a_{ij} = a_{ji}$ para toda i, j . ✓

Ejemplos:

$$G = \begin{bmatrix} 1 & 3 \\ 3 & 2 \end{bmatrix} \quad H = \begin{bmatrix} 0 & -3 & 4 \\ -3 & 1 & 0 \\ 4 & 0 & 2 \end{bmatrix} \quad J = \begin{bmatrix} k & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & k \end{bmatrix}$$

MATRIZ ANTISIMETRICA

Es una matriz cuadrada donde $a_{ij} = -a_{ji}$ para toda i, j . ✓

Ejemplos.

$$L = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix} \quad M = \begin{bmatrix} 0 & 1 & -4 \\ -1 & 2 & 5 \\ 4 & -5 & 3 \end{bmatrix} \quad P = \begin{bmatrix} x & 0 & 0 \\ 0 & x & 0 \\ 0 & 0 & x \end{bmatrix}$$

MATRIZ RECTANGULAR.

Es aquella matriz donde el número de columnas es diferente al número de renglones.

De este tipo de matrices las más importantes son:

VECTOR O MATRIZ RENGLO.

Se forma con un solo renglón. Su orden es $(1, n)$

Ejemplos:

$S = [0, 1]$	$T = [0, 2, 1]$	$V = [0.2, -3.14159, 4, \dots, 7]$
$m = ?$	$m = ?$	$m = ?$
$n = ?$	$n = ?$	$n = ?$

VECTOR COLUMNA O MATRIZ COLUMNA.

Es una matriz formada por una sola columna.

Su orden es $(m, 1)$

Ejemplos:

$W = \begin{bmatrix} 7 \\ 2 \\ 0 \end{bmatrix}$	$X = \begin{bmatrix} 3.14159... \\ 2.71728... \end{bmatrix}$	$Y = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$Z = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$
---	--	---	--

VECTOR NULO.

Es un vector fila o columna cuyos componentes son los ceros.

Ejemplos: $[0]$ $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ $[0 \ 0 \ 0 \ 0 \ 0]$

Nótese que es un caso particular de la Matriz Nula.

IGUALDAD DE MATRICES.

Dos matrices $A = [a_{ij}]_{m \times n}$ y $B = [b_{ij}]_{r \times s}$ son iguales si y sólo si satisfacen:

- Son del mismo orden
- Los elementos correspondientes son iguales.

Es decir,

$$A = B \Rightarrow \begin{cases} m = r ; n = s \\ a_{ij} = b_{ij} \text{ para todo } i, j. \end{cases}$$

Ejemplos:

$$A = \begin{bmatrix} 5 & 2 \\ 4 & 3 \end{bmatrix}$$

$$B = \begin{bmatrix} 5 & 2 \\ 4 & 3 \end{bmatrix}$$

$$C = \begin{bmatrix} (x+1)(x-1) \\ 3x-2x \end{bmatrix}$$

$$D = \begin{bmatrix} x^2-1 \\ x \end{bmatrix}$$

$$E = \begin{bmatrix} 2 \\ 4 \\ 8 \end{bmatrix}$$

$$F = \begin{bmatrix} 2^1 \\ 2^2 \\ 2^3 \end{bmatrix}$$

$$J = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$K = [0, 0]$$

En caso de no cumplirse alguna de las condiciones indicadas las matrices son desiguales o no comparables.

OPERACIONES ENTRE MATRICES.

TRANSPOSICION.

La transpuesta de una matriz $A = [a_{ij}]$ de orden (m, n) es una matriz de orden (n, m) . El elemento a_{ij} de la matriz A ocupa el lugar a_{ji} en la matriz transpuesta de A , que se simboliza por A' o A^t

$A \rightarrow$ Transposición $\rightarrow A^t$

Ejemplos:

$A = \begin{bmatrix} 2 & 0 & -1 \\ 3 & -2 & 4 \end{bmatrix}$ $A^t = \begin{bmatrix} 2 & 3 \\ 0 & -2 \\ -1 & 4 \end{bmatrix}$
(2,3) (3,2)

$B = [abc]$ $B^t = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$
(1x3) (3x1)

$C = \begin{bmatrix} 4 & 7 & 6 \\ 5 & 9 & -1 \\ -4 & -1 & 0 \end{bmatrix}$ $C^t = \begin{bmatrix} 4 & 5 & -4 \\ 7 & 9 & -1 \\ 6 & -1 & 0 \end{bmatrix}$
(3x3) (3x3)

$D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & \dots & \vdots \\ d_{m1} & d_{m2} & \dots & d_{mn} \end{bmatrix}$ $D^t = \begin{bmatrix} d_{11} & d_{21} & \dots & d_{m1} \\ d_{12} & d_{22} & \dots & d_{m2} \\ \vdots & \vdots & \dots & \vdots \\ d_{1n} & d_{2n} & \dots & d_{mn} \end{bmatrix}$
(m x n) (n x m)

- Si E es una matriz simétrica ¿cuánto vale E^t ?
- Sea G la transpuesta de F , ¿cuánto vale la transp. de G ?
 $G = F^t \Rightarrow G^t = (F^t)^t = ?$
- Si H es una matriz diagonal, ¿cuánto vale H^t ?
- Si J es una matriz escalar ¿cuánto vale J^t ?
- Si I es la Identidad ¿cuánto vale I^t ?

CASOS ESPECIALES DE LA SUMA.

i) $A + 0 = A$

ii) $Z + (-Z) = 0$

iii) A y B son simétricas ¿C = A + B que será?

iv) A y B son triangulares superiores
¿C = A + B? que será?

v) D₁ y D₂ son matrices diagonales
¿D₃ = D₁ + D₂ que será?

vi) Si D es diagonal y S es simétrica.
¿H = D + S que será?

¿Qué sucede con la resta de los casos i...vi?

PRODUCTO DE UNA MATRIZ POR UN NUMERO.

Dada una matriz $A = [a_{ij}]$, y un número λ , el producto $\lambda \cdot A = A \cdot \lambda$ es otra matriz del mismo orden B ($B = \lambda A$) que se obtiene multiplicando por λ cada uno de los elementos de la Matriz A.

$[B] = \lambda \cdot [A] = [\lambda A] = [b_{ij}] = [\lambda a_{ij}]$ para toda i,j.

Ejemplos:

$\lambda = 0.5 \quad A = \begin{bmatrix} 2 & 1 & 3 \\ 4 & -7 & 5 \end{bmatrix}$

$B = \lambda A = \begin{bmatrix} 1 & 0.5 & 1.5 \\ 2 & -3.5 & 2.5 \end{bmatrix}$

$\lambda = -1 \quad G = \begin{bmatrix} 4 & -7 \\ 8 & -9 \end{bmatrix} \quad H = \lambda G = \begin{bmatrix} ? \\ ? \end{bmatrix}$

Casos Especiales

i) $\lambda \cdot 0 = 0$

ii) $0 \cdot A = 0$

iii) $1 \cdot A = A$

iv) Prop. Distributiva $(\lambda + \mu) \cdot A = \lambda A + \mu A$

$$\lambda(A+B) = \lambda A + \lambda B$$

v) $\lambda I = A$ ¿A es de tipo?

PRODUCTO DE MATRICES.

Para efectuar el producto de dos matrices se requiere:

a) Que sean conformables para el producto, Esto significa que si $C = A \cdot B$, A debe tener el mismo número de columnas que el número de filas de B.

Si A es (m, n) y B es (s, t)

$C = A \cdot B$ existe únicamente si $n = s$

b) El producto se define como. (un elemento)

$$c_{ij} = \sum_{k=1}^n a_{ik} \cdot b_{kj}$$

o sea, el elemento que se encuentre en el renglón i y la columna j de la matriz producto, (c) , se obtiene multiplicando los elementos del i -ésimo renglón de A por los elementos correspondientes de la j -ésima columna de B. y sumando los productos parciales.

¿Dudas?

Ejemplo:

Sean A y B

$$A = \begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix} \quad B = \begin{bmatrix} -1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \end{bmatrix}$$

¿Es posible $A \cdot B$?

¿Es posible $B \cdot A$?

Dado que A es (3×2) y B es (2×4) . AB es posible y BA no lo es.

$$C = A \cdot B = \begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \end{bmatrix}$$

$(3 \times 2) \quad (2 \times 4)$

$$C = \begin{bmatrix} 0 \cdot (-1) + 1 \cdot 1 & 0 \cdot 0 + 1 \cdot 1 & 0 \cdot 1 + 1 \cdot 0 & 0 \cdot (-1) + 1 \cdot (-1) \\ 2 \cdot (-1) + 3 \cdot 1 & 2 \cdot 0 + 3 \cdot 1 & 2 \cdot 1 + 3 \cdot 0 & 2 \cdot (-1) + 3 \cdot (-1) \\ 4 \cdot (-1) + 5 \cdot 1 & 4 \cdot 0 + 5 \cdot 1 & 4 \cdot 1 + 5 \cdot 0 & 4 \cdot (-1) + 5 \cdot (-1) \end{bmatrix} \text{ que es } (3 \times 4)$$
$$= \begin{bmatrix} 1 & 1 & 0 & -1 \\ 1 & 3 & 2 & -5 \\ 1 & 5 & 4 & -9 \end{bmatrix} \quad (3 \times 4)$$

Ejemplo

$$A = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \quad (2 \times 2)$$

$$B = \begin{bmatrix} 2 & 3 & 1 \\ 1 & -1 & 1 \end{bmatrix} \quad (2 \times 3)$$

$$\longrightarrow C = A \cdot B = (2 \times 3)$$

$$C = \begin{bmatrix} 2 \cdot 2 + 1 \cdot 1 & 2 \cdot 3 + 1 \cdot (-1) & 2 \cdot 1 + 1 \cdot 1 \\ 3 \cdot 2 + 4 \cdot 1 & 3 \cdot 3 + 4 \cdot (-1) & 3 \cdot 1 + 4 \cdot 1 \end{bmatrix} = \begin{bmatrix} 5 & 5 & 3 \\ 10 & 5 & 7 \end{bmatrix}$$

Ejemplo

$$C = A \cdot B = \begin{bmatrix} -2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 1 & 3 & 2 \end{bmatrix} = \begin{bmatrix} -2 & -4 & -6 \\ 12 & 8 & 4 \\ 1 & 3 & 2 \end{bmatrix}$$

$$(2 \times 3) \quad (3 \times 3) \Rightarrow C (2 \times 3)$$

Ejemplo:

Una empresa que fabrica televisores desea calcular el número de bulbos y bocinas necesarias para programar el proceso de producción de sus tres modelos.

Sus requerimientos se dan en la siguiente tabla

	Med A	Med B	Med C	
bulbos	$\begin{bmatrix} 13 & 18 & 20 \\ 2 & 3 & 4 \end{bmatrix}$			\leftarrow Matriz de requerimientos de Partes por televisión
bocinas				

Para enero proximo se requieren 120 unidades del modelo A, 240 del B y 120 del C.

Para Febrero se estiman las necesidades en 60 de A, 120 de B y 90 de C. ¿Cuántos bulbos y bocinas se requieren para los dos meses?

Arreglando las necesidades de enero y febrero en forma matricial se tiene:

	Enro	Febr.
Med A	$\begin{bmatrix} 120 & 60 \\ 240 & 120 \\ 120 & 90 \end{bmatrix}$	
Med B		
Med C		

$$\begin{bmatrix} 13 & 18 & 20 \\ 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 120 & 60 \\ 240 & 120 \\ 120 & 90 \end{bmatrix} = \begin{bmatrix} 8280 & 4740 \\ 1440 & 840 \end{bmatrix} \quad \text{o sea:}$$

	Enro	Febrero
Bulbos	$\begin{bmatrix} 8280 & 4740 \\ 1440 & 840 \end{bmatrix}$	
Bocinas		

¿Que pasa si se requiere Marzo también?

CASOS ESPECIALES.

i) Vector columna $(m, 1)$ por vector fila $(1, n)$

$$X \cdot Y = Z$$

$$(m \times 1)(1 \times n) \rightarrow (m \times n)$$

Ejemplo: $X = \begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix}$ $Y = [5 \ -4 \ 2 \ -1]$
 3×1 1×4

$$Z = \begin{bmatrix} 5 & -4 & 2 & -1 \\ -10 & 8 & -4 & 2 \\ 15 & -12 & 6 & -3 \end{bmatrix}$$
 3×4

ii) Vector fila $(1, n)$ por vector columna

$$\begin{matrix} X \\ (1 \times n) \end{matrix} \cdot \begin{matrix} Y \\ (n \times 1) \end{matrix} = \begin{matrix} Z \\ (1 \times 1) \end{matrix}$$

$$Y = [5 \ 0 \ -1] \quad X = \begin{bmatrix} 0 \\ 2 \\ -7 \end{bmatrix}$$

$$Z = [5 \cdot 0 + 0 \cdot 2 + (-1) \cdot (-7)] = [7]$$

Sean las matrices A y B conformables para la multiplicación tales que.

$$C = A \cdot B$$

$$(p \times q) \quad (p \times r) \quad (r \times q)$$

¿Es posible efectuar el producto $B \cdot A$?

¿Será lo mismo $A \cdot B = B \cdot A$?

En general NO, lo que implica que el producto entre matrices NO es conmutativo.

PRODUCTOS CONMUTABLES:

(Casos especiales)

i) Una matriz cuadrada por la Identidad

$$A \cdot I = I \cdot A = A$$

(n x n) (n x n)

ii) ¿Será conmutable una matriz cuadrada por una matriz ciclica?

* iii) $A = \begin{bmatrix} 1 & 2 \\ -3 & 4 \end{bmatrix}$ $B = \begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix}$

$$A \cdot B = \begin{bmatrix} 2 & 1 \\ 4 & 7 \end{bmatrix} \quad B \cdot A = \begin{bmatrix} 3 & -4 \\ -2 & 6 \end{bmatrix}$$

$$\therefore AB \neq BA.$$

iv) ¿Producto de 2 matrices diagonales?

v) ¿Producto por la Matriz Nula?

vi) ¿Producto de una matriz ^{cuadrada} por si misma?

vii) Producto de Matrices Inversas.

≙ A es inversa de B si:

$$AB = BA = I$$

Ejemplo $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$ $B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.333 \end{bmatrix}$

$$A \cdot B = B \cdot A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ref: MATRICES. Aplicaciones matemáticas en economía y administración.
A. KLEIMAN.

Ed. Limusa 1973.

PROGRAMA PARA SUMAR DOS MATRICES.

DIMENSION A(20,20), B(20,20), C(20,20)

LECTURA DEL ORDEN DE LAS MATRICES

READ (5,100) M, N

100. FORMAT (2I5)

LECTURA DE LAS MATRICES (POR RENGLONES)

READ (5,101) ((A(I,J), J=1,N), I=1,M)

101. FORMAT (10F8.3)

READ (5,101) ((B(I,J), J=1,N), I=1,M)

VARIACION DE RENGLONES

DO 1 I=1,M

VARIACION DE COLUMNAS

DO 2 J=1,N

SUMA DE LOS ELEMENTOS C(I,J) = A(I,J) + B(I,J)

C(I,J) = A(I,J) + B(I,J)

2 CONTINUE

1 CONTINUE

IMPRESION DE LA MATRIZ SUMA

WRITE (6,102) ((C(I,J), J=1,N), I=1,M)

102. FORMAT (20X,10F8.3)

CALL EXIT

END.

PROGRAMA PARA MULTIPLICAR DOS MATRICES

DIMENSION A(20,20), B(20,20), C(20,20)

LECTURA DEL ORDEN DE LAS MATRICES

```
READ (5,100) M1, N1, M2, N2
100 FORMAT (4I5)
```

ANALISIS DE CONFORMABILIDAD PARA LA MULTIPLICACION

```
IF (N1.NE. M2) CALL EXIT
```

LECTURA DE LAS MATRICES (POR RENGLONES)

```
READ (5,101) ((A(I,J), J=1, N1), I=1, M1)
READ (5,101) ((B(I,J), J=1, N2), I=1, M2)
101 FORMAT (10F8.3)
```

VARIACION DE RENGLONES DE LA MATRIZ PRODUCTO

```
DO 1 I = 1, M1
```

VARIACION DE COLUMNAS DE LA MATRIZ PRODUCTO

```
DO 2 J = 1, N2
```

VARIACION DEL INDICE DE MULTIPLICACION

```
S = 0.0
```

```
DO 3 K = 1, M2
```

```
3 S = S + A(I,K) * B(K,J)
```

```
C(I,J) = S
```

```
2 CONTINUE
```

```
1 CONTINUE
```

IMPRESION DE LA MATRIZ PRODUCTO.

```
WRITE (6,102) ((C(I,J), J=1, N2), I=1, M1)
102 FORMAT (20X, 10F8.3/)
CALL EXIT
END.
```



centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



MÉTODOS NUMÉRICOS Y APLICACIONES CON LA
COMPUTADORA DIGITAL

TEMA II: ALGEBRA MATRICIAL
(COMPLEMENTO)

M. EN C. VERONICA CZITROM

INVERSO DE UNA MATRIZ

NUM. REALES: $a \neq 0$, $a \cdot b = b \cdot a = 1$ $b = a^{-1} = \frac{1}{a}$
 $b \cdot b = b \cdot b = 1 \Rightarrow b = \frac{1}{b} = \dots$

MATRICES: $A \cdot B = I$, $B \cdot A = I$

$A \cdot B = B \cdot A = I \Rightarrow A$ cuadrada

$B = A^{-1}$ INVERSA MULTIPLIC.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$B = \begin{bmatrix} -2 & 1 \\ 3/2 & -1/2 \end{bmatrix}$$

$$A \cdot B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} -2 & 1 \\ 3/2 & -1/2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I_2 = B \cdot A$$

$A = B^{-1}$, $B = A^{-1}$

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$B = ? = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$A \cdot B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a+2c & b+2d \\ 3a+4c & 3b+4d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{cases} a+2c=1 \\ 3a+4c=0 \\ b+2d=0 \\ 3b+4d=1 \end{cases}$$

$$\begin{aligned} a &= 1-2c \\ 3(1-2c)+4c &= 0 \quad -2c = -3 \\ c &= ? \\ a &= 1-2(-3) = 1+6 = 7 \\ d &= -2 \end{aligned}$$

$$\begin{cases} x_1 + 2x_2 = 4 \\ 3x_1 + 4x_2 = -2 \end{cases}$$

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ -2 \end{pmatrix}$$

$$A \underline{x} = \underline{b}$$

?

$$\underline{x} = A^{-1} \underline{b}$$

$$A^{-1} = \begin{pmatrix} -2 & 1 \\ 3/2 & -1/2 \end{pmatrix}, \quad \underline{x} = \begin{pmatrix} -2 & 1 \\ 3/2 & -1/2 \end{pmatrix} \begin{pmatrix} 4 \\ -2 \end{pmatrix} = \begin{pmatrix} -10 \\ 7 \end{pmatrix}$$

x_1

x_2

$$\underline{A} \underline{x} = \underline{b}$$

varias diferentes, inventa A

A VECES, LAS COMPONENTES DE A^{-1} TIENEN INTERPRETACION ESPECIAL.

AJUSTE POR MÍNIMOS CUADRADOS:

A^{-1} : COMPONENTES \rightarrow CLASE Y MAGNITUD DE ERRORES EN LOS DATOS.

INVERSION DE MATRICES

4

MÉTODOS: $\left\{ \begin{array}{l} \text{ALGEBRAICOS (EXACTOS)} \\ \text{NUMÉRICOS (APROXIMACIONES} \\ \text{SUCCESIVAS)} \end{array} \right.$

ALGEBRAICOS

ADJUNTA

$$A^{-1} = \frac{\text{adj } A}{|A|}$$

$$A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

$$\text{adj } A = \begin{pmatrix} + \begin{vmatrix} e & f \\ h & i \end{vmatrix} & - \begin{vmatrix} d & f \\ g & i \end{vmatrix} & + \begin{vmatrix} d & e \\ g & h \end{vmatrix} \\ - \begin{vmatrix} b & c \\ h & i \end{vmatrix} & + \begin{vmatrix} a & c \\ g & i \end{vmatrix} & - \begin{vmatrix} a & b \\ g & h \end{vmatrix} \\ + \begin{vmatrix} b & c \\ e & f \end{vmatrix} & - \begin{vmatrix} a & e \\ d & f \end{vmatrix} & + \begin{vmatrix} a & b \\ d & e \end{vmatrix} \end{pmatrix}$$

$$\begin{aligned} |A| &= \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix} \\ &= a(ei - fh) - b(di - fg) + c(dh - eg) \end{aligned}$$

5

SOLUCION SISTEMA DE ECUACIONES

$$A = \begin{pmatrix} 6 & 2 \\ 2 & 1 \end{pmatrix}$$

$$AB = I$$

$$\begin{pmatrix} 6 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\begin{cases} 6a + 2c = 1 \\ 2a + c = 0 \\ 6b + 2d = 0 \\ 2b + d = 1 \end{cases}$$

$$a = 0.5$$

$$c = -1$$

$$b = -1$$

$$d = 3$$

$$\begin{pmatrix} 6 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} a \\ c \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 6 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} b \\ d \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

A DE 3x3 $A^{-1} = B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}$

HAY QUE RESOLVER

$$A \begin{pmatrix} b_{11} \\ b_{21} \\ b_{31} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad A \begin{pmatrix} b_{12} \\ b_{22} \\ b_{32} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad A \begin{pmatrix} b_{13} \\ b_{23} \\ b_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

INVERSION POR PARTICION

$$A = \left(\begin{array}{c|c} A_1 & A_2 \\ \hline A_3 & A_4 \end{array} \right) \quad A^{-1} = B = \left(\begin{array}{c|c} B_1 & B_2 \\ \hline B_3 & B_4 \end{array} \right)$$

$$A A^{-1} = AB = \left(\begin{array}{c|c} A_1 & A_2 \\ \hline A_3 & A_4 \end{array} \right) \left(\begin{array}{c|c} B_1 & B_2 \\ \hline B_3 & B_4 \end{array} \right) = I = \left(\begin{array}{c|c} I & O \\ \hline O & I \end{array} \right)$$

$$\left(\begin{array}{cc|cc} A_1 B_1 + A_2 B_3 & A_1 B_2 + A_2 B_4 & & \\ A_3 B_1 + A_4 B_3 & A_3 B_2 + A_4 B_4 & & \\ \hline & & I & O \\ & & O & I \end{array} \right) = \left(\begin{array}{cc} I & O \\ O & I \end{array} \right)$$

$$A^{-1} = B = \left(\begin{array}{c|c} A_1^{-1} (I - A_2 B_3) & -A_1^{-1} A_2 B_4 \\ \hline -B_4 A_3 A_1^{-1} & (A_4 - A_3 A_1^{-1} A_2)^{-1} \end{array} \right)$$

INVERTIR A_1^{-1} Y $(A_4 - A_3 A_1^{-1} A_2)^{-1}$

DE ORDENES MENORES QUE A

DIAGONALIZACIÓN (GAUSS-JORDAN)

$$[A : I]$$



$$[I : A^{-1}]$$

$$\left[\begin{array}{cccc|cccc} a_{11} & a_{12} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 1 & \dots & 0 \\ & & \vdots & & & & \ddots & \\ a_{n1} & a_{n2} & \dots & a_{nn} & 0 & 0 & \dots & 1 \end{array} \right]$$



$$\left[\begin{array}{cccc|cccc} 1 & 0 & \dots & 0 & b_{11} & b_{12} & \dots & b_{1n} \\ 0 & 1 & \dots & 0 & b_{21} & b_{22} & \dots & b_{2n} \\ & & \vdots & & & & \ddots & \\ 0 & 0 & \dots & 1 & b_{n1} & b_{n2} & \dots & b_{nn} \end{array} \right]$$

POR OPERACIONES ELEMENTALES DE RENGLÓN:

1) INTERCAMBIO DE RENGLONES.

2) MULTIPLICAR UN RENGLON POR UN ESCALAR.

3) MULTIPLICAR UN RENGLON POR UN ESCALAR Y SUMARLO A OTRO RENGLÓN.

8

POTENCIAS DE UNA MATRIZ

$$A^n = \underbrace{A \cdot A \cdots A}_n \text{ FACTORES}$$

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \quad A^2 = A \cdot A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 7 & 10 \\ 15 & 22 \end{pmatrix}$$

$$A^0 = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (a^0 = 1)$$

$$A^{-1} = \begin{pmatrix} -2 & 1 \\ 3/2 & -1/2 \end{pmatrix}$$

$$A^{-2} = (A^{-1})^2 = \begin{pmatrix} -2 & 1 \\ 3/2 & -1/2 \end{pmatrix} \begin{pmatrix} -2 & 1 \\ 3/2 & -1/2 \end{pmatrix} = \begin{pmatrix} 11 & -5/2 \\ 15 & 1/4 \end{pmatrix}$$

PROPIEDADES DE MATRICES

$$1) \text{ NUMEROS REALES: } ab = 0 \Rightarrow \begin{cases} a = 0 \\ \text{or} \\ b = 0 \end{cases}$$

$$\text{MATRICES: } AB = 0 \not\Rightarrow \begin{cases} A = 0 \\ \text{ni} \\ B = 0 \end{cases}$$

$$\begin{pmatrix} 1 & -1 \\ 7 & -7 \end{pmatrix} \begin{pmatrix} \alpha & \alpha \\ \alpha & \alpha \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$\neq 0 \quad \neq 0$

$$2) \quad ab = ac \Rightarrow b = c$$

$$AB = AC \not\Rightarrow B = C$$

$$\begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 7 & 10 \\ 14 & 20 \end{pmatrix}$$

|| ~~||~~ ||

$$\begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} -3 & -2 \\ 5 & 6 \end{pmatrix} = \begin{pmatrix} 7 & 10 \\ 14 & 20 \end{pmatrix}$$

$$3) \quad ab = ba$$

$$AB \neq BA \quad \text{EN GENERAL}$$

$$4) \quad ab = 1 \Rightarrow b = \frac{1}{a} = a^{-1}$$

$$AB = I \quad B = A^{-1} \quad \text{NO SIEMPRE EXISTE}$$

5) NO EXISTE TRANSPUESTO DE NUM. REAL

$$A \quad A' \quad (A')' = A$$

~~EL~~

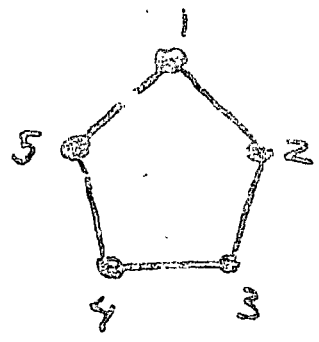
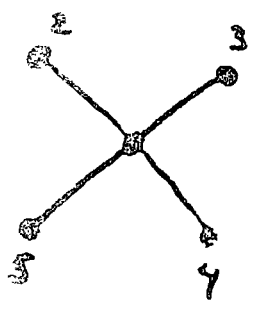
ALGEBRA

UN

APLICACIONES DE MATRICES

TEORIA DE REDES

GRAFICAS QUIMICAS O ESTRUCTURALES



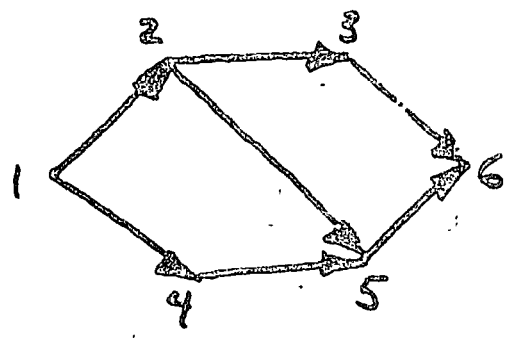
ENLACE : 1 } MATRIZ DE
 NO ENLACE : 0 } INCIDENCIA

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

ruta CRITICA

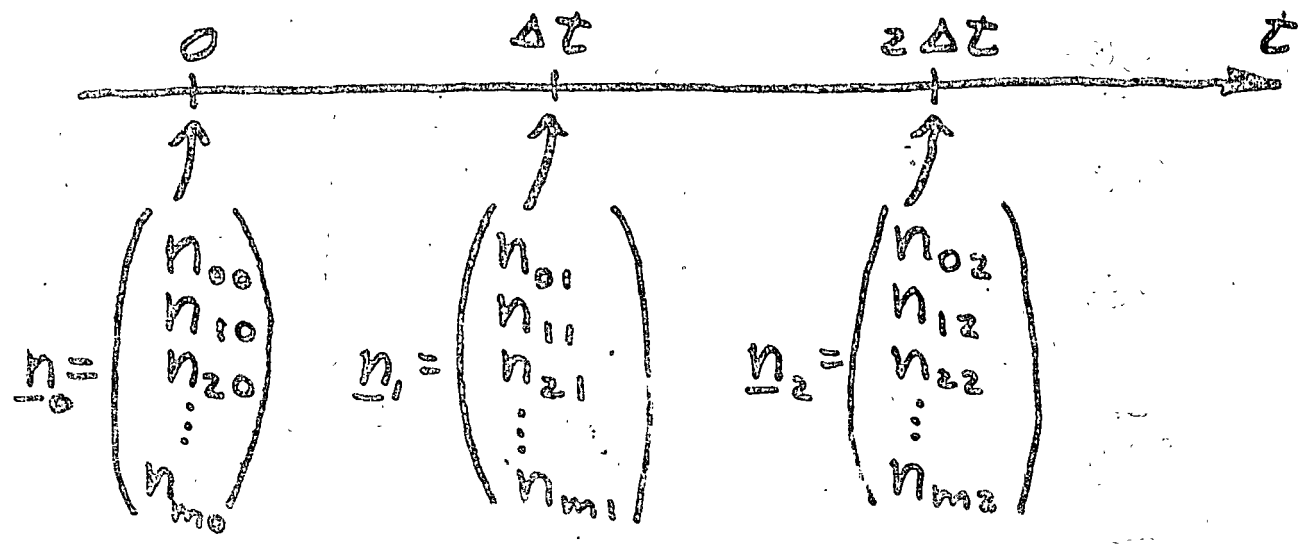


$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

2) DINÁMICA DE POBLACIÓN

$\Delta t =$ INTERVALO DE TIEMPO

GRUPOS DE EDADES	EDAD	NUMERO PROMEDIO DE HIJAS	PROBABILIDAD DE QUE UNA HEMBRA VIVA PARA ENTRAR AL SIGUIENTE GRUPO DE EDADES $x+1$
$x = 0$	$0 - \Delta t$	F_0	P_0
$x = 1$	$\Delta t - 2\Delta t$	F_1	P_1
$x = 2$	$2\Delta t - 3\Delta t$	F_2	P_2
\vdots	\vdots	\vdots	\vdots
$x = m$	$m\Delta t - (m+1)\Delta t$	F_m	$P_m = 0$



$n_{x,t}$ = NUM. DE HEMBRAS EN EL GRUPO DE EDADES x AL TIEMPO $t\Delta t$

NUM. HIJAS NACIDAS EN $0 - \Delta t$

$$F_0 n_{00} + F_1 n_{10} + \dots + F_m n_{m0} = n_{01}$$

$$P_x n_{xk} = n_{x+1, k+1} \quad x = 0, 1, \dots, m-1$$

TRANSICION DE LA POBLACION DE
 $t=0$ A $t=\Delta t$

$$\begin{pmatrix} F_0 & F_1 & \dots & F_{m-1} & F_m \\ P_0 & P_1 & \dots & P_{m-1} & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & P_{m-1} & 0 \end{pmatrix} \begin{pmatrix} n_{00} \\ n_{10} \\ n_{20} \\ \vdots \\ n_{m0} \end{pmatrix} = \begin{pmatrix} n_{01} \\ n_{11} \\ n_{21} \\ \vdots \\ n_{m1} \end{pmatrix}$$

O SEA: $M \underline{n}_0 = \underline{n}_1$

$$M \underline{n}_1 = \underline{n}_2, \quad \underline{n}_2 = M \underline{n}_1 = M(M \underline{n}_0) = M^2 \underline{n}_0$$

$$M \underline{n}_2 = \underline{n}_3, \quad \underline{n}_3 = M \underline{n}_2 = M(M^2 \underline{n}_0) = M^3 \underline{n}_0$$

$$\vdots$$

$M \underline{n}_{k-1} = \underline{n}_k$

$\underline{n}_k = M^k \underline{n}_0$

M = MATRIZ DE PROYECCION

$$\underline{n}_k = M \underline{n}_{k-1}$$

$$\underline{n}_k = M^k \underline{n}_0$$

1) DADA POBLACION INICIAL Y SU DISTRIBUCION (\underline{n}_0), Y DADA M

SE PUEDE CALCULAR POBLACION FUTURA Y SU DISTRIBUCION.

2) DADA POBLACION INICIAL Y SU DISTRIBUCION, QUE PROPIEDAD DEBE TENER M PARA TENER POBLACION ESTABLE ($\underline{n}_0 = \underline{n}_1 = \underline{n}_2 = \dots$)?

$$\underline{n}_k = M \underline{n}_k$$

$$\underline{0} = (M - I) \underline{n}_k$$

$$\Rightarrow |M - I| = 0$$

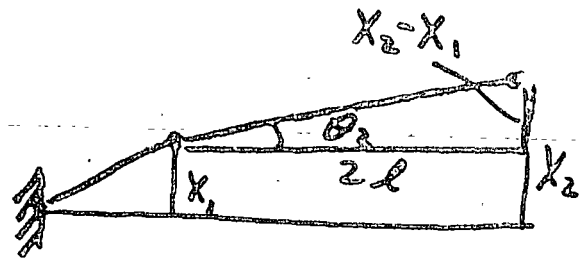
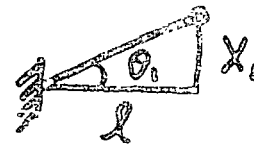
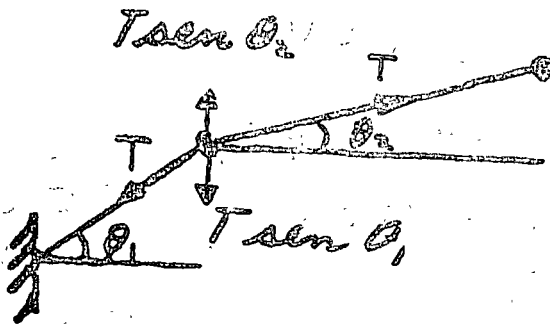
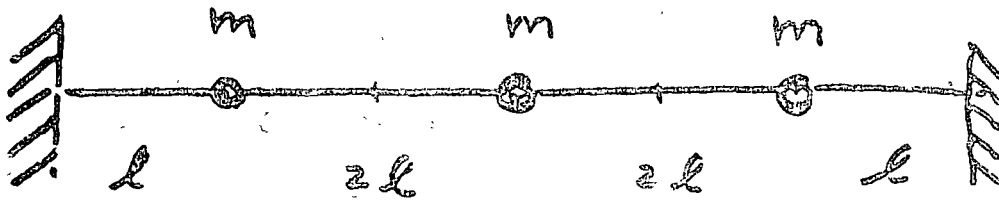
$$F_0 + F_1 P_0 + F_2 P_0 P_1 + \dots + F_m P_0 P_1 \dots P_{m-1} = 1$$

3) DADA POBLACION INICIAL Y SU DISTRIBUCION, Y DISTRIBUCION DE EDADES ESTABLE, ¿CUÁL ES LA POBLACION?

$$\underline{n}_{k+1} = \lambda \underline{n}_k$$

$$M \underline{n}_k = \lambda \underline{n}_k$$

VALORES Y VECTORES CARACTERISTICOS



$$T \sin \theta_1 \approx T \frac{x_1}{l}$$

$$T \sin \theta_2 \approx T \frac{x_2 - x_1}{2l}$$

$$F = ma = m \frac{d^2x}{dt^2}$$

$$m \frac{d^2x_1}{dt^2} = T \frac{(x_2 - x_1)}{2l} - T \frac{x_1}{l}$$

$$m \frac{d^2x_2}{dt^2} = -T \frac{(x_2 - x_1)}{2l} + T \frac{(x_3 - x_2)}{2l}$$

$$m \frac{d^2x_3}{dt^2} = -T \frac{(x_3 - x_2)}{2l} - T \frac{x_3}{l}$$

$$X_i = x_i e^{i\omega t}$$

$$\lambda = -\omega^2 m l / T$$

$$\begin{cases} (3-\lambda)x_1 - x_2 = 0 \\ -x_1 + (2-\lambda)x_2 - x_3 = 0 \\ -x_2 + (3-\lambda)x_3 = 0 \end{cases}$$

$$\begin{pmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

$$A \underline{x} = \lambda \underline{x}$$

$$(A - \lambda I) \underline{x} = 0$$

$$\exists \text{ sol} \Rightarrow |A - \lambda I| = 0$$

$$\begin{vmatrix} 3-\lambda & -1 & 0 \\ -1 & 2-\lambda & -1 \\ 0 & -1 & 3-\lambda \end{vmatrix} = 0$$

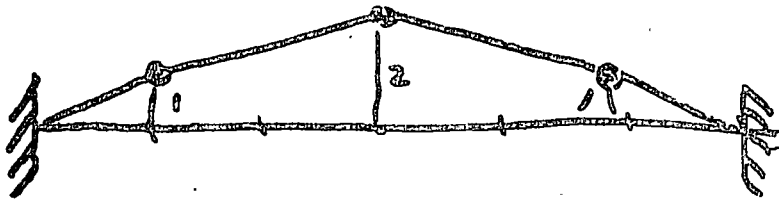
$$(1-\lambda)(3-\lambda)(4-\lambda) = 0$$

$$1) \lambda_1 = 1, \quad \begin{cases} 2x_1 - x_2 = 0 \\ -x_1 + x_2 - x_3 = 0 \\ -x_2 + 2x_3 = 0 \end{cases}$$

$$x_1 = \frac{x_2}{2} = x_3$$

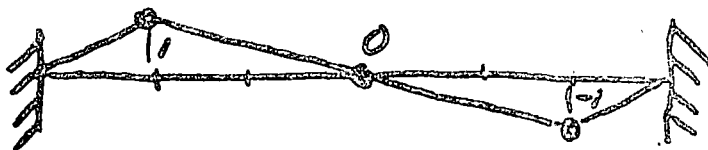
$$\underline{x} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

MODO DE VIBRACIÓN



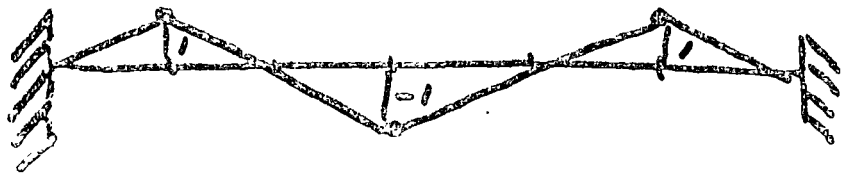
$$\omega^2 = \frac{\lambda T}{2ml} = \frac{T}{2ml} \quad \text{FRECUENCIA ANG. DE VIBRACIÓN}$$

$$2) \lambda_2 = 3 \quad \underline{x} = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$$

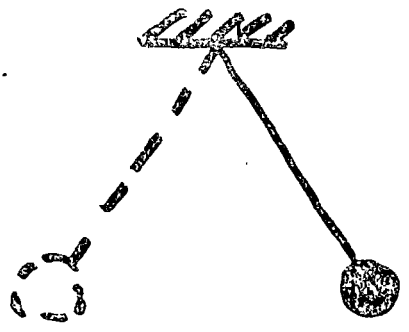


$$\omega^2 = \frac{3T}{2ml}$$

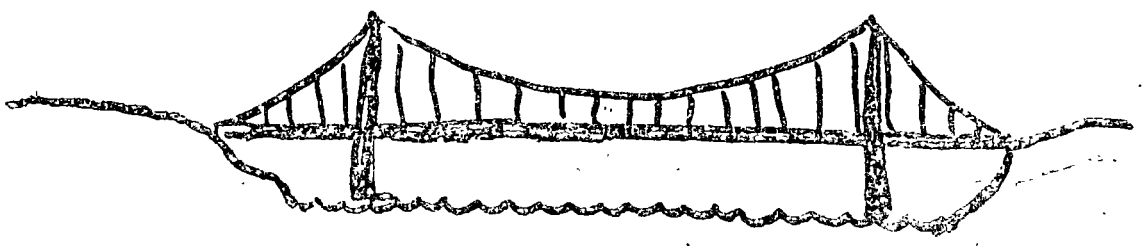
3) $\lambda = 9,$ $\underline{x} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$



RESONANCIA



PUENTE DE TACOMA

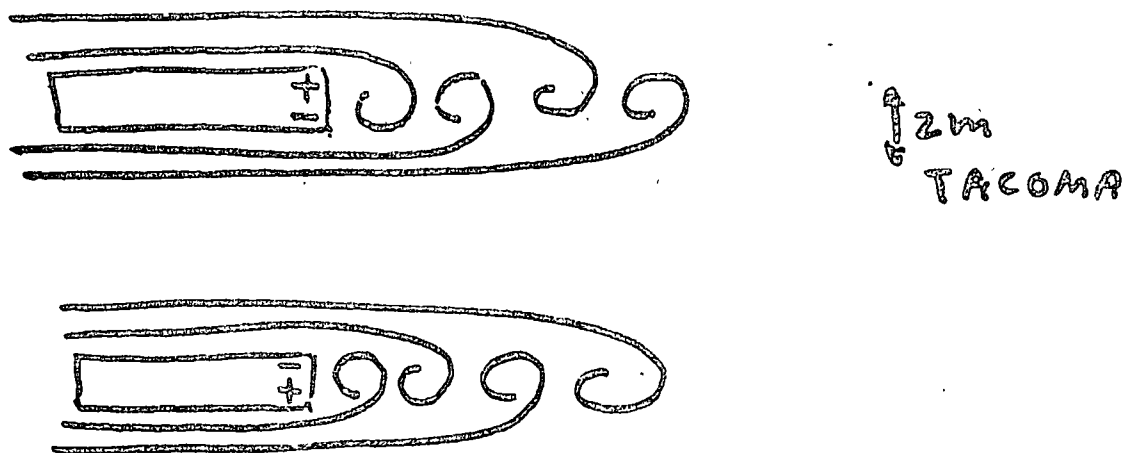
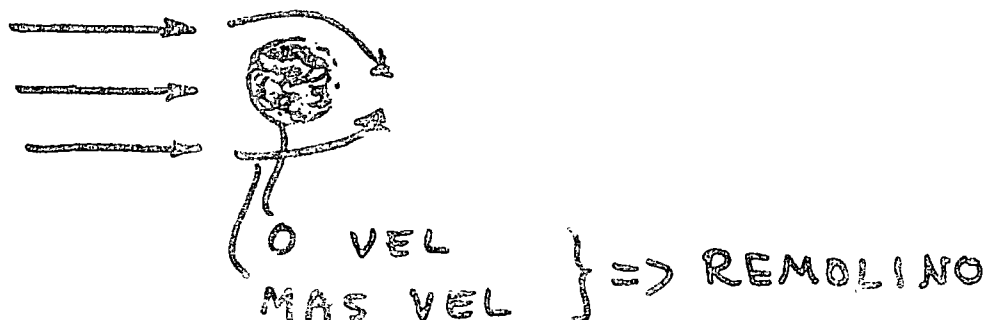


7 NOV 1940, 4 MESES DE INAUGURADO
VIENTO 67 km/h 12m x 853m

OSCILACIONES TORSIONALES

IMPULSO INVESTIGACIÓN AERODINÁMICA

FLUIDO CON OBSTÁCULO: VÓRTICES DE VON KÁRMÁN



VÓRTICES DE VON KÁRMÁN:

ALTERNADOS EN TIEMPO Y ESPACIO

- RUGOSIDAD MEDIO
- FORMA
- VELOCIDAD FLUIDO

PROBLEMA: SI LA FRECUENCIA DE LOS VÓRTICES DE VON KÁRMÁN COINCIDE CON FREC. NATURAL DE OSCILACIÓN TORSIONAL DEL PUENTE.

SOLUCIÓN: ESTRIP TUBA MAC TIERA PEFICUIAN

THEODORE VON KÁRMÁN (1881-1963)

PIONERO USO MATEMÁTICAS EN CIENCIAS BÁSICAS (AERONAUTICA, ASTRONAUTICA)

1911: ANÁLISIS DE LOS VÓRTICES

$$A \underline{x} = \lambda \underline{x}$$

SOLUCIÓN $\left\{ \begin{array}{l} \underline{x} = 0 \quad \text{TRIVIAL} \\ \underline{x} \neq 0 \Leftrightarrow |A - \lambda I| = 0 \end{array} \right.$

$$A \underline{x} = \lambda \underline{x}$$

A, n x n

$$A \underline{x} = \lambda I \underline{x}$$

$$(A - \lambda I) \underline{x} = 0$$

$\Rightarrow |A - \lambda I| = 0$ ECUACION CARACTERISTICA

$p(\lambda) = |A - \lambda I| =$ POLINOMIO CARACTERISTICO

POLINOMIO EN λ DE GRADO n

RAICES: $\lambda_1, \lambda_2, \dots, \lambda_n$

EIGENVALORES

$$|A - \lambda I| = 0$$

VALORES PROPIOS

VALORES CARACTERISTI-

COS

$$\lambda_i \longrightarrow c \underline{x}_i$$

\underline{x}_i : EIGENVECTOR
VECTOR CARACTERÍSTICO
VECTOR PROPIO

CORRESPONDIENTE A λ_i

\underline{x}_i SE OBTIENE SUSTITUYENDO EN

$$(A - \lambda I) \underline{x} = 0$$

CON $\lambda = \lambda_i$.

EJEMPLO

$$A = \begin{pmatrix} 3 & -1 \\ 4 & -2 \end{pmatrix}$$

$$|A - \lambda I| = 0$$

$$\begin{vmatrix} 3-\lambda & -1 \\ 4 & -2-\lambda \end{vmatrix} = 0$$

$$(3-\lambda)(-2-\lambda) + 4 = 0$$

$$\lambda^2 - \lambda - 2 = 0$$

$$\lambda = \frac{+1 \pm \sqrt{1 - 4(-2)}}{2} = \frac{+1 \pm 3}{2} \begin{cases} \lambda_1 = -1 \\ \lambda_2 = 2 \end{cases}$$

$$(A - \lambda I) \underline{x} = \underline{0} \quad \begin{cases} (3 - \lambda)x_1 - x_2 = 0 \\ 4x_1 - (2 + \lambda)x_2 = 0 \end{cases}$$

1) $\lambda_1 = -1$

$$\begin{cases} 4x_1 - x_2 = 0 \\ 4x_1 - x_2 = 0 \end{cases}$$

$\therefore x_2 = 4x_1$

$$\underline{x}_1 = \begin{pmatrix} x_1 \\ 4x_1 \end{pmatrix} = x_1 \begin{pmatrix} 1 \\ 4 \end{pmatrix} \quad \underline{x}_1 = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$$

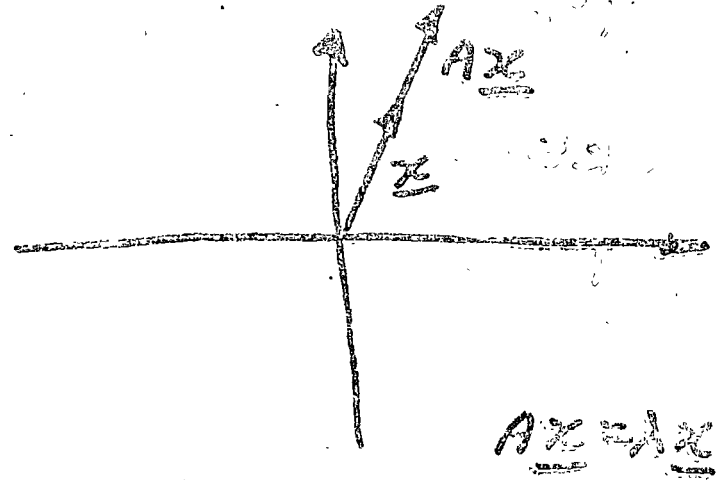
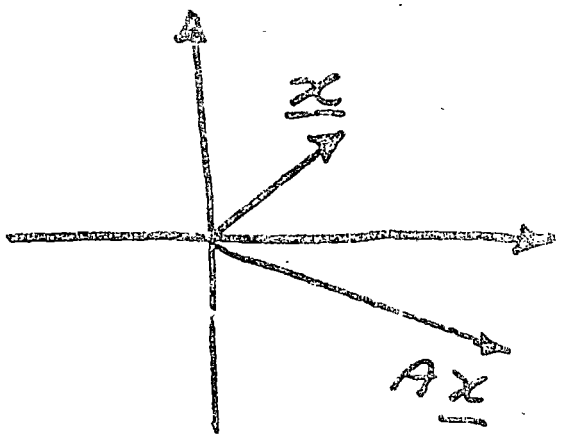
2) $\lambda_2 = 2$

$$\begin{cases} x_1 - x_2 = 0 \\ 4x_1 - 4x_2 = 0 \end{cases}$$

$\therefore x_2 = x_1$

$$\underline{x}_2 = \begin{pmatrix} x_1 \\ x_1 \end{pmatrix} = x_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \underline{x}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

INTERPRETACIÓN GEOMÉTRICA



(22)

OBTENCIÓN DE LOS VALORES CARACTERÍSTICOS

METODO DE KRYLOV

$$\left. \begin{array}{l} |A - \lambda \cdot I| = 0 \\ p(\lambda) = 0 \end{array} \right\} \text{ECUACION CARACTERÍSTICA}$$

1° OBTENER NUMÉRICAMENTE LOS COEFICIENTES c_i DEL POLINOMIO CARACTERÍSTICO

$$p(\lambda) = \lambda^n + c_{n-1}\lambda^{n-1} + \dots + c_1\lambda + c_0$$

2° OBTENER NUMERICAMENTE LOS VALORES CARACTERÍSTICOS λ_i , QUE SON LAS RAICES DE LA ECUACIÓN CARACTERÍSTICA

$$p(\lambda) = 0$$

$$\lambda^n + c_{n-1}\lambda^{n-1} + \dots + c_1\lambda + c_0 = 0$$

1° OBTENER COEFICIENTES c_i .

TEOREMA DE CAYLEY-HAMILTON:

$$\text{SI } p(\lambda) = 0 \Rightarrow p(A) = 0$$

$$P(A) = 0$$

$$A^n + c_{n-1}A^{n-1} + \dots + c_1A + c_0 = 0$$

POSTMULTIPLICANDO POR UN VECTOR ARBITRARIO \underline{y} CONOCIDO, $\underline{y} \neq \underline{0}$:

$$A^n \underline{y} + c_{n-1}A^{n-1} \underline{y} + \dots + c_1A \underline{y} + c_0 \underline{y} = \underline{0}$$

REPRESENTA UN SISTEMA DE ECUACIONES CON INCÓGNITAS c_{n-1}, \dots, c_1, c_0 .

SE RESUELVE EL SIST. DE ECS. Y SE DETERMINAN LAS c_i .

EJEMPLO:

$$A = \begin{pmatrix} 3 & -1 \\ 4 & -2 \end{pmatrix}$$

$$\text{EC. CARACT: } \lambda^2 + c_1\lambda + c_0 = 0$$

$$\text{CAYLEY HAMILTON: } A^2 + c_1A + c_0I = 0$$

$$\text{POR } \underline{y}: \quad A^2 \underline{y} + c_1A \underline{y} + c_0 \underline{y} = \underline{0}$$

$$A^2 = \begin{pmatrix} 3 & -1 \\ 4 & -2 \end{pmatrix} \begin{pmatrix} 3 & -1 \\ 4 & -2 \end{pmatrix} = \begin{pmatrix} 5 & -1 \\ 4 & 0 \end{pmatrix}; \text{ SEA } \underline{y} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

SUST:

$$\begin{pmatrix} 5 & -1 \\ 4 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} + c_1 \begin{pmatrix} 3 & -1 \\ 4 & -2 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} + c_0 \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 3 \\ 4 \end{pmatrix} + c_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + c_0 \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$c_1 + c_0 = -3$$

$$2c_0 = -4$$

$$c_0 = -2 \quad c_1 = -1$$

ECUACIÓN CARACTERÍSTICA:

$$\lambda^2 + (-4)\lambda + (-2) = 0$$

SE DEBEN OBTENER NUMÉRICAMENTE LAS RAÍCES DE LA ECN. CARACT.

MÉTODO DE JACOBI:

- MAYOR O MENOR EIGENVALOR Y EIGENVECTOR CORRESPONDIENTE

$$A \underline{x}_0 = \lambda_1 \underline{x}_1$$

APROXIMADO $\underline{x}_0 \approx \underline{x}$ NUEVA APROXIMACION A EIGENVECTOR
 FACTOR COMUN: MAYOR ELEMENTO DEL VECTOR. 1ª APROXIMACION A λ

$$A \underline{x}_0 = \lambda_1 \underline{x}_1$$

$$A \underline{x}_1 = \lambda_2 \underline{x}_2$$

⋮

$$A \underline{x}_n = \lambda_{n+1} \underline{x}_{n+1}$$

ALTO: SI $|\lambda_{n+1} - \lambda_n| < \epsilon$

MENOR EIGENVALOR:

$$A \underline{x} = \lambda \underline{x}$$

A^{-1}

$$A^{-1} A \underline{x} = A^{-1} \lambda \underline{x}$$

$$\underline{x} = \lambda A^{-1} \underline{x}$$

$$A^{-1} \underline{x} = \frac{1}{\lambda} \underline{x}$$

$$A^{-1} \underline{x} = \lambda^* \underline{x}$$

MAYOR $\lambda^* \rightarrow$ MENOR λ

$$\lambda^* = \frac{1}{\lambda}$$



centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



METODOS NUMERICOS Y APLICACIONES CON LA COMPUTADORA
DIGITAL

COMPLEMENTOS:

INTREGRACION Y DIFERENCIACION NUMERICA
RAICES DE FUNCIONES
INTERPOLACION.

PROF. M. en C. VERONICA CZITROM.

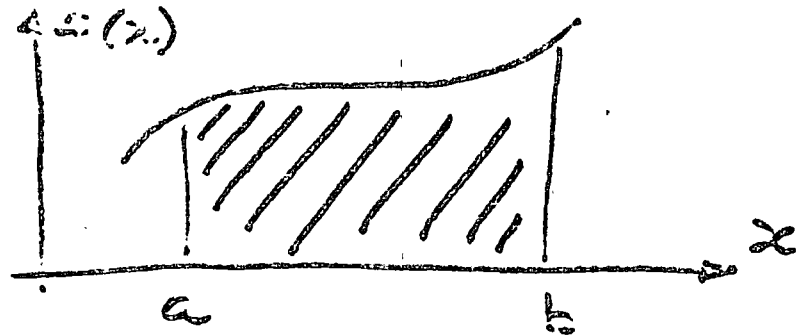
ABRIL, 1978.

①

INTEGRACIÓN Y DIFERENCIACIÓN NUMÉRICA

INTEGRACIÓN: MUCHO MÁS PRECISA QUE DER.
NUM.

INTEGRACIÓN NUMÉRICA



$$\int_a^b f(x) dx = \text{AREA BAJO LA CURVA}$$

INTEGRACIÓN NUMÉRICA PARA:

- FUNCIONES DADAS EN FORMA GRÁFICA Ó TABULAR.
- FUNCIONES MUY COMPLEJAS
- NO EXISTE INTEGRAL EXACTA.

INTEGRACIÓN NUMÉRICA: AREA BAJO CURVA
DE LA FUNCIÓN DADA EN VALORES DISCRETOS.

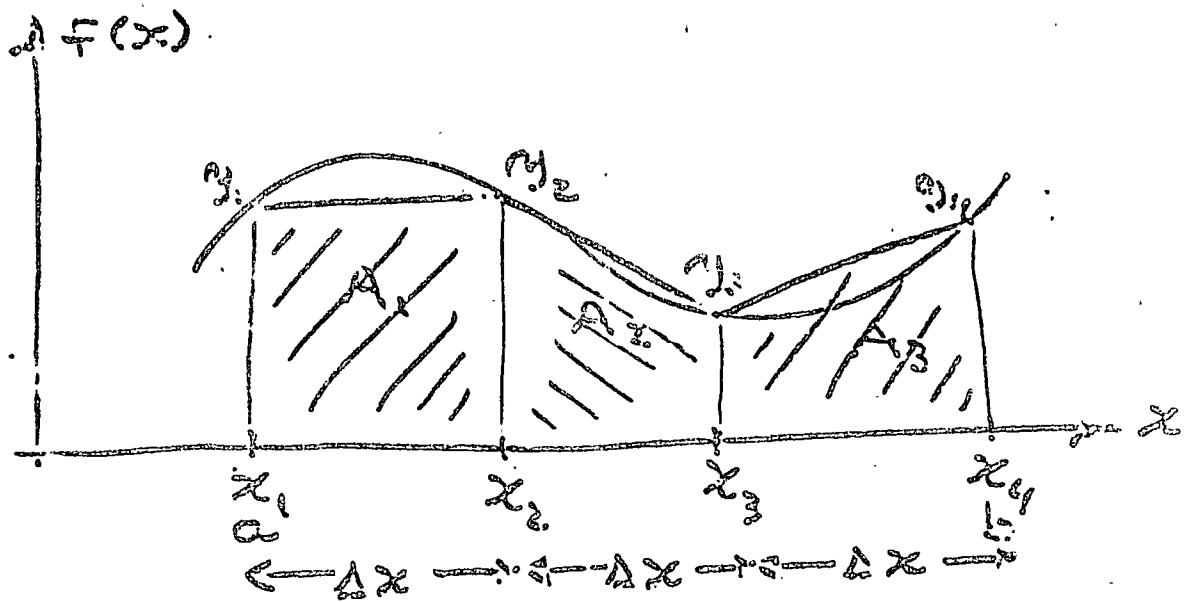
$$N = \int x(t) dt$$

$$a = \int n(t) dt$$

$$M = \int p(x) dx$$

$$W = \int F(x) dx$$

INTEGRACIÓN NUM. TRAPEZOIDAL



$$\int_a^b f(x) dx \approx A_1 + A_2 + A_3$$

AREA BAJO CURVA \approx SUMA AREAS TRAPEZOID.

$$\begin{aligned} A_1 + A_2 + A_3 &= \frac{\Delta x}{2} (y_1 + y_2) + \frac{\Delta x}{2} (y_2 + y_3) + \frac{\Delta x}{2} (y_3 + y_4) \\ &= \frac{\Delta x}{2} (y_1 + 2y_2 + 2y_3 + y_4) \end{aligned}$$

Δx PEQUEÑA DA MEJOR APROXIMACIÓN (MENOR ERROR)

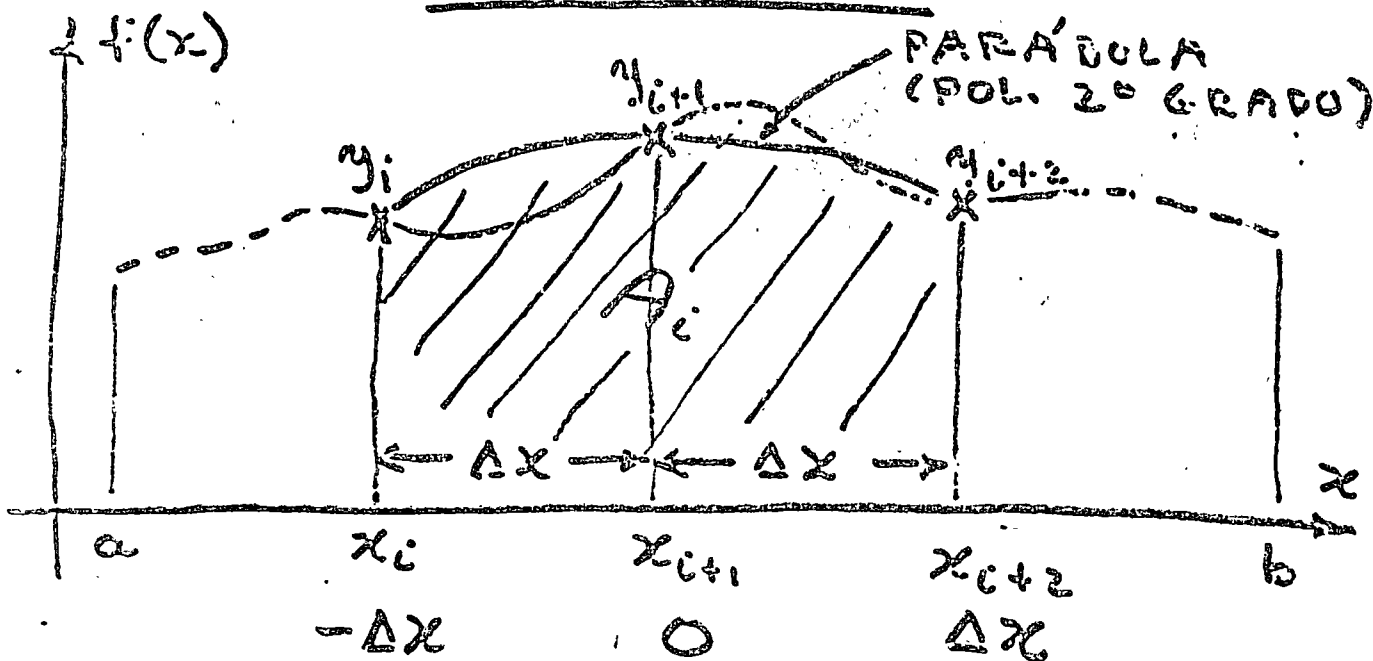
ERROR: ORDEN $(\Delta x)^2$

\Rightarrow SI LOS INTERVALOS SE REDUCEN A LA MITAD, EL ERROR SE REDUCE A LA CUARTA PARTE $\left(\left(\frac{\Delta x}{2}\right)^2 = \frac{(\Delta x)^2}{4}\right)$.

n-1 INTERVALOS:

$$\int_a^b f(x) dx \approx \frac{\Delta x}{2} [y_1 + 2y_2 + 2y_3 + \dots + 2y_{n-1} + y_n]$$

INTEGRACION NUMÉRICA DE SIMPSON 1/3



$$\int_{x_i}^{x_{i+2}} f(x) dx \approx A_i$$

AREA BAJO CURVA \approx AREA BAJO PARÁBOLA

$$f(x) \approx p(x), \quad p(x) = a + bx + cx^2$$

$$A_i = \int_{-\Delta x}^{\Delta x} p(x) dx = \int_{-\Delta x}^{\Delta x} (a + bx + cx^2) dx = (ax + \frac{b}{2}x^2 + \frac{c}{3}x^3) \Big|_{-\Delta x}^{\Delta x}$$

$$A_i = 2a\Delta x + \frac{2}{3}c(\Delta x)^3$$

$p(x)$ DEBE PASAR POR y_i, y_{i+1}, y_{i+2} :

$$\begin{cases} y_i = p(-\Delta x) = a - b\Delta x + c\Delta x^2 \\ y_{i+1} = p(0) = a \\ y_{i+2} = p(\Delta x) = a + b\Delta x + c\Delta x^2 \end{cases}$$

RESOLVIENDO SIST. ECS:

$$\begin{cases} a = y_{i+1} \\ b = (y_{i+2} - y_i) / 2\Delta x \\ c = (y_i + 2y_{i+1} + y_{i+2}) / 2\Delta x^2 \end{cases}$$

SUSTITUYENDO EN A_i:

$$\int_{x_i}^{x_{i+2}} f(x) dx \approx \frac{\Delta x}{3} (y_i + 4y_{i+1} + y_{i+2})$$

n-1 INTERVALOS:

$$\int_a^b f(x) dx \approx \frac{\Delta x}{3} \left[y_1 + y_n + 2(y_3 + y_5 + y_7 + \dots) + 4(y_2 + y_4 + y_6 + \dots) \right]$$

- NÚMERO DE PUNTOS MUESTRALES NON: n - NON

- ERROR ~ (Δx)⁴

INTERVALO A LA MITAD: Δx → Δx/2

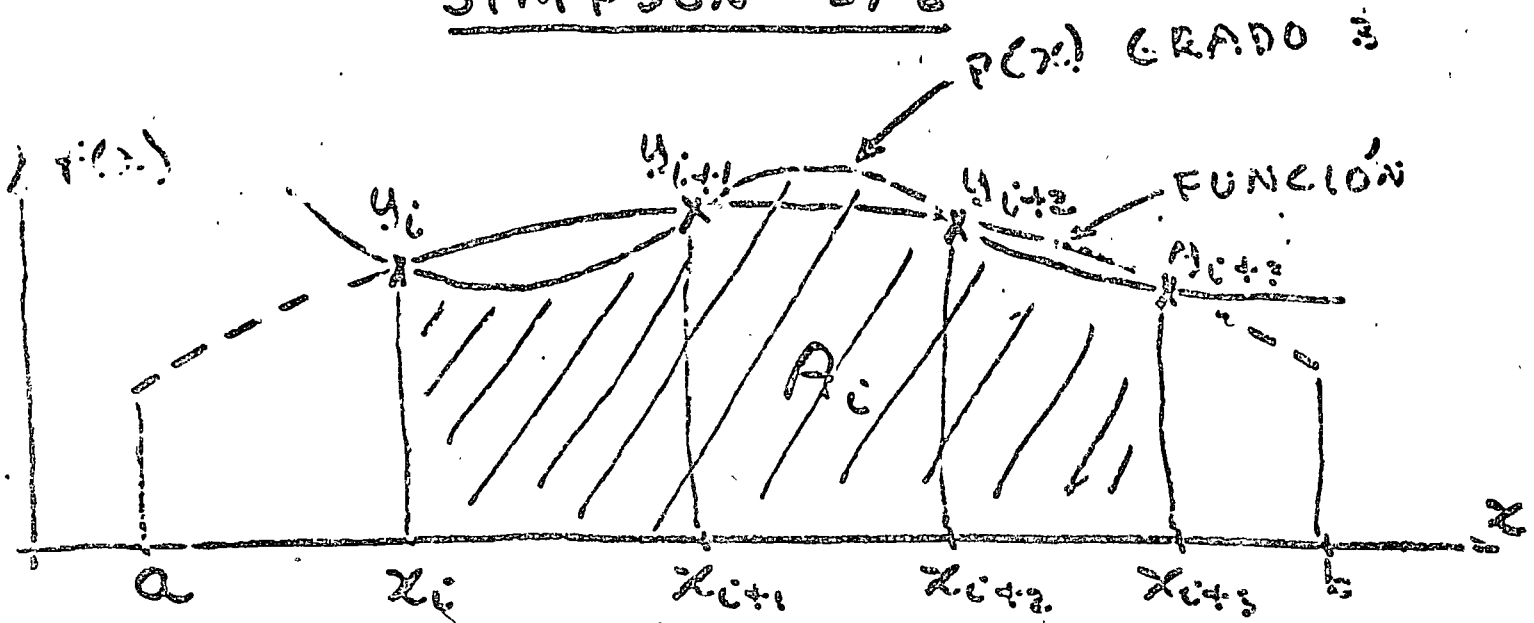
ERROR ENTRE IG: (Δx)⁴ → (Δx/2)⁴ = (Δx)⁴ / 16

- TAMBIEN EXISTE ERROR DE REDONDEO

Δx ↓ / DISMINUYE ERROR TRUNCACION Δx⁴
 / AUMENTA ERROR DE REDONDEO (MAS OPERACIONES)

ES DECIR, Δx NO DEBE SER DEMASIADO

INTEGRACION NUMÉRICA DE SIMPSON 3/8



$$\int_{x_i}^{x_{i+3}} f(x) dx \approx A_i$$

AREA BAJO CURVA \approx AREA BAJO POLINOMIO $p(x)$ DE GRADO 3 QUE APROXIMA LA CURVA $f(x)$

$$f(x) \approx p(x), \quad p(x) = a + bx + cx^2 + dx^3$$

$$\int_{x_i}^{x_{i+3}} f(x) dx \approx A_i = \int_{x_i}^{x_{i+3}} p(x) dx = \frac{3\Delta x}{8} [y_i + 3y_{i+1} + 3y_{i+2} + y_{i+3}]$$

$$\int_a^b f(x) dx \approx \frac{3\Delta x}{8} [y_1 + y_n + 3(y_2 + y_3 + y_5 + y_6 + y_8 + y_9 + \dots) + 2(y_4 + y_7 + y_{10} + \dots)]$$

- NUMERO DE PUNTOS MUESTRALES n :
 $n = (\text{MULTIFLO DE } 3) + 1$

- ERROR $\approx (\Delta x)^4$

MEJOR ORDEN: SIMPSON 3/8
SIMPSON 1/3
TRAPEZOIDAL

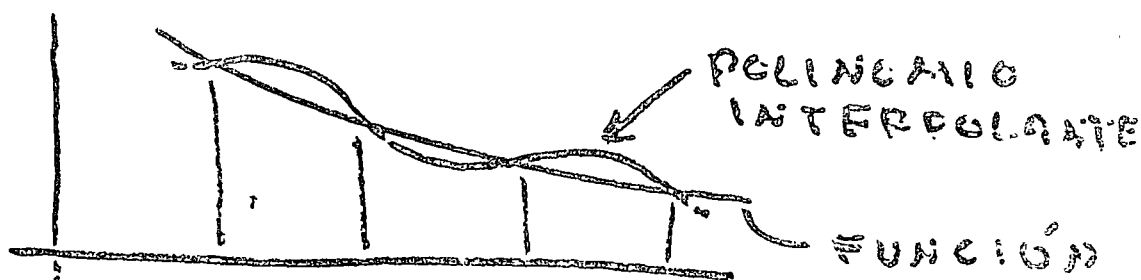
DIFERENCIACIÓN NUMÉRICA

⑥

DIFERENCIACIÓN NUMÉRICA: BÁSICAMENTE MUCHO MENOS PRECISA QUE LA INTEGRACIÓN NUMÉRICA. (∴ SE TRATA DE EVITAR).

- FUNCIONES DEFINIDAS TABULARMENTE O GRÁFICAMENTE

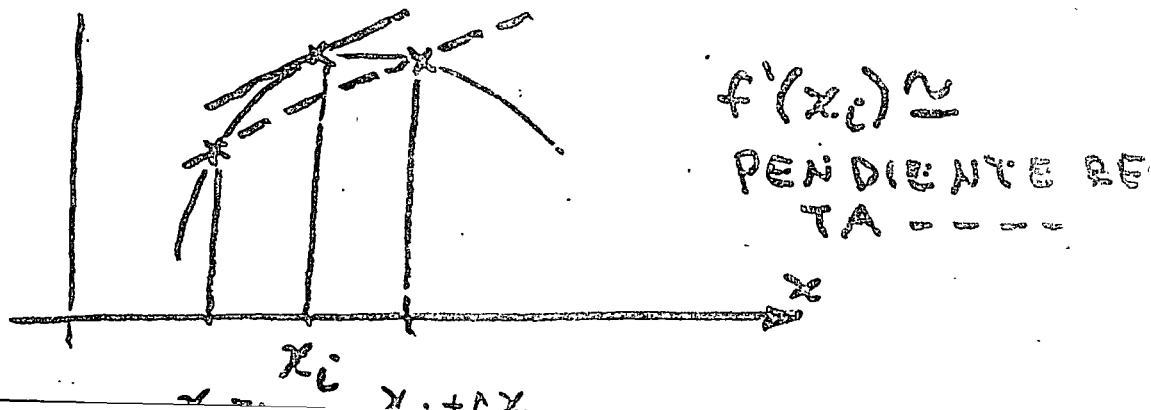
DIFERENCIACIÓN POR POLINOMIOS



{ DERIVADAS (DE $P(x)$) }	BASTANTE DIFERENTES DE	{ DERIVADAS (DE $f(x)$) }
{ INTEGRAL (DE $P(x)$) }	\approx	{ INTEGRAL (DE $f(x)$) }

SE HACE $f(x) \approx P(x)$ ← SE DERIVA

DIFERENCIACIÓN POR SERIE DE TAYLOR:



7

SERIES DE TAYLOR:

$$y(x_i + \Delta x) = y_i + y'_i (\Delta x) + \frac{y''_i (\Delta x)^2}{2} + \dots$$

$$y(x_i - \Delta x) = y_i - y'_i (\Delta x) + \frac{y''_i (\Delta x)^2}{2} + \dots$$

$$y'_i \approx \frac{y(x_i + \Delta x) - y(x_i - \Delta x)}{2 \Delta x}$$

$$y'_i \approx \frac{y_{i+1} - y_{i-1}}{2 \Delta x}$$

APROXIMACIÓN POR DIFERENCIAS CENTRALES DE y' EN x_i .

RAICES DE FUNCIONES

$$3x^2 = 7x + 2$$

$$3x^2 - 7x - 2 = 0$$

POLINOMIO

$$5 \operatorname{sen} 3x = x + 4$$

$$5 \operatorname{sen} 3x - x - 4 = 0$$

TRASCENDENTE

$$\boxed{f(x) = 0}$$

x QUE SATISFACE $f(x) = 0$: RAIZ o CERO DE $f(x)$

1) MÉTODO GRAFICO

$$x^3 - x - 1 = 0$$

$$f(x) = x^3 - x - 1$$

$$f(x) = x^2(x-1) - 1$$

x	$f(x)$
$+\infty$	∞
$-\infty$	$-\infty$
0	-1
1	
2	
-1	

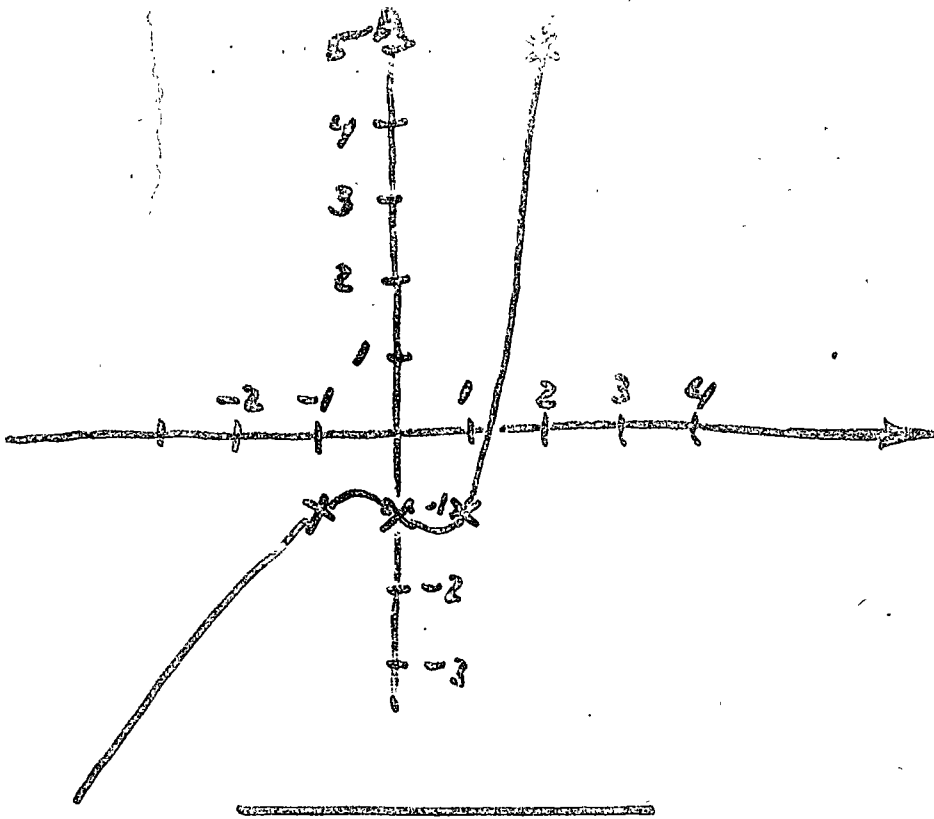
$$f'(x) = 3x^2 - 1$$

$$f'(x) = 0$$

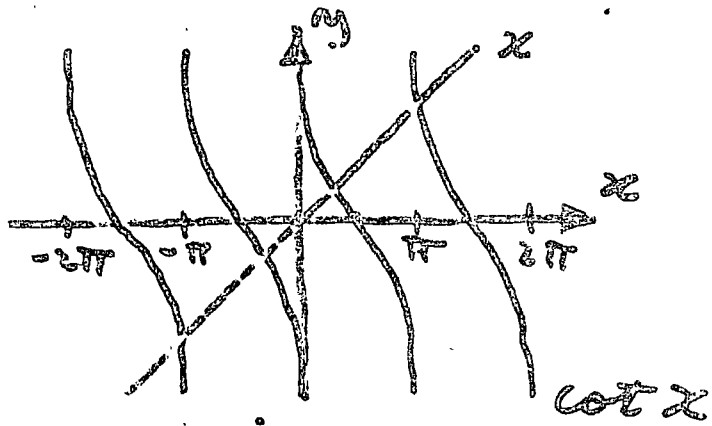
$$3x^2 - 1 = 0$$

$$x^2 = \frac{1}{3}$$

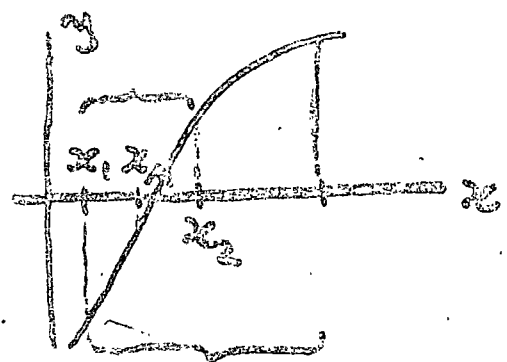
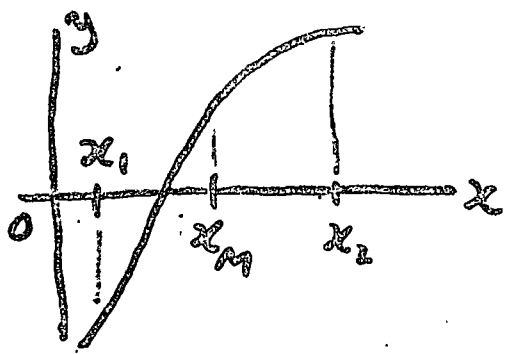
$$x = \pm \frac{1}{\sqrt{3}}$$



$x = \cot x$



2) BISECCIÓN



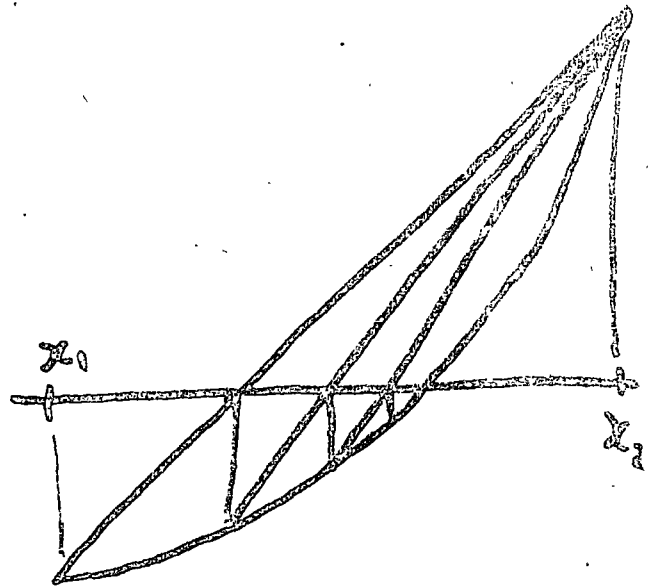
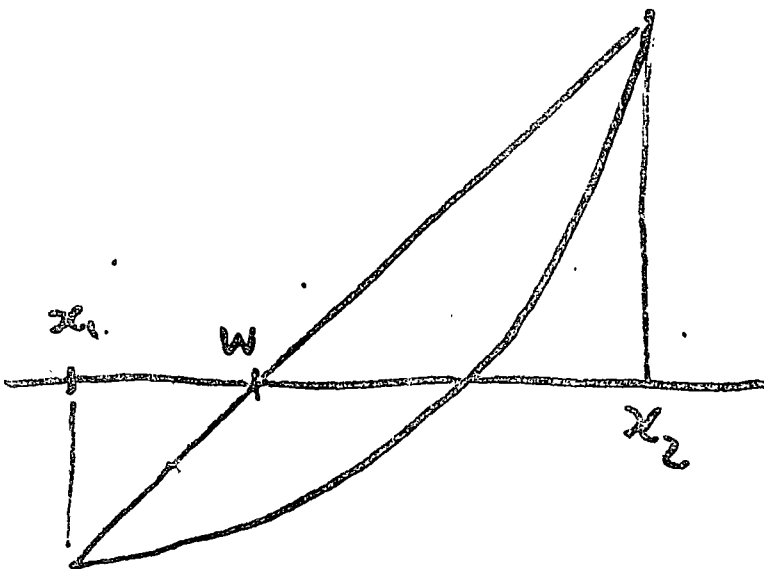
$$x_m = \frac{x_1 + x_2}{2}$$

3) REGULA FALSI (FALSA POSICIÓN) ④

$$f(x) = x^3 - x - 1 = 0$$

$$f(1) = -1 < 0 < 5 = f(2)$$

RAÍZ DEBE ESTAR MAS CERCA DE 1 QUE DE 2



$$w = \frac{f(b_n)a_n - f(a_n)b_n}{f(b_n) - f(a_n)}$$

(PROMEDIO PONDERADO)

{ ALGO MAS RÁPIDO QUE BISECCIÓN
 $|f(x)|$ SE REDUCE
INTERVALO NO SE REDUCE RÁPIDAMENTE

16 ETAPAS:

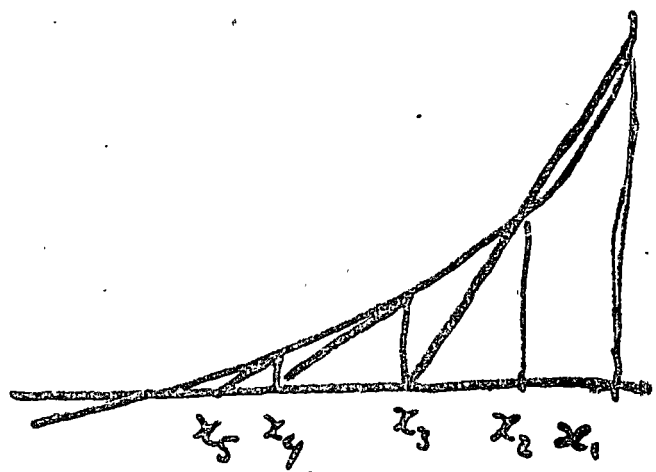
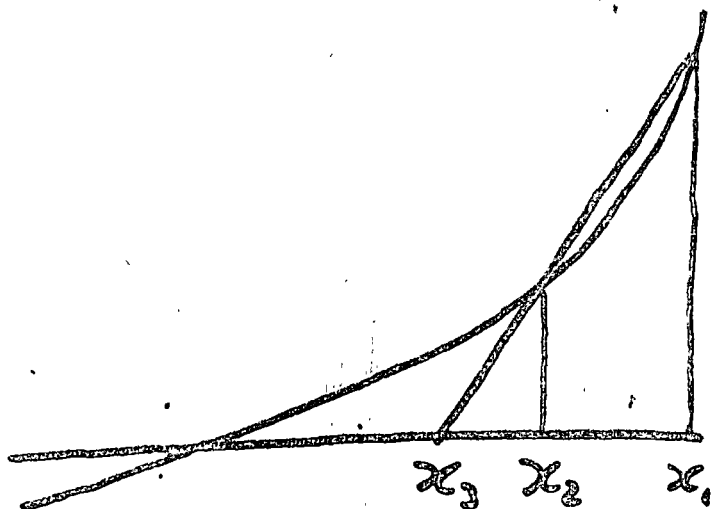
$$1.3247174 \dots \in \text{RAÍZ} \in 2$$

$$f(\overset{\vee}{\quad}) = -1.9 \times 10^{-6} < 0 < 5 = f(2)$$

4) SECANTE

MODIFICACIÓN DE REGULA FALSI:

- NO SE EXAMINAN LOS SIGNOS DE $f(x)$.
- SE EMPLEAN LOS DOS ULTIMOS VALORES x PARA EVALUAR LA SECANTE.



$$x_{n+1} = \frac{f(x_n)x_{n-1} - f(x_{n-1})x_n}{f(x_n) - f(x_{n-1})}$$

$$= x_n - f(x_n) \cdot \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

TERMINO CORRECTIVO

$$\approx \frac{f(x_n)}{f'(x_n)} \quad (\text{NEWTON})$$

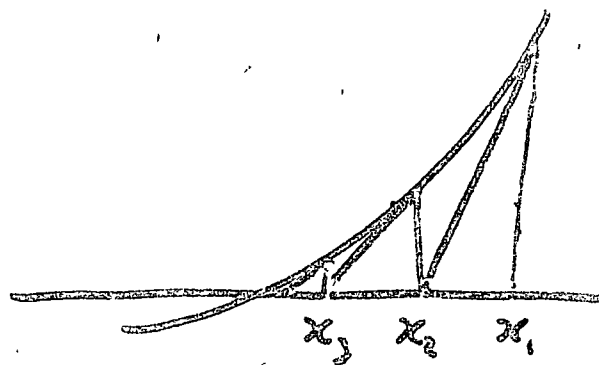
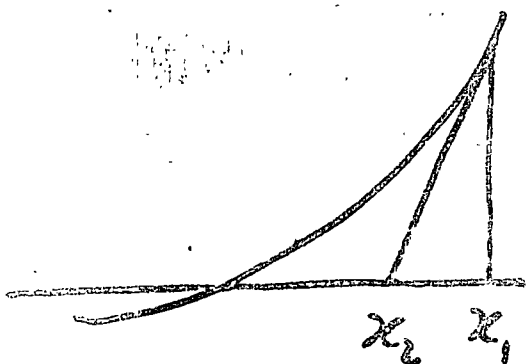
6 ITERACIONES:

$$x_6 = 1.3247179 \dots$$

$$f(x_6) = 3.4 \times 10^{-8}$$

5) NEWTON-RAPHSON

6



$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

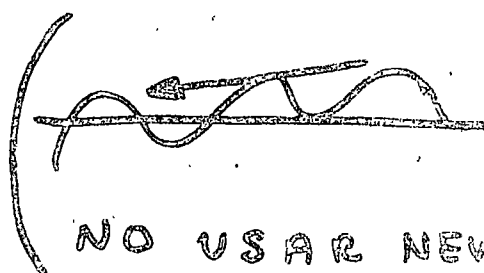
- NECESITA CONOCERSE DERIVADA
- EMPLEA UN SOLO PUNTO POR ITERACIÓN
- CONVERGE RÁPIDAMENTE

$x=1$ ESTÁ MAS CERCA DEL CERO; $x_0=1$

4 ITERACIONES:

$$x_4 = 1.3247181\dots$$

$$f(x_4) = 9.24 \times 10^{-7}$$



NO USAR NEWTON

SOLUCIÓN APROXIMADA x^* :

1) $f(x^*) \approx 0$ ($|f(x^*)|$ PEQUEÑA)

2) $x^* \approx \text{RAIZ}$ ($|x^* - x_n|$ PEQUEÑA)

EJEMPLO

②

ECUACIONES DE LA CINÉTICA PUNTUAL DE UN REACTOR NUCLEAR

$$\frac{d}{dt} \underline{\Psi} = A \underline{\Psi}$$

DONDE

$$\underline{\Psi} = \begin{bmatrix} n(t) \\ c_1(t) \\ c_2(t) \\ c_3(t) \\ c_4(t) \\ c_5(t) \\ c_6(t) \end{bmatrix}, \quad A = \begin{bmatrix} (\rho - \beta)/\Lambda & \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 & \lambda_6 \\ \beta_1/\Lambda & -\lambda_1 & 0 & 0 & 0 & 0 & 0 \\ \beta_2/\Lambda & 0 & -\lambda_2 & 0 & 0 & 0 & 0 \\ \beta_3/\Lambda & 0 & 0 & -\lambda_3 & 0 & 0 & 0 \\ \beta_4/\Lambda & 0 & 0 & 0 & -\lambda_4 & 0 & 0 \\ \beta_5/\Lambda & 0 & 0 & 0 & 0 & -\lambda_5 & 0 \\ \beta_6/\Lambda & 0 & 0 & 0 & 0 & 0 & -\lambda_6 \end{bmatrix}$$

$n(t)$ = POTENCIA DEL REACTOR

$c_i(t)$ = CONCENTRACIÓN DEL PRECURSOR i

ρ = REACTIVIDAD

β_i = FRACCIÓN DE NEUTRONES RETARDADOS GPO. i

β = " TOTAL " " " $\sum \beta_i$

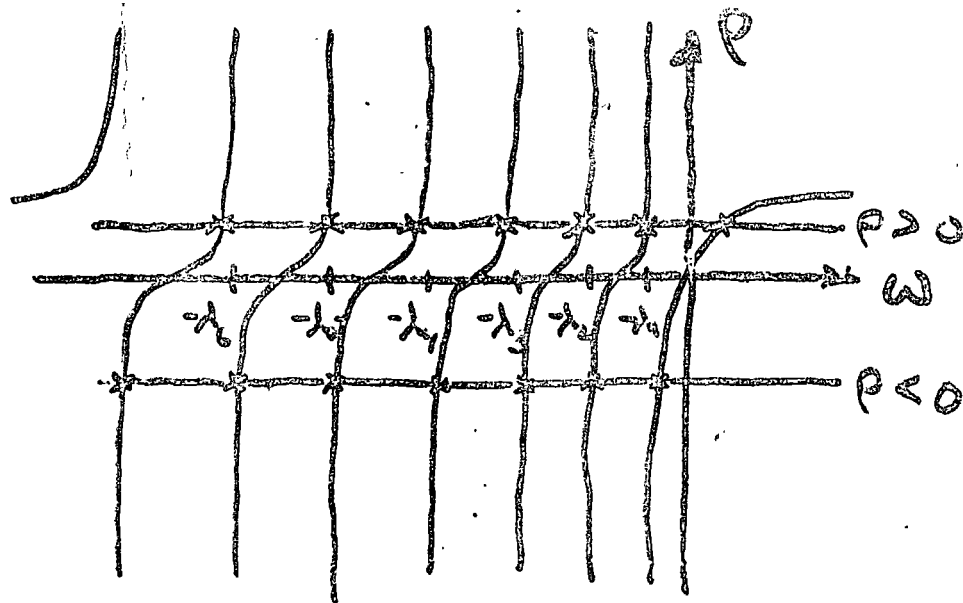
λ_i = CONSTANTE DE DECAIMIENTO PRECURSOR i

Λ = TIEMPO DE GENERACIÓN DE LOS NEUTRONES

LOS EIGENVALORES w_i DE LA MATRIZ A SON RAICES DE LA ECUACIÓN:

$$|A - w I| = 0$$

$$w\Lambda + w \left[\frac{\beta_1}{-\lambda_1 + w} + \frac{\beta_2}{-\lambda_2 + w} + \dots + \frac{\beta_6}{-\lambda_6 + w} \right] = \rho$$



NEWTON - RAPHSON :

$$F(w) = w\Lambda + w \sum_i \frac{\beta_i}{\lambda_i + w} - p$$

$$F'(w) = \Lambda + \sum_i \frac{\beta_i \lambda_i}{(\lambda_i + w)^2}$$

$$\underline{w_{n+1} = w_n - \frac{F(w)}{F'(w)}}$$

OBTENIDAS LOS EIGENVALORES w_i , SE OBTIENEN LOS EIGENVECTORES $\underline{\psi}_i$, Y LA SOLUCIÓN ES:

$$\underline{\psi}(t) = a_1 e^{-w_1 t} \underline{\psi}_1 + \dots + a_n e^{-w_n t} \underline{\psi}_n$$

RAÍCES DE POLINOMIOS

- SE PUEDEN APLICAR LOS METODOS ANTERIORES
- COMO SE DEBE EVALUAR EL POLINOMIO MUCHAS VECES, CONVIENE HACERLO EFICIENTEMENTE (FORMA ENCAJADA):

$$P(x) = a_0 + a_1x + a_2x^2 + a_3x^3 = a_0 + x(a_1 + x(a_2 + x(a_3)))$$

n ADICIONES
 $\frac{N(N+1)}{2}$ MULTIPLIC.

n ADICIONES
 n MULTIPLIC.

GRADO
= $n = 10$

$$\begin{array}{r} 10 \\ + 55 \\ \hline 65 \end{array}$$

$$\begin{array}{r} 10 \\ + 10 \\ \hline 20 \end{array}$$

$$P(x) = \underset{q_n}{q}(x) \underset{q_{n-1}}{(x-z)} + b_0$$

$$P'(x) = q(x) + q'(x)(x-z)$$

INTERPOLACIÓN

- ≠ AJUSTE DE CURVAS
- APROXIMAR UNA FUNCIÓN POR OTRA FUNCIÓN MÁS SENCILLA (PARA PODER INTEGRAR Y DIFERENCIAR MÁS FÁCILMENTE, POR EJEMPLO)
- INTERPOLAR EN TABLAS (CALCULAR FUNCIÓN EN PUNTOS NUEVOS)
- FUNCIONES INTERPOLANTES: POLINOMIOS ←
 FUN. TRIGONOMETRICAS
 EXPONENCIALES
 RACIONALES

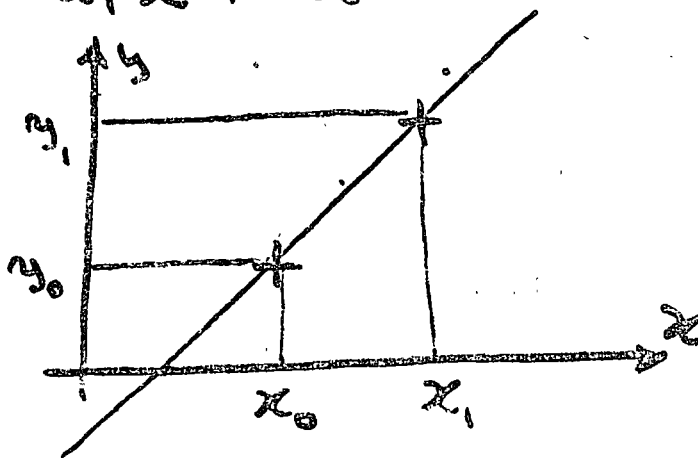
POLINOMIO INTERPOLANTE: LAGRANGE

INTERPOLACIÓN 2 PUNTOS (LINEAL):

$$y - y_0 = \frac{y_1 - y_0}{x_1 - x_0} (x - x_0)$$

$$y = p(x) = \frac{y_1 - y_0}{x_1 - x_0} x + \left(y_0 - \frac{y_1 - y_0}{x_1 - x_0} x_0 \right)$$

$$= a_1 x + a_0$$



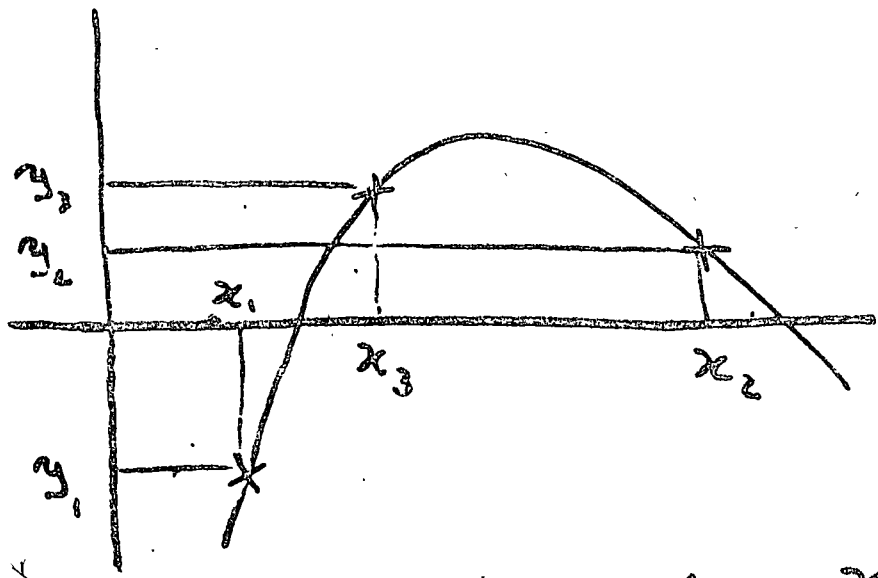
POLINOMIOS DE LAGRANGE

$$L_0: \frac{(x - x_1)}{x_0 - x_1} \begin{cases} = 1 & x = x_0 \\ = 0 & x = x_1 \end{cases}$$

$$x_1: \frac{x-x_0}{x_1-x_0} \begin{cases} = 0 & x=x_0 \\ = 1 & x=x_1 \end{cases}$$

$$p(x) = y_0 \frac{(x-x_1)}{(x_0-x_1)} + y_1 \frac{(x-x_0)}{(x_1-x_0)}$$

INTERPOLACIÓN 3 PUNTOS (CUADRÁTICA)



POLINOMIOS DE LAGRANGE:

$$x_1: \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} \begin{cases} = 1 & x=x_1 \\ = 0 & x=x_2 \\ = 0 & x=x_3 \end{cases}$$

$$x_2: \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} \begin{cases} = 0 & x=x_1 \\ = 1 & x=x_2 \\ = 0 & x=x_3 \end{cases}$$

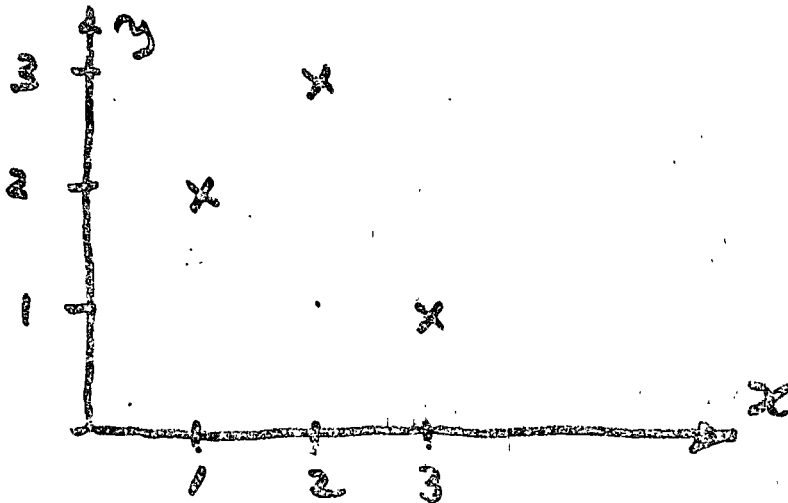
$$x_3: \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} \begin{cases} = 0 & x=x_1 \\ = 0 & x=x_2 \\ = 1 & x=x_3 \end{cases}$$

$$P(x) = y_1 \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + y_2 \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + y_3 \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}$$

$$\begin{cases} P(x_1) = y_1 \cdot 1 + y_2 \cdot 0 + y_3 \cdot 0 = y_1 \\ P(x_2) = y_1 \cdot 0 + y_2 \cdot 1 + y_3 \cdot 0 = y_2 \\ P(x_3) = y_1 \cdot 0 + y_2 \cdot 0 + y_3 \cdot 1 = y_3 \end{cases}$$

$$p(x) = a_0 + a_1 x + a_2 x^2$$

EJEMPLO:



$$P(x) = 2 \frac{(x-2)(x-3)}{(1-2)(1-3)} + 3 \frac{(x-1)(x-3)}{(2-1)(2-3)} + 1 \frac{(x-1)(x-2)}{(3-1)(3-2)}$$

$$= 2 \frac{(x-2)(x-3)}{(-1)(-2)} + 3 \frac{(x-1)(x-3)}{1 \cdot (-1)} + 1 \frac{(x-1)(x-2)}{2 \cdot 1}$$

$$P(x) = -\frac{1}{2}x^2 + \frac{11}{2}x - 2$$

CHECANDO:

$$P(x) = -\frac{3}{2}x^2 + \frac{11}{2}x - 2$$

$$P(1) = -\frac{3}{2} \cdot 1^2 + \frac{11}{2} \cdot 1 - 2 = 2$$

$$P(2) = -\frac{3}{2} \cdot 2^2 + \frac{11}{2} \cdot 2 - 2 = 3$$

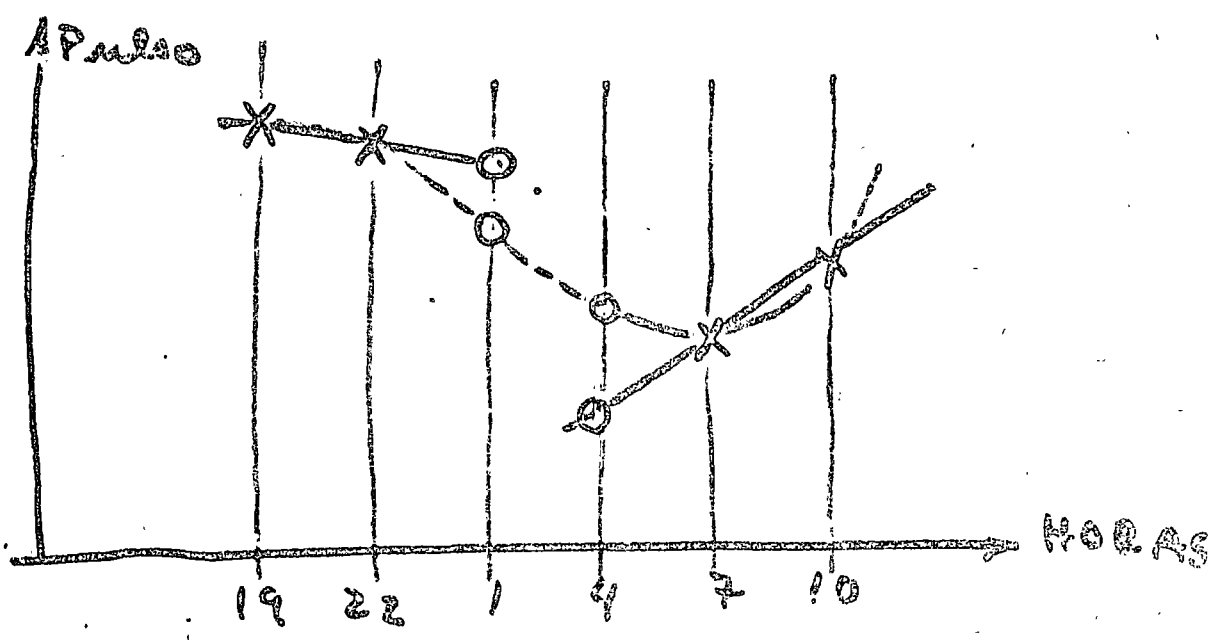
$$P(3) = -\frac{3}{2} \cdot 3^2 + \frac{11}{2} \cdot 3 - 2 = 1$$

APLICACIÓN: RITMOS BIOLÓGICOS

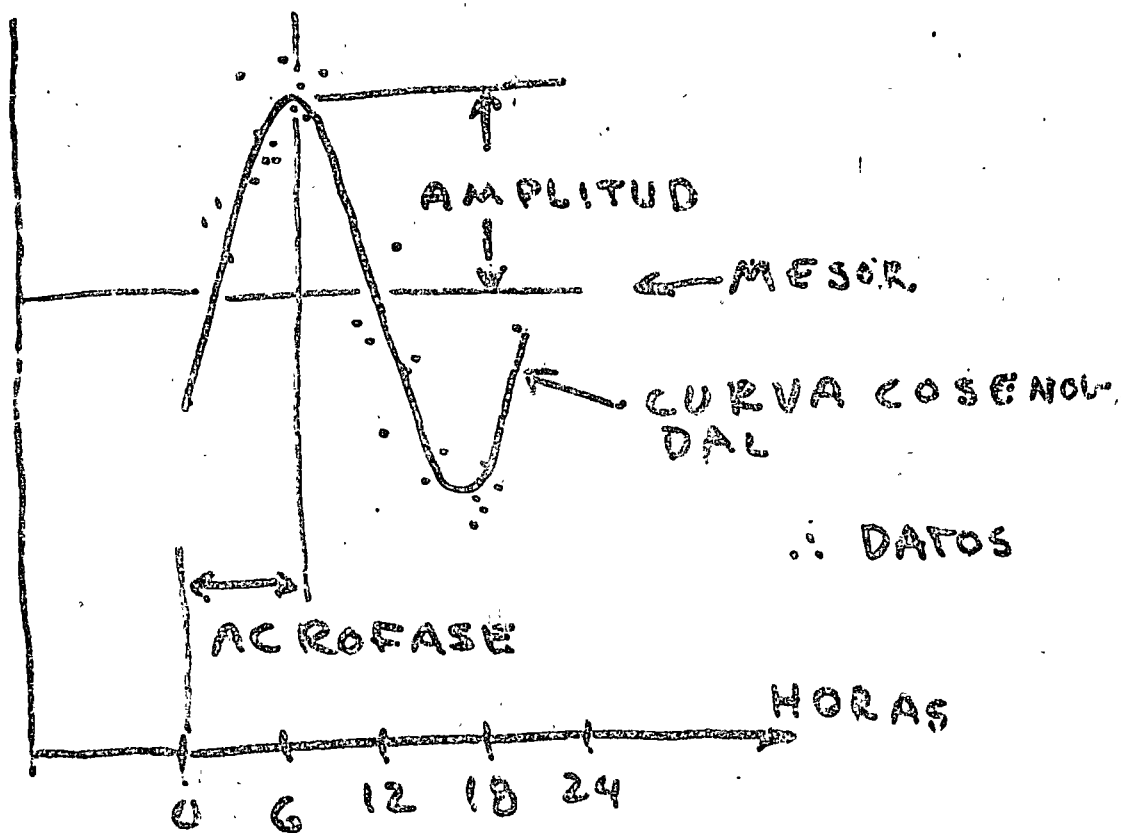
LATIDOS DEL CORAZÓN, RESPIRACIÓN,
ACTIVIDAD DIARIA HÍGADO Y RIÑONES,
MOVIMIENTO DIARIO HOJAS, ESTRUS ANUAL.

- RELOJ BIOLÓGICO

(HOJAS MOVIMIENTO DIARIO EN OSC.
HAMSTERS RITMO DIARIO CON LUCECONT.)



5



AJUSTE CURVA COSENOIDAL POR
MÍNIMOS CUADRADOS (FOURIER)

$$f(x) = a_0 + a_1 \cos \omega t + a_2 \cos 2\omega t + \dots \\ + b_1 \sin \omega t + b_2 \sin 2\omega t + \dots$$

$$= a_0 + c \cos(\omega t + \phi)$$

MESOR AMPLITUD ACROFASE

CUANDO SE INTERPOLAN n PUNTOS

$$p(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_{n-1} x^{n-1}$$

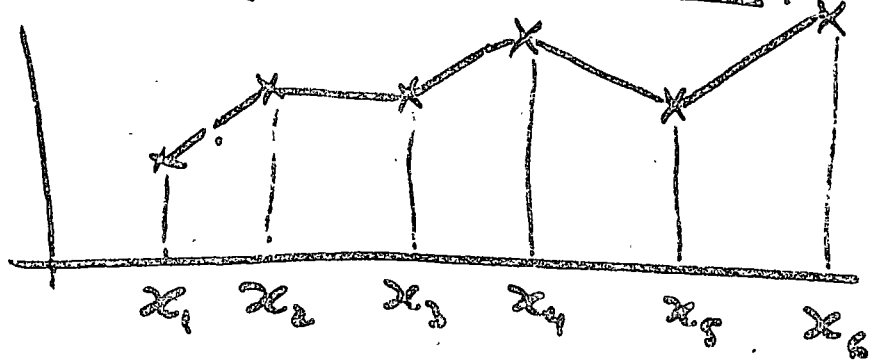
- n PARÁMETROS a_0, a_1, \dots, a_{n-1} .
- POLINOMIO GRADO $n-1$

SI n ES MUY GRANDE, $p(x)$ OSCILA DEMASIADO.

SOLUCIÓN: INTERPOLACION POR PARTES

INTERPOLACIÓN DE LAGRANGE
POR PARTES

INTERPOLACIÓN LINEAL P.P.



LAS RECTAS SE FORMAN TOMANDO DOS PUNTOS A LA VEZ:

- x_1, x_2
- x_2, x_3
- x_3, x_4



centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



METODOS NUMERICOS Y APLICACIONES CON LA COMPUTADORA
DIGITAL

TEMA 3 : SISTEMAS DE ECUACIONES LINEALES.

ABRIL, 1978.

3. SOLUCION DE SISTEMAS DE ECUACIONES LINEALES

3.1. Introducción

Por sistemas de ecuaciones lineales se entiende un grupo de ecuaciones que presentan la siguiente estructura:

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \cdot \\ \cdot \\ \cdot \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \end{array} \right\} \quad (3.1)$$

donde a_{ij} y b_i son constantes y las incógnitas del sistema son los valores x_i , donde $1 \leq i \leq n$.

Dichos sistemas se pueden representar en la forma:

$$\underline{A} \underline{X} = \underline{B} \quad (3.2)$$

donde \underline{A} se conoce como la matriz de coeficientes del sistema, \underline{B} como vector de términos independientes y \underline{X} como vector de incógnitas.

Si el vector de términos independientes es diferente de cero se habla de sistemas de ecuaciones no homogéneas y en caso contrario de sistemas homogéneos.

Antes de proceder a resolver un sistema de ecuaciones es necesario determinar si dicho sistema tiene solución y en caso de tenerla, cuántas posibles soluciones tiene. En base a lo anterior se tiene la siguiente clasificación:

$$\text{Sistema de ecuaciones lineales} \left\{ \begin{array}{l} \text{no homogéneo} \left\{ \begin{array}{l} \text{compatible} \left\{ \begin{array}{l} \text{determinado} \\ \text{indeterminado} \end{array} \right. \\ \text{incompatible} \end{array} \right. \\ \\ \text{homogéneo} \left\{ \begin{array}{l} \text{compatible} \left\{ \begin{array}{l} \text{determinado} \\ \text{(Sol. trivial)} \\ \text{indeterminado} \end{array} \right. \end{array} \right. \end{array} \right.$$

Sistema compatible es aquél que sí tiene solución y para que esto se cumpla se requiere:

$$\text{rango } [A] = \text{rango } [A|B] \quad * \quad (3.3)$$

donde a la matriz $[A|B]$ se le conoce como la matriz ampliada del sistema.

Sistema incompatible es aquél que no tiene solución y se cumple que:

$$\text{rango } [A] < \text{rango } [A|B] \quad (3.4)$$

Sistema determinado es un sistema compatible que presenta solución única y se verifica que:

$$\text{rango } [A] = \text{número de incógnitas} \quad (3.5)$$

Cuando se presenta esta situación en sistemas homogéneos se habla de solución trivial, ya que $\underline{X} = \underline{0}$.

Un sistema compatible que presenta infinidad de soluciones se conoce como sistema indeterminado y se caracteriza por:

$$\text{rango } [A] < \text{número de incógnitas} \quad (3.6)$$

Para la solución de sistemas de ecuaciones lineales existen diversos métodos de los cuales solo se tratarán: Método de Gauss-Jordan modificado y el Método de Gauss-Seidel.

3.2 Método de Gauss-Jordan modificado.

3.2.1 Objeto

Obtener la solución de sistemas de ecuaciones lineales de la forma:

$$\underline{A} \underline{X} = \underline{B} \quad (3.7)$$

3.2.2 Método

Dado el sistema de ecuaciones:

$$\underline{A} \underline{X} = \underline{B} \quad (3.8)$$

* rango $[A]$ es la cantidad de vectores linealmente independientes del conjunto de vectores columna que forman la matriz A .

el método consiste en trabajar con la matriz de coeficientes y el vector de términos independientes, es decir, con la matriz ampliada del sistema:

$$\left[\underline{A} \mid \underline{B} \right] \quad (3.9)$$

A dicha matriz se le aplican una serie de transformaciones que conducen a obtener otra matriz ampliada equivalente:

$$\left[\underline{I}_n \mid \underline{C} \right] \quad (3.10)$$

donde \underline{C} representa la solución de cada una de las incógnitas del sistema.

El proceso equivale a premultiplicar la ecuación (3.9) por \underline{A}^{-1} , es decir, el método de la matriz inversa, solo que este método consiste en una eliminación sistemática de valores.

La transformación de la matriz (3.9) en la matriz (3.10) se efectúa basándose en tres operaciones que no alteran el sistema de ecuaciones sino que proporcionan sistemas de ecuaciones equivalentes, ellas son:

- intercambio de dos renglones, lo cual equivale a intercambiar dos ecuaciones;
- multiplicación de un renglón por un escalar diferente de cero, lo cual equivale a multiplicar ambos miembros de una ecuación por la misma constante.
- suma de equimúltiplos de un renglón a otro renglón, es decir, multiplicar una ecuación por una constante "K" y sumarla a otra ecuación.

Para aplicar las operaciones anteriores se procede en la siguiente forma:

- ① Seleccionar un renglón pivote y un elemento pivote dentro de dicho renglón.
- ② Normalizar el elemento pivote, es decir, convertirlo en unitario.
- ③ Cancelar elementos que se encuentren en la columna arriba y/o abajo del elemento pivote mediante la suma de equimúltiplos.
- ④ Regresar al paso ① y así sucesivamente hasta que se transforma la matriz de coeficientes \underline{A} en una matriz --

identidad I_n .

Debido a que durante el proceso se presentan errores por redondeo, la forma óptima de escoger los elementos pivote es seleccionando el mayor elemento que quede en la matriz A o en sus transformaciones. Hay que tener presente que los elementos de un renglón que ya fue seleccionado como línea pivote no se pueden usar como pivotes, aún cuando el mayor elemento quede colocado en dicho renglón.

Al seleccionar los pivotes en la forma antes mencionada el error se reduce al mínimo y, debido a que puede quedar una matriz no identidad al término de las iteraciones, es necesario efectuar un intercambio de líneas hasta obtener I_n .

Cabe mencionar que el presente método es un método directo de solución que no requiere que se determine con anterioridad si el sistema es compatible y determinado, el método durante el proceso proporciona dicha información.

Si el sistema es compatible y determinado, el procedimiento descrito se puede llevar a cabo sin contratiempos hasta llegar a $[I_n | C]$.

Si el sistema es compatible pero indeterminado, la matriz ampliada adquirirá la configuración:

$$\left[\begin{array}{ccc|c} 1 & 0 & 2 & 1 \\ 0 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 \end{array} \right] \quad (3.11)$$

es decir, un renglón será nulo; en esta situación se obtienen las ecuaciones independientes que restan en el sistema y se aplica la metodología correspondiente a sistemas indeterminados.

Si el sistema es incompatible, se presentará lo siguiente:

$$\left[\begin{array}{ccc|c} 1 & 1 & 2 & 1 \\ 0 & 2 & 3 & 2 \\ 0 & 0 & 0 & \lambda \neq 0 \end{array} \right] \quad (3.12)$$

o sea, $0 = \lambda \neq 0$, lo cual es una contradicción.

3.2.3 Descripción del Programa

a) Subrutinas requeridas:

SUBROUTINE GAUTOR (A, B, N, EPS, DET), esta subrutina obtiene la solución del sistema de ecuaciones por el método de Gauss-Jordan modificado, el programa principal solo sirve para entrada y salida de datos.

b) Descripción de las variables:

Para la subrutina GAUTOR:

A(I, J) matriz de coeficientes del sistema de ecuaciones.

B(I) vector de términos independientes del sistema de ecuaciones, durante el proceso se transforma en la solución.

N orden del sistema de ecuaciones.

RAMAX mayor elemento de la matriz A que se emplea como pivote.

MVR(I) y MVC(I) contadores que indican qué renglón y columnas ya fueron empleados.

EPS criterio para determinar si el determinante de la matriz A es nulo.

DET parámetro que indica si el determinante de A es nulo.

LR y LC indicadores del renglón y columna que se utilizan.

TEMP variable de localización temporal.

Para el programa principal:

A(I, J) matriz de coeficientes del sistema de ecuaciones.

B(I) vector de términos independientes.

N orden del sistema de ecuaciones.

EPS criterio para determinar si el determinante de A es nulo.

DET parámetro que indica si el determinante de A es nulo.

c) Dimensiones:

La proposición DIMENSION del programa principal y de la subrutina se deberán modificar en el caso de que:

$$N > 10$$

d) Formatos para los datos de entrada:

SEC. TARJETAS	FORMATO	INFORMACION
1	(I5)	N
2	(8F10.0)	A(I,J), se dan los <u>elemen</u> <u>tos</u> de <u>A</u> renglón por ren- glón, empleando tantas -- tarjetas como sean <u>neces</u> <u>arias</u> para cada renglón.
3	(8F10.0)	B(I), el vector de térmi- nos independientes se da en una tarjeta o más se-- gún la cantidad de <u>elemen</u> <u>tos</u> .

 otros paquetes de datos (opcional)

n

TARJETA EN BLANCO, al fi
nalizar toda la informa
ción.

e) Diagrama de bloques:

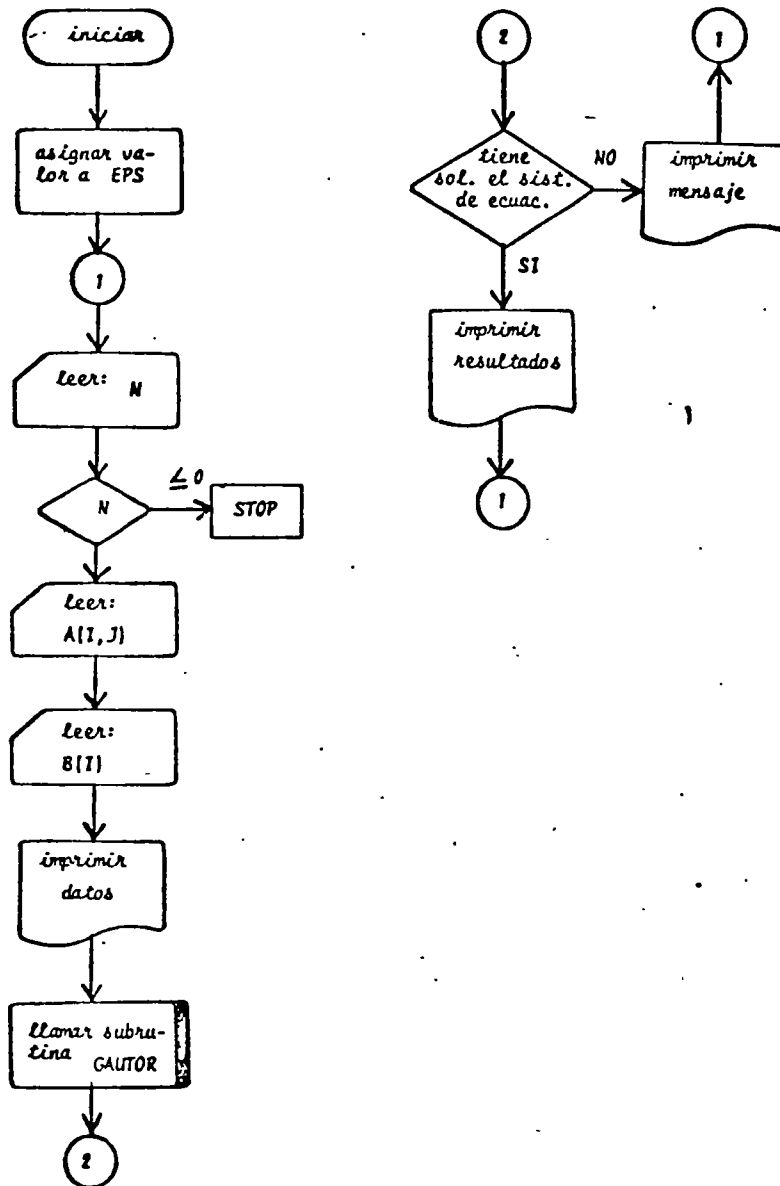


Fig. 3.1 Diagrama de bloques del programa principal.

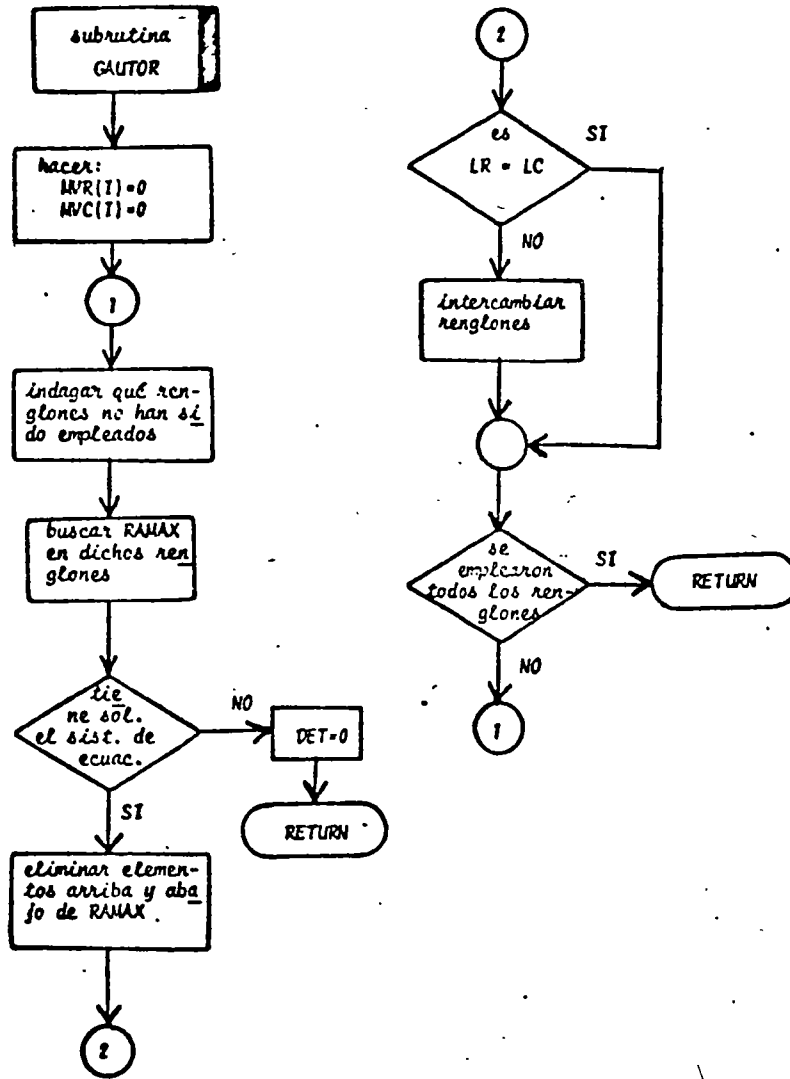


Fig. 3.2 Diagrama de bloques de la subrutina GAUTOR.

6) Listado:

```

C   PROGRAMA PARA RESOLVER SISTEMAS DE ECUACIONES LINEALES POR EL METO
C   DO DE GAUSS-JORDAN
C   SIGNIFICADO DE LAS VARIABLES EMPLEADAS
C   N=ORDEN DEL SISTEMA DE ECUACIONES
C   A=MATRIZ DE COEFICIENTES DEL SISTEMA DE ECUACIONES
C   B=VECTOR DE TERMINOS INDEPENDIENTES, SE CONVIERTE EN LA SOLUCION
C   CRITERIO PARA DETERMINAR SI EL DETERMINANTE DE A ES DIFERENTE DE 0
C   DET=VARIABLE QUE INDICA SI EL SISTEMA TIENE O NO SOLUCION

DIMENSION A(10,10),B(10)
IR=5
IN=6
EPS=0.00001
C   LECTURA DE DATOS
1  READ(I,20) N
   IF(N) 2,2,3
2  CALL EXIT
3  DO 4 I=1,N
4  READ(I,21) (A(I,J),J=1,N)
   READ(I,21) (B(I),I=1,N)
C   IMPRESION DE DATOS
   WRITE(I,22)
   GO 5 I=1,N
5  WRITE(I,23) (A(I,J),J=1,N),B(I)
C   LLAMADO DE SUBROUTINA PARA RESOLVER EL SISTEMA DE ECUACIONES
   CALL GALTOR(A,B,N,EPS,DET)
C   IF(DET.LE.EPS) GO TO 7
C   IMPRESION DE RESULTADOS
   WRITE(I,24)
   DO 6 I=1,N
6  WRITE(I,25) I,B(I)
   GO TO 1
7  WRITE(I,26)
   GO TO 1
C   FORMATS DE LECTURA E IMPRESION
20  FORMAT(I5)
21  FORMAT(2F10.0)
22  FORMAT(4(//),5X,'EL SISTEMA DE ECUACIONES ES',//)
23  FORMAT(//,2X,10(E10.3,1X))
24  FORMAT(4(//),5X,'LA SOLUCION DEL SISTEMA DE ECUACIONES ES',//,5X,'1
1',5X,'X(I)',//)
25  FORMAT(//,5X,12,4X,E12.5)
26  FORMAT(4(//),5X,'EL SISTEMA DE ECUACIONES NO TIENE SOLUCION')
END

```

Fig. 3.3 Listado del programa principal

```

SUBROUTINE GAUTOR(A,N,EPS,DET)
C
C SUBROUTINA PARA RESOLVER UN SISTEMA DE ECUACIONES POR EL METODO DE
C GAUSS-JORDAN MODIFICADO
C EL SIGNIFICADO DE LAS VARIABLES EMPLEADAS ES
C A=MATRIZ DE COEFICIENTES DEL SISTEMA DE ECUACIONES
C B=VECTOR DE TERMINOS INDEPENDIENTES QUE DURANTE EL PROCESO SE
C TRANSFORMA EN LA SOLUCION DEL SISTEMA DE ECUACIONES
C N=ORDEN DEL SISTEMA DE ECUACIONES
C RAMAX=MAYOR ELEMENTO DE LA MATRIZ A QUE SE USA COMO PIVOTE
C MVR Y MVC=CONTADORES QUE INDICAN QUE RENGLON Y QUE COLUMNA YA FUE-
C RON UTILIZADOS
C EPS=CRITERIO PARA DETERMINAR SI EL DETERMINANTE DE LA MATRIZ A ES
C NULO
C DET=VALOR ABSOLUTO DEL DETERMINANTE DE LA MATRIZ A
C
C DIMENSION A(10,10),B(10),MVR(10),MVC(10)
C
C ACTUALIZACION DE VALORES PARA INICIAR EL PROCESO
C
C DO 1 I=1,N
MVR(I)=0
1 MVC(I)=0
C
C SOLUCION DEL SISTEMA DE ECUACIONES
C
C DO 9 K=1,N
RAMAX=0.0
LC=0
LR=0
DO 3 I=1,N
IF(MVR(I).EQ.1) GO TO 3
DO 2 J=1,N
IF(MVC(J).EQ.1) GO TO 2
IF(ABS(RAMAX).GE.ABS(A(I,J))) GO TO 2
RAMAX=A(I,J)
LR=I
LC=J
2 CONTINUE
3 CONTINUE
DET=DET*A(RAMAX)
IF(ABS(DET).LE.EPS) GO TO 10
IF(LR.EQ.LC) GO TO 5
DO 4 I=1,N
TEMP=A(LR,I)
A(LR,I)=A(LC,I)
4 A(LC,I)=TEMP
TEMP=B(LR)
B(LR)=B(LC)
B(LC)=TEMP
5 DO 6 I=1,N
6 A(LC,I)=A(LC,I)/RAMAX
B(LC)=B(LC)/RAMAX
DO 8 I=1,N
IF(I.EQ.LC) GO TO 8
TEMP=A(I,LC)
B(I)=B(I) - TEMP*B(LC)
DO 7 J=1,N
7 A(I,J)=A(I,J) - TEMP*A(LC,J)
8 CONTINUE
MVR(LC)=LC
MVC(LC)=LC
9 CONTINUE
10 RETURN
END

```

Fig. 3.4 Listado de la subrutina GAUTOR

3.2.4 Ejemplo

Empleando las leyes de Kirchoff (ver referencia 2), se obtuvieron las siguientes ecuaciones lineales para el circuito mostrado en la figura 3.5:

$$\begin{aligned}
 i_8 - i_4 - I_A &= 0 \\
 i_4 + i_5 + I_A - i_1 - i_3 &= 0 \\
 i_1 - i_2 - I_B &= 0 \\
 i_2 + I_B + i_3 + i_6 - i_7 &= 0 \\
 I_C - i_8 - i_5 - i_6 - i_9 &= 0 \\
 R_1 i_1 + R_2 i_2 - R_3 i_3 &= 0 \\
 R_4 i_4 - R_5 i_5 + R_8 i_8 &= 0 \\
 R_5 i_5 + R_3 i_3 - R_6 i_6 &= 0 \\
 R_6 i_6 + R_7 i_7 - R_9 i_9 &= 0
 \end{aligned}$$

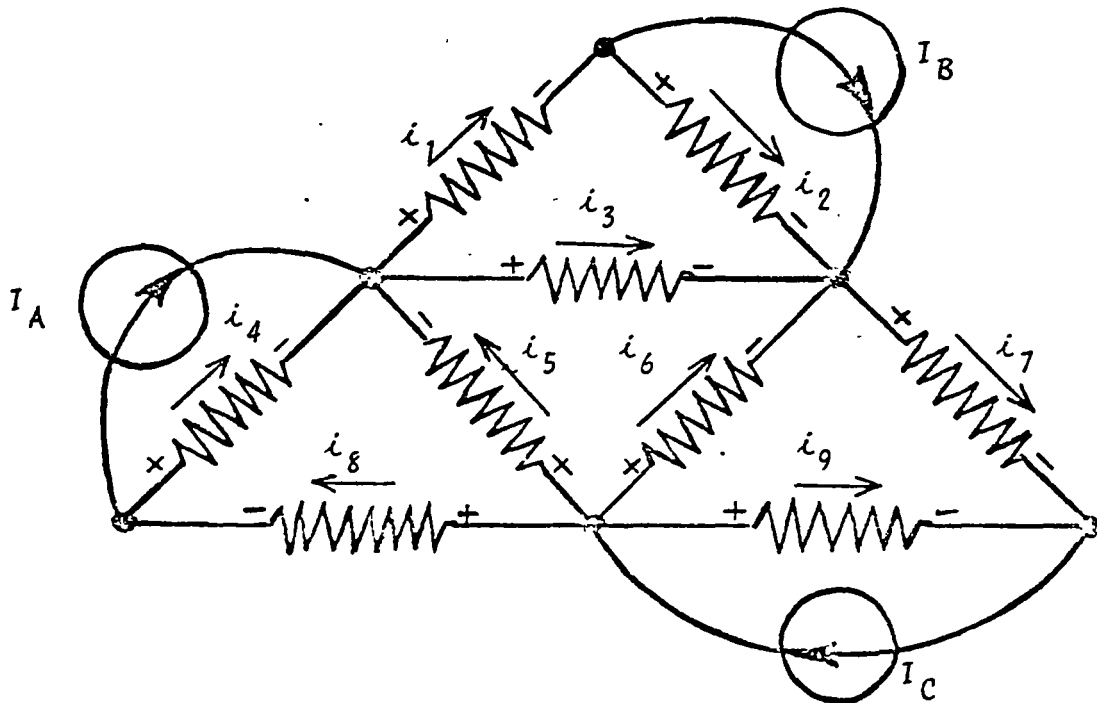


Fig. 3.5 Circuito del ejemplo 3.2.4

Si el valor de las fuentes es $I_A = 2A$, $I_B = 6A$, $I_C = 4A$ y el de las resistencias:

$$R_1 = R_2 = 2 \Omega$$

$$R_4 = R_8 = 3 \Omega$$

$$R_5 = R_6 = 5 \Omega$$

$$R_7 = R_9 = 4 \Omega$$

$$R_3 = 6 \Omega$$

Obtenga las corrientes de rama $i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9$.

* SOLUCION

TABLA 3.1 Datos para el problema del ejemplo 3.2.4

$$N = 9$$

$$A = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 2 & 2 & -6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & -5 & 0 & 0 & 3 & 0 \\ 0 & 0 & 6 & 0 & 5 & -5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 & 4 & 0 & -4 \end{bmatrix}$$

$$B = \begin{bmatrix} 2 \\ -2 \\ 6 \\ -6 \\ 4 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

TABLA 3.2 Resultados del problema del ejemplo 3.2.4

EL SISTEMA DE ECUACIONES ES

0.	C.	0.	-.100E+01	0.	0.	0.	.100E+01	0.	.200E+01
-.100E+01	0.	-.100E+01	.100E+01	.100E+01	0.	0.	0.	0.	-.200E+01
.100E+01	-.100E+01	0.	0.	0.	0.	0.	0.	0.	.600E+01
0.	.100E+01	.100E+01	0.	0.	.100E+01	-.100E+01	0.	0.	-.600E+01
0.	C.	0.	0.	.100E+01	.100E+01	0.	.100E+01	.100E+01	.400E+01
.200E+01	.200E+01	-.600E+01	0.	0.	0.	0.	0.	0.	0.
0.	C.	0.	.300E+01	-.500E+01	0.	0.	.300E+01	0.	C.
0.	C.	.600E+01	C.	.500E+01	-.500E+01	0.	0.	0.	C.
0.	0.	0.	0.	0.	.500E+01	.400E+01	0.	-.400E+01	C.

LA SOLUCION DEL SISTEMA DE ECUACIONES ES

I X(I)

1	.23761E+01
2	-.36239E+01
3	-.41596E+00
4	-.56359E+00
5	.52370E+00
6	.24548E+01
7	.19847E+01
8	.14364E+01
9	.20153E+01

3.3 Método de Gauss-Seidel

3.3.1 Objeto

Obtener la solución de sistemas de ecuaciones lineales -- con la configuración:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ \cdot & \cdot \\ \cdot & \cdot \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned} \quad (3.13)$$

empleando el método de Gauss-Seidel.

3.3.2 Método

El método de Gauss-Seidel es un método de tipo iterativo que sirve para la solución de sistemas de ecuaciones lineales del tipo:

$$\underline{A} \underline{X} = \underline{B} \quad (3.14)$$

cuando los valores numéricos de los elementos de la diagonal principal son mayores que los demás de su correspondiente renglón.

Para asegurar la convergencia del método se requiere que:

- los elementos no nulos de la matriz de coeficientes (A) se acumulen en la diagonal principal.
- los elementos de la diagonal principal de la matriz de coeficientes (A) sean mayores en valor absoluto que la sumatoria de los valores absolutos de los elementos -- restantes del renglón correspondiente, es decir:

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \quad i = 1, 2, \dots, n \quad (3.15)$$

Para aplicar el método se procede a despejar una incógnita

ta de cada ecuación del arreglo (3.13), es decir, despejar la incógnita x_i de la "i-ésima" ecuación, o sea:

$$\left. \begin{aligned} x_1 &= \frac{1}{a_{11}} \left[b_1 - a_{12}x_2 - a_{13}x_3 - \dots - a_{1n}x_n \right] \\ x_2 &= \frac{1}{a_{22}} \left[b_2 - a_{21}x_1 - a_{23}x_3 - \dots - a_{2n}x_n \right] \\ &\vdots \\ x_n &= \frac{1}{a_{nn}} \left[b_n - a_{n1}x_1 - a_{n2}x_2 - \dots - a_{n,n-1}x_{n-1} \right] \end{aligned} \right\} (3.16)$$

y se establecen las siguientes ecuaciones iterativas:

$$\left. \begin{aligned} x_1^{(k+1)} &= \frac{1}{a_{11}} \left[b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \dots - a_{1n}x_n^{(k)} \right] \\ x_2^{(k+1)} &= \frac{1}{a_{22}} \left[b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)} \right] \\ &\vdots \\ x_n^{(k+1)} &= \frac{1}{a_{nn}} \left[b_n - a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} - \dots - a_{n,n-1}x_{n-1}^{(k+1)} \right] \end{aligned} \right\} (3.17)$$

donde $x_i^{(k+1)}$ indica el valor de la "i-ésima" incógnita en la iteración "k + 1"

Para arrancar el método se establece una solución inicial \underline{x}_0 :

$$\underline{x}_0 = \begin{bmatrix} x_1^0 \\ x_2^0 \\ \vdots \\ x_n^0 \end{bmatrix} \quad (3.18)$$

dichos valores se sustituyen en el lado derecho de la ecuación (3.17) para obtener la siguiente solución aproximada:

$$\underline{x}_1 = \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ \vdots \\ x_n^{(1)} \end{bmatrix} \quad (3.19)$$

y así sucesivamente hasta que

$$\left| \underline{x}_{n+1} - \underline{x}_n \right| < \underline{\epsilon} \quad (3.20)$$

Para poder emplear este método es necesario verificar con anterioridad que el sistema sea compatible y determinado; además de que cumpla con las condiciones de convergencia del método. - Afortunadamente la mayoría de los problemas de tipo ingenieril cumplen los requisitos mencionados.

Ciertos sistemas que a primera vista no cumplen los requisitos del método pueden llenar los requisitos mediante un simple intercambio en la posición de las ecuaciones.

3.3.3 Descripción del programa

a) Subrutinas requeridas:

Ninguna.

b) Descripción de las variables.

A(I,J)	matriz de coeficientes del sistema
B(I)	vector de términos independientes
N	orden del sistema de ecuaciones
X(I)	valor inicial de las incógnitas del sistema y variable de localización temporal
Y(I)	valor de las incógnitas en la iteración "n"
XN(I)	valor de las incógnitas en la iteración "n + 1"

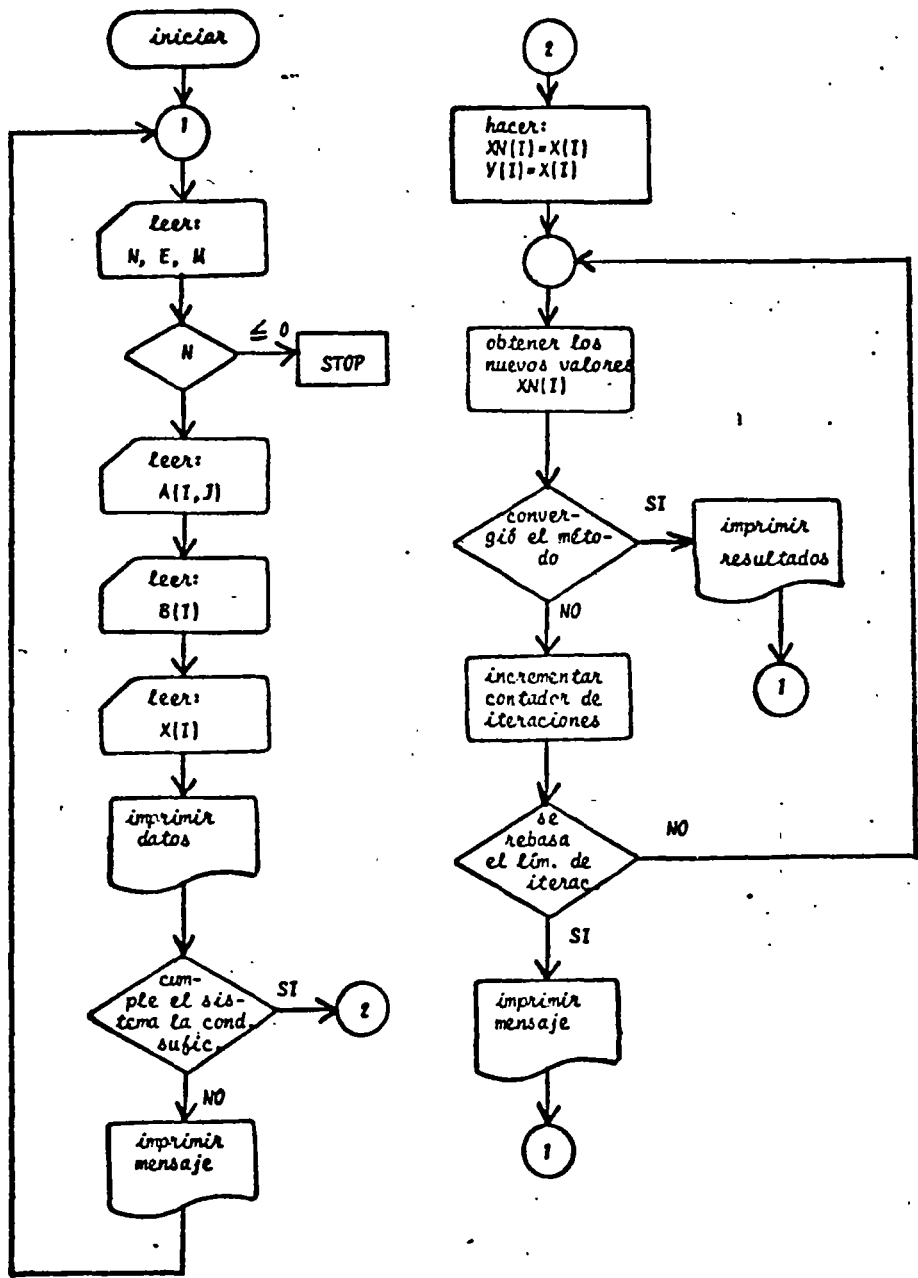


Fig. 3.6 Diagrama de bloques para el programa

6) Listado:

```

C   PROGRAMA PARA RESOLVER SISTEMAS DE ECUACIONES POR EL METODO DE
C   GAUSS-SEIDEL
C   SIGNIFICACC DE LAS VARIABLES EMPLEADAS
C   A=MATRIZ DE COEFICIENTES DEL SISTEMA DE ECUACIONES
C   B=VECTOR DE TERMINOS INDEPENDIENTES
C   X=VALOR INICIAL DE LA SOLUCION DEL SISTEMA
C   XN=SOLUCION DEL SISTEMA DE ECUACIONES EN LA SIGUIENTE ITERACION
C   N=ORDEN DEL SISTEMA
C   Y=VALOR DE LA SOLUCION DEL SISTEMA DE ECUACIONES EN LA ITERACION
C   ANTERIOR
C   M=MAXIMO NUMERO DE ITERACIONES
C   E=CRITERIO DE CONVERGENCIA

      DIMENSION A(20,20),B(20),X(20),Y(20),XN(20)
C   LECTURA DE DATOS
1   READ(5,200) M,M,E
      IF(N) 2,2,3
2   CALL EXIT
3   DO 4 I=1,M
4   READ(5,300) (A(I,J),J=1,M)
      READ(5,300) (B(I),I=1,M)
      READ(5,300) (X(I),I=1,M)
C   IMPRESION DE DATOS
      WRITE(6,400)
      DO 5 I=1,M
5   WRITE(6,500) (A(I,J),J=1,M),B(I)
      WRITE(6,600) (X(I),I=1,M)
C   SE INDAGA SI EL SISTEMA CUMPLE LA CONDICION SUFICIENTE DE CONVER-
C   GENCIA
      DO 7 I=1,M
      DO 6 J=1,M
      IF(A95(A(I,I)) - ABS(A(I,J))) 8,6,6
6   CONTINUE
7   CONTINUE
      GO TO 9
8   WRITE(6,700) I,J,I
      GO TO 1
C   OBTENCION DEL VALOR DE LAS INCOGNITAS
9   NCCN=1
      DO 10 I=1,M
      XN(I)=X(I)
10  Y(I)=X(I)
11  DO 14 K=1,M
      SUM=0.
      DO 13 I=1,M
      IF(K=I) 12,13,12
12  SUM=SUM + A(K,I)*XN(I)
13  CONTINUE
      XN(K)=(B(K)-SUM)/A(K,K)
14  CONTINUE
      DO 15 I=1,M
C   SE VERIFICA SI YA CONVERGIO EL METODO
      IF(ABS(XN(I)-Y(I))*E) 15,16,16
15  CONTINUE
C   IMPRESION DE RESULTADOS
      WRITE(6,800) (XN(I),I=1,M)
      WRITE(6,900) NCCN
      GO TO 1
16  NCCN=NCCN + 1
      IF(NCCN=M) 18,17,17
17  WRITE(6,900) (XN(I),I=1,M)
      WRITE(6,950) NCCN
      GO TO 1
18  DO 19 I=1,M
19  Y(I)=XN(I)
      GO TO 11
C   FORMATOS DE LECTURA E IMPRESION
200 FORMAT (2I5,F10.0)
300 FORMAT (10F3.0)
400 FORMAT(1H1,5(/),15X,'MATRIZ AMPLIADA')
500 FORMAT (/15X,10(F5.3,5X))
600 FORMAT(/,15X,'PRIMERA APROXIMACION DE LA SOLUCION',///,10X,10CF
16.2,5X))
700 FORMAT(4(/),15X,'EL METODO PUEDE NO CONVERGER DADO QUE',///,15X,'A(
2',12,'',12,'') ES MAYOR QUE A(1,12)',12,'')
800 FORMAT(4(/),15X,'LA SOLUCION DEL SISTEMA ES',///,5X,9(E12.5,2X))
900 FORMAT(4(/),15X,'NO SE LLEGA A LA SOLUCION',///,15X,'LA ULTIMA APR
OXIMACION ENCONTRADA FUE',///,5X,9(E12.5,2X))
950 FORMAT(4(/),15X,'ITERACIONES REALIZADAS= ',I4)
      END

```

Fig. 3.7 Listado del programa

3.3.4 Ejemplo

Para el circuito eléctrico de la fig. 3.8 se sabe que ---
 $I_1 = 1A$ e $I_2 = 2A$, $R_1 = R_2 = R_3 = R_4 = R_5 = R_6 = 1 \Omega$.

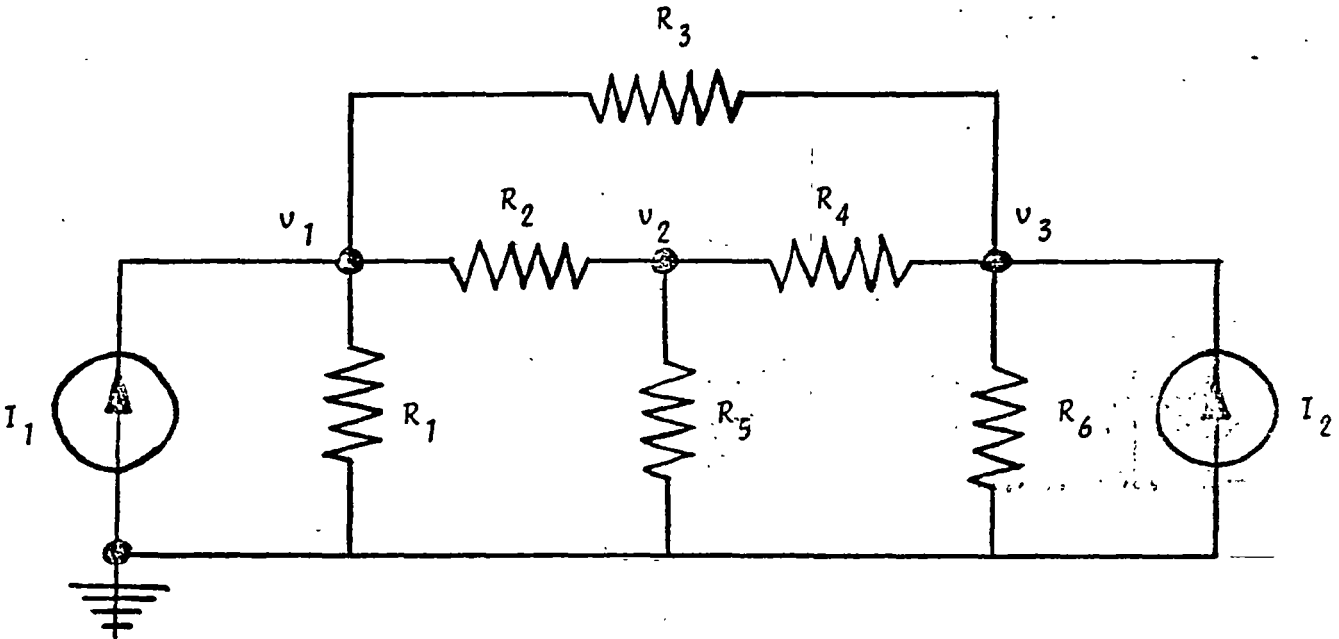


Fig. 3.8 Circuito eléctrico del problema del ejemplo 3.3.4

Se desea obtener el voltaje de los nodos V_1 , V_2 y V_3 .
Aplicando análisis nodal al circuito se obtiene:

$$\begin{aligned}
 3V_1 - V_2 - V_3 &= 1 \\
 -V_1 + 3V_2 - V_3 &= 0 \\
 -V_1 - V_2 + 3V_3 &= 2
 \end{aligned}$$

arreglo que es un sistema de ecuaciones lineales con todas las características propias para aplicar el método de Gauss-Seidel.

Se seleccionará como solución inicial al siguiente vector:

$$\begin{bmatrix} V_1^0 \\ V_2^0 \\ V_3^0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

* SOLUCION

TABLA 3.3 Datos del problema del ejemplo 3.3.4

$$N = 3$$

$$M = 50$$

$$EPS = 0.0001$$

$$A = \begin{bmatrix} 3 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}$$

$$\underline{X} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

TABLA 3.4 Resultados del problema del ejemplo 3.3.4

MATRIZ AMPLIADA

3.000	-1.000	-1.000	1.000
-1.000	3.000	-1.000	0.000
-1.000	-1.000	3.000	2.000

PRIMERA APROXIMACION DE LA SOLUCION

0.50 0.50 0.50

LA SOLUCION DEL SISTEMA ES

.10000E+01 .75000E+00 .12500E+01

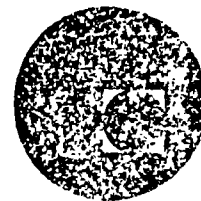
ITERACIONES REALIZADAS= 14

3.4 Bibliografía

1. CARNAHAN B., LUTHER H., WILKES J., "Applied Numerical Methods". New York: John Wiley and Sons Inc., 1969. pp. 269-307.
2. GEREZ G. Víctor, MURRAY LASSO M.A., "Teoría de Sistemas y Circuitos". México: Representaciones y Servicios de Ingeniería, S.A., 1972. pp. 99-123.
3. HADLEY G. "Algebra Lineal". Bogotá: Fondo Educativo - Interamericano, 1969. pp. 162-187.
4. HAMMING Richard, "Numerical Methods for Scientists -- and Engineers". New York: Mc Graw Hill Book Co., 1962. pp. 360-365.
5. JAMES M., SMITH G., WOLFORD J., "Applied Numerical -- Methods for Digital Computation with FORTRAN". ---- Scranton Penn: International Text Book Co., 1967. pp. 184-230.
6. JOHN STON J., BAILEY PRICE G., VAN VLECK F., "Linear - Equations and Matrices", Reading Mass.: Addison---- Wesley Co., 1966. pp. 1-94.
7. KUO S. Shan, "Computer Applications of Numerical ---- Methods". Reading Mass.: Addison Wesley Co., 1972. pp. 179-212.
8. OLIVERA S. Antonio, "Apuntes de Métodos Numéricos". - México.: Facultad de Ingeniería, UNAM. 1972. pp. 4.1-4.34.



centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



METODOS NUMERICOS Y APLICACIONES CON LA COMPUTADORA
DIGITAL

TEMA 3 : SISTEMAS DE ECUACIONES LINEALES.
(Continuación).

ABRIL, 1978.

- b. Find the highest common factor of these polynomials.
 c. Find the roots.
14. By direct substitution show that $y = ce^{\alpha t}$ is a solution of the differential equation

$$\frac{d^4 y}{dt^4} + 3 \frac{d^3 y}{dt^3} - 2 \frac{d^2 y}{dt^2} + 3 \frac{dy}{dt} + y = 0$$

if α is a root of the polynomial equation

$$\alpha^4 + 3\alpha^3 - 2\alpha^2 + 3\alpha + 1 = 0$$

If $\alpha_1, \alpha_2, \alpha_3,$ and α_4 are the four roots of this equation, show that

$$y = c_1 e^{\alpha_1 t} + c_2 e^{\alpha_2 t} + c_3 e^{\alpha_3 t} + c_4 e^{\alpha_4 t}$$

also satisfies the differential equation for any values of $c_1, c_2, c_3,$ and c_4 .

15. Show that for t sufficiently large and $\alpha_1, \alpha_2, \alpha_3,$ and α_4 all real, the value of y will be determined by the largest positive α_i .
16. Show that if $y = ce^{\alpha t}$, then y is less than or equal to c in absolute value for all $t > 0$, if the real part of α is zero or negative.
17. In a mechanical system of springs and masses, the motion of any part after a sudden impulse acceleration is governed by a differential equation of the form

$$a_1 \frac{d^n y}{dt^n} + a_2 \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_n \frac{dy}{dt} + a_{n+1} y = 0$$

The system will be stable, that is, will not tend to shake itself apart, if none of the solutions $y = ce^{\alpha t}$ grow very large as t increases. Show that the system will be stable if all the roots of the polynomial equation

$$a_1 \alpha^n + a_2 \alpha^{n-1} + \cdots + a_n \alpha + a_{n+1} = 0$$

have zero or negative real parts.

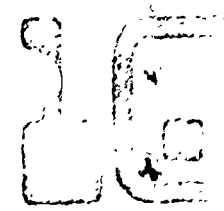
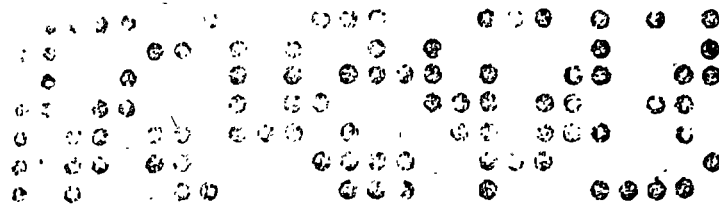
18. In an electrical circuit of resistors, capacitors, and inductances, the current at any point after a sudden initial impulse current is governed by a differential equation of the form

$$a_1 \frac{d^n i}{dt^n} + a_2 \frac{d^{n-1} i}{dt^{n-1}} + \cdots + a_n \frac{di}{dt} + a_{n+1} = 0$$

The system will be stable, that is, will not tend to develop very large local currents and burn out components if none of the solutions of the form $i = ce^{\alpha t}$ grow very large as t increases. Show that the system will be stable if all the roots of the polynomial equation

$$a_1 \alpha^n + a_2 \alpha^{n-1} + \cdots + a_n \alpha + a_{n+1} = 0.$$

have zero or negative real parts.



Simultaneous Linear Equations and Matrices

10.1 INTRODUCTION

In this chapter we turn to a problem of finding the values of unknowns, $x_1, x_2,$ etc., which satisfy systems of equations of type

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3n}x_n &= b_3 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n &= b_n \end{aligned} \quad (10-1)$$

When the number of equations is equal to the number of unknowns, there will ordinarily be a unique solution; that is, one set of values of x_1, x_2, \dots, x_n which satisfy all of the equations. At least such is the concept in the world of exact numbers and exact arithmetic. When the coefficients are approximate numbers, the concept of a solution becomes less clear, as the following example demonstrates.

Example 1. Find the solution of

$$\begin{aligned} 1.0x - 2.0y &= 1.0 \\ .5x - 4.0y &= 1.0 \end{aligned}$$

Figure 10-1 represents the solution, taking into account the approximate nature of the coefficients. Each equation is represented not by a line but by a band. Within our knowledge of the accuracy of the above numbers, any values in the band is as acceptable as any other. For example, in the first equation, when $x = 0$, y can be as small as $-1.05/1.95 \approx -.54$ or as large as $-.95/2.05 \approx -.46$. Thus at $x = 0$, the band for the first equation covers the region from $y = -.54$ to $y = -.46$. The two bands intersect not in a unique point but in a region, and any point in this region might be accepted as a solution. The nominal solution, for the above system of equations, obtained by accepting the coefficients as exact, is $x = 2/3$, $y = -1/6$, or approximately $x = .67$, $y = -.17$. However, the points $x = .86$, $y = -.12$

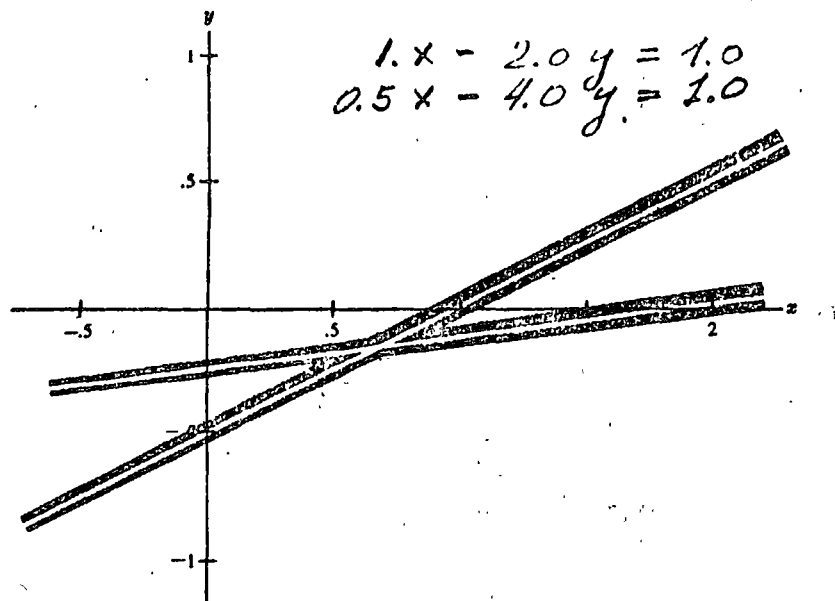


Figure 10-1

and $x = .5$, $y = -.21$ are also within the acceptable region. It is somewhat disconcerting to note that in this rather straightforward case, with the coefficients known to 10% or better, the solution is uncertain by 30% or more. It can be seen that if the equations represent lines that are nearly parallel, the region of overlap of the two bands representing the equations can be quite extended, as illustrated in Figure 10-2(a). In this case, even if the coefficients were exact, a small change in one of them can make a sizable difference in the solution, as illustrated in Figure 10-2(b) and (c). Equations having this property are termed ill-conditioned. An accurate solution can be found only by performing the computation with great care, since even small

errors may influence the answer greatly. Further, in practical problems, the answer itself must be viewed with some circumspection, since any inherent inaccuracies in the values of the coefficients may cause large changes in the answers.

The above example concerned itself with two equations and two unknowns, but analogous situations exist for higher numbers of equations and unknowns.

In this chapter, three general methods of solving a set of simultaneous linear equations are discussed: direct methods, in which the solution is found by a finite number of algebraic manipulations of the coefficients; iterative methods, which produce a set of successive approximations to the solution which hopefully become very close to the solution but never actually reach it; and matrix

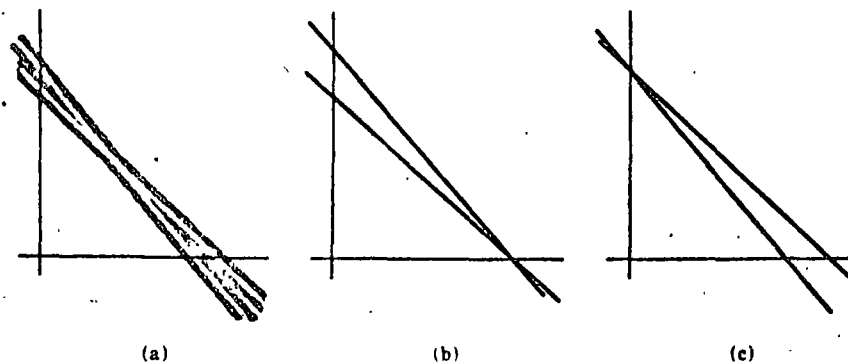


Figure 10-2

inversion methods, which are quite similar to the direct methods in numerical content but which provide conceptually more elegant bases for such methods. As was indicated in Chapter 5, no one of these methods is always best. The direct methods and matrix methods can have accuracy problems for some values of the coefficients and constant terms. The iterative methods can fail to converge to a solution. An attempt will be made to indicate the conditions under which the various methods can be expected to give satisfactory results.

10.2 THE ELIMINATION METHOD

The elimination method consists of multiplying various of the equations by appropriate constants and adding to other equations so as to obtain zero coefficients in some locations and eventually obtain equations that can be solved directly. The particular form of the elimination we shall use is that known as the Gauss-Jordan method. In this method, an appropriate multiple of the first equation is added to each of the other equations so that the resulting $n - 1$ equations have zero coefficients for the x_1 term. (If the first equation

does not have a term involving x_1 , we must first interchange two equations to obtain one with an x_1 term as the first equation.) Then an appropriate multiple of the next equation is added to all equations to eliminate the x_1 term from all but one equation. The process is continued until each equation contains only one unknown, and the equations are solved. At each step, the coefficient being used to eliminate other coefficients is called the pivotal coefficient.

To demonstrate how a machine program can be organized to perform this process, we shall construct some diagrams. Equation (10-1) will be represented internally in a computer only by the stored value of the coefficients a_{11} through a_{1n} and b_1 through b_n , perhaps as subscripted variables $A(I,J)$ and $B(J)$. Since the plus signs, x 's, and equals signs will not be stored in the computer anyway, let us omit them and write down only the constants and coefficients, arranged as in the equations but omitting the x 's and algebraic symbols, thus:

$$\begin{array}{ccccccc} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & b_3 \\ \vdots & & & & & \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & b_n \end{array} \quad (10-2)$$

remembering that we will mentally supply the x 's and symbols where needed.

To make the notation appear more uniform, let us rename b_1, b_2, \dots, b_n as $a_{1n+1}, a_{2n+1}, \dots, a_{nn+1}$. Then the array can be written

$$\begin{array}{cccccc} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & a_{1n+1} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & a_{2n+1} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & a_{3n+1} \\ \vdots & & & & & \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & a_{nn+1} \end{array} \quad (10-3)$$

As a first step in the elimination process we can divide the first equation by a_{11} to make the coefficient of x_1 become 1, and obtain the equations represented by

$$\begin{array}{cccccc} 1 & \frac{a_{12}}{a_{11}} & \frac{a_{13}}{a_{11}} & \cdots & \frac{a_{1n}}{a_{11}} & \frac{a_{1n+1}}{a_{11}} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & a_{2n+1} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & a_{3n+1} \\ \vdots & & & & & \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & a_{nn+1} \end{array}$$

Now we can eliminate the x_1 term from each of the other equations by multiplying the first equation by a_{21} and subtracting from the second, a_{31} and subtracting from the third, etc., giving

$$\begin{array}{cccccc} 1 & \frac{a_{12}}{a_{11}} & \frac{a_{13}}{a_{11}} & \cdots & \frac{a_{1n}}{a_{11}} & \frac{a_{1n+1}}{a_{11}} \\ 0 & a_{22} - a_{21} \left(\frac{a_{12}}{a_{11}} \right) & a_{23} - a_{21} \left(\frac{a_{13}}{a_{11}} \right) & \cdots & a_{2n} - a_{21} \left(\frac{a_{1n}}{a_{11}} \right) & a_{2n+1} - a_{21} \left(\frac{a_{1n+1}}{a_{11}} \right) \\ 0 & a_{32} - a_{31} \left(\frac{a_{12}}{a_{11}} \right) & a_{33} - a_{31} \left(\frac{a_{13}}{a_{11}} \right) & \cdots & a_{3n} - a_{31} \left(\frac{a_{1n}}{a_{11}} \right) & a_{3n+1} - a_{31} \left(\frac{a_{1n+1}}{a_{11}} \right) \\ \vdots & & & & & \\ 0 & a_{n2} - a_{n1} \left(\frac{a_{12}}{a_{11}} \right) & a_{n3} - a_{n1} \left(\frac{a_{13}}{a_{11}} \right) & \cdots & a_{nn} - a_{n1} \left(\frac{a_{1n}}{a_{11}} \right) & a_{nn+1} - a_{n1} \left(\frac{a_{1n+1}}{a_{11}} \right) \end{array}$$

At this point, we have eliminated the x_1 term from all but the first equation, using a_{11} as the pivotal coefficient. Note that in the computer, the new coefficients may as well be stored in the locations which held the old ones; that is, a_{12}/a_{11} simply replaces a_{12} , etc. If this is done, the above array becomes

$$\begin{array}{cccccc} 1 & a_{12} & a_{13} & \cdots & a_{1n} & a_{1n+1} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} & a_{2n+1} \\ 0 & a_{32} & a_{33} & \cdots & a_{3n} & a_{3n+1} \\ \vdots & & & & & \\ 0 & a_{n2} & a_{n3} & \cdots & a_{nn} & a_{nn+1} \end{array}$$

and the process which gives this array from the original one can be described by

$$\begin{array}{ll} a_{1j}/a_{11} \rightarrow a_{1j} & \text{for } j = 2, \dots, n+1 \\ a_{ij} - a_{11}a_{1j} \rightarrow a_{ij} & \text{for } i = 2, \dots, n \\ & j = 2, \dots, n+1 \end{array}$$

Note that these steps will not actually put $a_{11} = 1$ and $a_{i1} = 0$ for $i > 1$, that is, will not set the first column to one and zeros. Since we know they should be there, we can simply remember the fact, and not force the computer to take the extra steps to actually put them there.

Now we need to eliminate x_2 from equations 3 through n and from equation 1 by an analogous process. The steps are described by

$$\begin{array}{ll} a_{2j}/a_{22} \rightarrow a_{2j} & \text{for } j = 3, \dots, n+1 \\ a_{ij} - a_{12}a_{2j} \rightarrow a_{ij} & \text{for } i = 3, \dots, n, \text{ and } i = 1 \\ & j = 3, \dots, n+1 \end{array}$$

and produce an array of the form

$$\begin{matrix} 1 & 0 & a_{13} & \cdots & a_{1n} & a_{1n+1} \\ 0 & 1 & a_{23} & \cdots & a_{2n} & a_{2n+1} \\ 0 & 0 & a_{33} & \cdots & a_{3n} & a_{3n+1} \\ \vdots & & & & & \\ 0 & 0 & a_{n3} & \cdots & a_{nn} & a_{nn+1} \end{matrix}$$

If the process is continued, we eventually obtain the array

$$\begin{matrix} 1 & 0 & 0 & \cdots & 0 & a_{1n+1} \\ 0 & 1 & 0 & \cdots & 0 & a_{2n+1} \\ 0 & 0 & 1 & \cdots & 0 & a_{3n+1} \\ \vdots & & & & & \\ 0 & 0 & 0 & \cdots & 1 & a_{nn+1} \end{matrix} \quad (10-4)$$

and so $x_1 = a_{1n+1}$, $x_2 = a_{2n+1}$, etc. The process can be summarized in flow chart form as in Figure 10-3. A remote-terminal routine which would perform the process for systems up to 10 by 10 can be written as follows:

```

1  DIMENSION A(10,11)
2  1 PRINT, "NUMBER OF EQUATIONS"
3  INPUT, N
4  NN=N+1
5  PRINT, "A(1,1),A(1,2),,,A(1,N),B(1),A(2,1),ETC"
6  INPUT, ((A(I,J),J=1,NN),I=1,N)
7  DO 3 K=1,N
8  KK=K+1
9  DO 3 J=KK,NN
10 A(K,J)=A(K,J)/A(K,K)
11 DO 3 I=1,N
12 IF(K-I)2,3,2
13 2 A(I,J)=A(I,J)-A(I,K)*A(K,J)
14 3 CONTINUE
15 PRINT, "SOLUTION", (A(I,NN),I=1,N)
16 GO TO 1
17 END
    
```

~ the syst. is 2x10.

The inputs and responses would appear as follows:

```

RUN
N
? 3
A(1,1),A(1,2),,,A(1,N),B(1),A(2,1),ETC
? 2,3,5,5,3,4,7,6,1,3,2,5
SOLUTION -3.000000      2.000000      1.000000
    
```

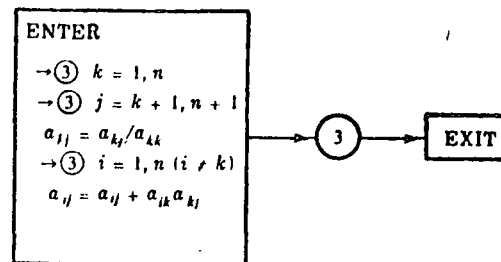


Figure 10-3: Flow chart for Gauss-Jordan method

The program given above will run into trouble if any of the coefficients $A(K,K)$ are zero, since it will attempt to divide by zero. One way to avoid this problem is to rearrange the equations any time a zero element on the diagonal is encountered.

Another way, not much more difficult to execute, is to rearrange the equations at each step so that the pivotal coefficient at each step is not only nonzero but is actually the largest coefficient. This approach not only avoids division by zero but also tends to enhance accuracy by minimizing round-off error. It has the disadvantage that the rearrangement will cause the unknowns to be scrambled at the end of the process. Suppose, for example, that initially the largest coefficient is a_{32} . Then we would like to arrange the equations as

$$\begin{aligned} a_{32}x_2 + a_{31}x_1 + a_{33}x_3 + \cdots + a_{3n}x_n &= b_3 \\ a_{22}x_2 + a_{21}x_1 + a_{23}x_3 + \cdots + a_{2n}x_n &= b_2 \\ a_{12}x_2 + a_{11}x_1 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\ a_{42}x_2 + a_{41}x_1 + a_{43}x_3 + \cdots + a_{4n}x_n &= b_4 \\ \vdots & \\ a_{n2}x_2 + a_{n1}x_1 + a_{n3}x_3 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

In terms of the original set of equations, (10-1), we have interchanged the first and third equations and have also interchanged the positions of x_1 and x_2 in all equations. In terms of the array of coefficients (10-3) we have inter-

Example 1. Show all inputs and machine responses for running the above program to solve the set of equations

$$\begin{aligned} 2x_1 + 3x_2 + 5x_3 &= 5 \\ 3x_1 + 4x_2 + 7x_3 &= 6 \\ x_1 + 3x_2 + 2x_3 &= 5 \end{aligned}$$

change the first and third rows and the first and second columns. If we continue the process to the end with no further rearrangement, the final value in a_{1n+1} when we reach the stage represented by (10-4) is not x_1 , but x_2 . Thus when we interchange rows or columns to obtain a large pivotal coefficient, we must also keep track of which unknown is represented by a particular column. This can be done by storing an identification number, ID, for each column which indicates the number of the unknown represented by that column. For example, in the rearrangement shown above, the information that the variable x_2 was now in the first column would be indicated by setting $ID(1) = 2$.

A separate subroutine can be written to handle the exchange of rows and columns to make the largest element appear at location $A(K,K)$. The subroutine given below would suffice for this purpose.

```

SUBROUTINE EXCH(A,N,NN,K,ID)
DIMENSION A(20,21),ID(20)
NROW=K
NCOL=K
B=ABS(A(K,K))
DO 2 I=1,N
DO 2 J=1,NN
IF(ABS(A(I,J)-B))2,2,21
21 NROW=I
NCOL=J
B=ABS(A(I,J))
2 CONTINUE
IF(NROW-K)3,3,31
31 DO 32 J=K,NN
C=A(NROW,J)
A(NROW,J)=A(K,J)
32 A(K,J)=C
3 CONTINUE
IF(NCOL-K)4,4,41
41 DO 42 I=1,N
C=A(I,NCOL)
A(I,NCOL)=A(I,K)
42 A(I,K)=C
I=ID(NCOL)
ID(NCOL)=ID(K)
ID(K)=I
4 CONTINUE
RETURN
END

```

In this subroutine, the statements up to number 2 locate the element having the largest absolute value and identify its location as NROW, NCOL. The statements from 2 to 3 interchange rows K and NROW if they are not the same row. The statements from 3 to 4 interchange columns K and NCOL, if they are not the same column, and also interchange the ID numbers to record this fact. Using this subroutine, one to solve the set of linear equations can be written as follows:

```

SUBROUTINE ELIM(AA,N,BB,X)
DIMENSION AA(20,20),BB(20),A(20,21),X(20),ID(20)
NN=N+1
DO 100 I=1,N
A(I,NN)=BB(I)
ID(I)=I
DO 100 J=1,N
100 A(I,J)=AA(I,J)
K=1
1 CALL EXCH(A,N,NN,K,ID)
2 IF(A(K,K))3,999,3
3 KK=K+1
DO 4 J=KK,NN
A(K,J)=A(K,J)/A(K,K)
DO 4 I=J,N
IF(K-I)41,4,41
41 A(I,J)=A(I,J)-A(I,K)*A(K,J)
4 CONTINUE
K=KK
IF(K-N)1,2,5
5 DO 10 I=1,N
DO 10 J=1,N
IF(ID(J)-I)10,6,10
6 X(I)=A(J,NN)
10 CONTINUE
RETURN
999 PRINT 1000
RETURN
1000 FORMAT(19H NO UNIQUE SOLUTION)
END

```

In this subroutine, the input coefficients are identified as $AA(I,J)$ and the input constants as $BB(I)$. The statements up to 100 reidentify these quantities as $A(I,J)$, so the original values will not be destroyed by the subroutine. Statement 1 calls subroutine EXCH to make the largest coefficient the pivotal

coefficient. If the largest coefficient is zero, the message "NO UNIQUE SOLUTION" is printed and an exit is taken. Otherwise, statements 3 through 4 solve the equations as in the remote-terminal program given earlier. Statements 5 through 10 use the identification numbers to unscramble the unknowns and return them in proper order.

10.3 GAUSS-SEIDEL METHOD

Another and quite different method of solving a system of linear equations is the so-called Gauss-Seidel method, in which equations (10-1) are rewritten in the following form:

$$\begin{aligned} a_{11}x_1 &= b_1 - a_{12}x_2 - a_{13}x_3 - \cdots - a_{1n}x_n \\ a_{22}x_2 &= b_2 - a_{21}x_1 - a_{23}x_3 - \cdots - a_{2n}x_n \\ a_{33}x_3 &= b_3 - a_{31}x_1 - a_{32}x_2 - a_{34}x_4 - \cdots - a_{3n}x_n \\ &\vdots \\ a_{nn}x_n &= b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots - a_{nn-1}x_{n-1} \end{aligned} \quad (10-5)$$

In words, in each of the equations all but one unknown is taken to the right-hand side of the equation. We then guess a set of values for x_2, x_3, \dots, x_n and substitute these in the right-hand side of the first equation and solve for x_1 . Then we substitute this value and the original values of x_3, \dots, x_n in the right-hand side of the second equation and solve for x_2 . We discard the old value of x_2 and keep this as a better one. We then substitute in the right-hand side of the third equation and obtain a new value for x_3 . After we have proceeded through all the equations in this fashion, we have a new set of values x_1, x_2, \dots, x_n . (We must first arrange the equations so that none of the $a_{ii} = 0$.) We then start again with the first equation and find a new x_1 , then a new x_2 , etc. Each time through this process gives us a new, and, we hope, better set of values for x_1, x_2, \dots, x_n . When the new values obtained agree with the previous set to within the accuracy we desire, we have the solution. This is an iteration process similar in nature to those discussed in Chapter 8. It is not absolutely certain that this process will converge, that is, that the differences between succeeding sets of values will get smaller and smaller. We shall discuss the convergence problem more fully a little later. It is not certain, either, how many multiplications will be required to obtain the solution to a desired accuracy. Each trip through the set of equations, or iteration, requires n^2 multiplications. If $(1/3)n$ iterations happen to be required, then the method will take about as long as the elimination method. It may take more or less time, depending entirely on the speed of convergence and accuracy required.

Example 1. Solve the system

$$\begin{aligned} x_1 - 2x_2 &= 1 \\ x_1 + 4x_2 &= 4 \end{aligned}$$

by the Gauss-Seidel method.

We write the equations as

$$x_1 = 1 + 2x_2 \quad (10-6)$$

$$x_2 = 1 - x_1/4 \quad (10-7)$$

Let us take as starting values $x_1 = x_2 = 0$.

Putting $x_2 = 0$ in equation (10-6), we obtain

$$x_1 = 1$$

Putting $x_1 = 1$ in equation (10-7), we obtain

$$x_2 = 3/4$$

At the end of the first iteration, then, we have

$$x_1 = 1, \quad x_2 = 3/4$$

Putting $x_2 = 3/4$ in equation (10-6), we have

$$x_1 = 5/2$$

Putting $x_1 = 5/2$ in equation (10-7), we have

$$x_2 = 3/8$$

At the end of the second iteration, then, we have

$$x_1 = 5/2, \quad x_2 = 3/8$$

We can continue this process. The results for the first several steps, starting from the beginning, are

x_1	x_2
0	0
1	.75
2.5	.375
1.75	.5625
2.125	.46875
1.9375	.515625
2.03125	.4921875
1.984375	.51390625

It is verified from the equation that the correct solution is $x_1 = 2$, $x_2 = 1/2$. This solution is slowly converging toward those values.

Example 2. Solve the system

$$\begin{aligned}x_1 + 4x_2 &= 4 \\ -x_1 - 2x_2 &= 1\end{aligned}$$

by the Gauss-Seidel method.

This is the same problem as Example 1, with the equations reversed. We write the equations as

$$\begin{aligned}x_1 &= 4 - 4x_2 \\ x_2 &= -1/2 + x_1/2\end{aligned}$$

Then the successive iterations give the following values:

x_1	x_2
0	0
4	1.5
-2	-1
8	3.5
-10	-5.5
26	12.5
-46	-23.5

It is clear that the process is diverging, and the solution will not be obtained.

Example 3. Apply the Gauss-Seidel method to Example 1, Section 10.2.

The equations are

$$\begin{aligned}2x_1 + 3x_2 + 5x_3 &= 5 \\ 3x_1 + 4x_2 + 7x_3 &= 6 \\ x_1 + 3x_2 + 2x_3 &= 5\end{aligned}$$

We write them as

$$\begin{aligned}2x_1 &= 5 - 3x_2 - 5x_3 \\ 4x_2 &= 6 - 3x_1 - 7x_3 \\ 2x_3 &= 5 - x_1 - 3x_2\end{aligned}$$

Successive iterations give (to four decimal places)

x_1	x_2	x_3
0	0	0
2.5	-.375	1.8125
-1.4688	.5703	2.3789
-4.3027	.5640	3.8054
-7.8595	.7352	5.3270
-11.9203	1.1180	6.7831

In Section 10.2 we found that the solution to this system was

$$x_1 = -3, \quad x_2 = 2, \quad x_3 = 1$$

Our iteration scheme is not converging toward those values.

10.31 Convergence of the Gauss-Seidel Method

Some insight into the convergence problem can be obtained by following Examples 1 and 2, Section 10.3, in graphical form. Figure 10-4 illustrates the scheme followed in Example 1. Starting at the point P_0 , we change x_1 (that is, move horizontally) to arrive on the line $x_1 - 2x_2 = 1$, and then change x_2 (that is, move vertically) to arrive on the line $x_1 + 4x_2 = 4$, bringing us to the point P_1 . This is the point given by the first iteration. On the second iteration we move horizontally, then vertically to arrive at P_2 . On the third we move horizontally, then vertically to arrive at P_3 , etc. It is clear from the figure that this process is bringing us closer and closer to the true point of intersection.

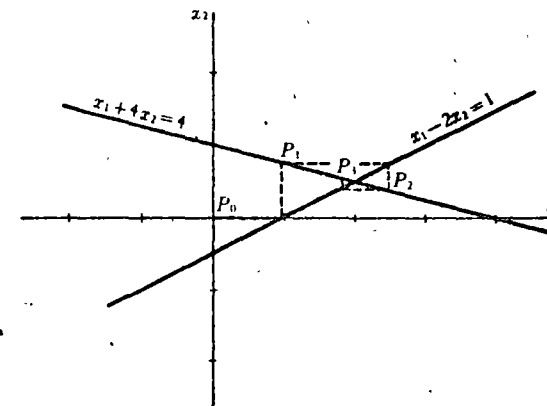


Figure 10-4

Figure 10-5 illustrates the scheme followed in Example 2. The same two lines are involved, but this time we always move horizontally to reach the line $x_1 + 4x_2 = 4$ and vertically to reach the line $x_1 - 2x_2 = 1$. The points P_0, P_1, P_2, \dots are the results of the successive iterations in this case.

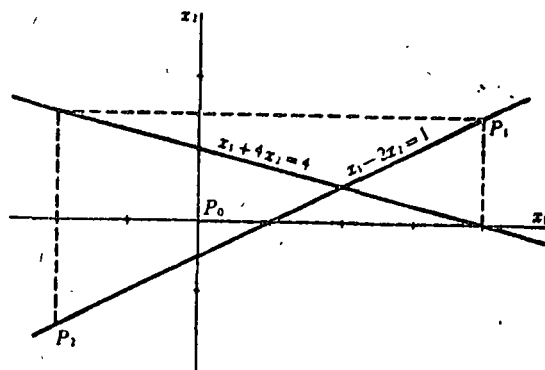


Figure 10-5

It appears that graphically the Gauss-Seidel method for two equations in two unknown consists of following the above boxlike pattern about the point of intersection of the two lines: If this pattern is followed in the correct direction the intersection will be approached, but if it is followed in the wrong direction the process will diverge from the intersection. This is the case if the slopes of the lines have opposite signs. If the signs of the slopes are the same, the situation is a little different, as depicted in Figure 10-6. The sequence of points P_0, P_1, P_2 is part of a convergent process, in which we proceed horizontally to line (b), then vertically to line (a). The points $P_0,$

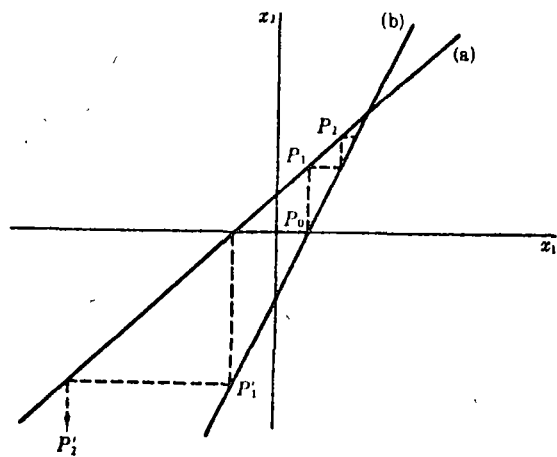


Figure 10-6

P_1, P_2, \dots are part of a divergent process, in which we proceed horizontally to line (a) then vertically to line (b).

As indicated by the above figures, the situation regarding convergence for the Gauss-Seidel method for two equations in two unknowns is as follows: The process will converge for the equations arranged in one order and diverge for the equations arranged in the opposite order. The only exception occurs when the equations represent perpendicular lines, in which case the process will not converge for either arrangement. It is interesting to note that, contrary to our experience with iteration methods in the preceding chapters, the convergence or nonconvergence for these linear equations does not depend on choice of initial estimate.

For larger systems of equations the situation becomes much more complex. The necessary and sufficient conditions for convergence are known but are not easily expressed in a very usable form. Sometimes a rearrangement of the equations will produce convergence, but this is not at all guaranteed. The likelihood of convergence is usually increased if the equations are rearranged so that the coefficients $a_{11}, a_{22}, a_{33}, \dots, a_{nn}$ which appear on the left-hand side in the system as written in Section 10.3 are the largest coefficients in absolute value. In fact, convergence is assured in this case if in each equation the absolute value of the coefficient a_{ii} is larger than the sum of the absolute values of the remaining coefficients. This condition is not often met. In fact, as in Example 3, Section 10.3, it is often impossible even to write all the equations with largest terms on the left-hand side.

10.32 Flow Chart and Program for the Gauss-Seidel Method

The flow chart in Figure 10-7 describes the Gauss-Seidel method. This flow chart uses the equations arranged just as they are, with no attempt to rearrange the equations to increase the likelihood of convergence. If desired, it could be preceded by another section of flow chart which would rearrange the equations in attempt to enhance the likelihood of convergence. In order to cut down on the number of divisions required, each of the equations is first divided through by the coefficient a_{ii} , so that in the set of new coefficients, c_{ij} , the c_{ii} 's are all one. This flow chart computes at each iteration a quantity

$$E = \sum_{i=1}^n |x_i^{\text{new}} - x_i^{\text{old}}|$$

and when this quantity becomes smaller than the given number d , the iteration stops. Note that, in the way the expression for P is written, P is precisely $x_i^{\text{new}} - x_i^{\text{old}}$.

The FORTRAN-subroutine below uses the Gauss-Seidel method to solve an N by N system of linear equations, following the flow chart. Again, if a rearrangement of the equations were desired, it could be accomplished by using a subroutine for that purpose just prior to using the one given below.

```

SUBROUTINE GAUSID(A,N,B,X,ERR)
DIMENSION A(20,20), B(20), C(20,21), X(20)
K=0
NN=N+1
DO 11 I=1,N
IF(A(I,I))12,6,12
12 X(I)=1.
C(I,NN)=B(I)/A(I,I)
DO 11 J=1,N
11 C(I,J)=A(I,J)/A(I,I)
1 CONTINUE
E=0.
DO 3 I=1,N
P=C(I,NN)
DO 2 J=1,N
P=P-C(I,J)*X(J)
2 CONTINUE
X(I)=X(I)+P
E=E+ABS(P)
3 CONTINUE
IF(E-ERR)4,4,5
4 RETURN
5 K=K+1
IF(100-K)6,1,1
6 PRINT 1000
RETURN
1000 FORMAT(25H GAUSID DOES NOT CONVERGE)
END

```

EXERCISE 26

- Following the remote-terminal program of Section 10.2, solve the following systems of equations,
 - $x - y = 2$
 $x - y = 4$
 - $x + 2y = 7$
 $4x + y = 5$
 - $2x + 3y + z = 2$
 $x + 2y - 4z = 3$
 $4x - 2y + z = -2$
 - $x + y + z = 4$
 $3x - y - z = 1$
 $x + 2y - z = 5$

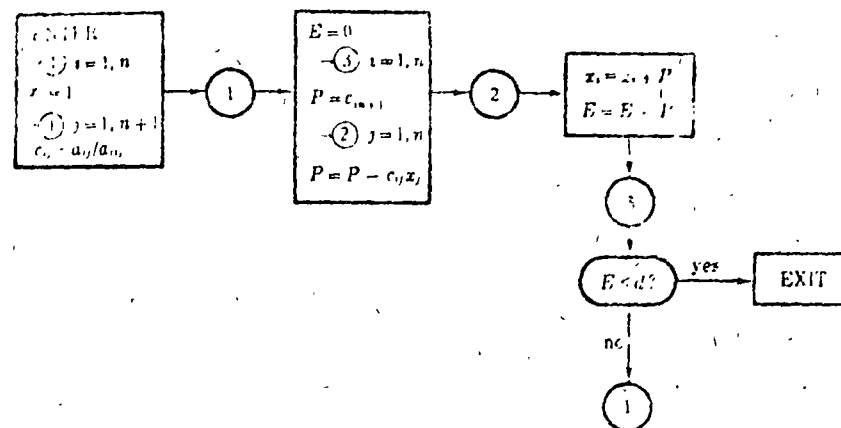


Figure 10-7: Solution of linear equations—Gauss-Seidel method

- Following the flow chart, Figure 10-7, perform the first four iterations for the following systems of equations.
 - $2x + y = 3$
 $x + 2y = 3$
 - $4x - y = 6$
 $x + 3y = 5$
 - $3x + 2y + z = 5$
 $2x + 5y + 4z = 8$
 $x + 4y + 6z = 4$
 - $x + 2y + 4z = 6$
 $3x + y + 2z = 5$
 $2x + 4y + z = 4$
- Write a FORTRAN program that will input a system of linear equations up to size 20 by 20, call subroutine ELIM of Section 10.2 to solve them, and print the result.
- Write a FORTRAN subroutine which will rearrange a set of linear equations, for use of subroutine GAUSID of Section 10.3, so that after rearrangement

$$a_{ki} \geq a_{k+1,i} \quad \text{for } k > 1$$
 - Show that your subroutine will correctly arrange the equations

$$x_1 + 8x_2 + x_3 = 10$$

$$x_1 + x_2 + 7x_3 = 9$$

$$9x_1 + x_2 + x_3 = 11$$
 so that the Gauss-Seidel method will converge.
 - Explain what may go wrong with this method of arrangement if some of the coefficients are zero.

10.4 MATRICES

In all the methods of solving linear equations by computer, we have seen that only the coefficients and the constants appear within the machine. The

formalism of writing down the unknowns x_1, x_2, \dots , when we write the equations longhand, merely serves to identify the proper locations of the coefficients and constants. In other words, the solution of the system of equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

is determined completely by the array of coefficients

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

and the array of constants

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

If we are given any two such arrays, we can write the set of equations they represent. If we were to change the numerical value of any number in one of these arrays, a different set of equations would be represented. Further, if we were to interchange the position of any two of the numbers, a still different set of equations would be represented. All this suggests that it may be useful to consider these arrays of numbers as separate entities, establish rules for manipulating them, and perhaps free ourselves somewhat of the repetitious writing of the basically nonessential symbols $x_1 +, x_2 +, x_3 +, \dots$. Considerations such as these have led to the definition of a matrix as an array of numbers, and to the development of an "algebra" of matrices, a set of rules for combining matrices to form other matrices. Once developed, matrix algebra has come to have far-reaching applications, completely apart from systems of linear equations.

10.5 DEFINITIONS AND ELEMENTARY OPERATIONS

A matrix is a rectangular array of quantities or numbers, such as

$$\begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array}$$

In order to distinguish a matrix from a determinant, which also frequently looks like an array of numbers, it is customary to enclose a matrix in brackets, or large parentheses, or double bars, as

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad \left(\begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array} \right), \quad \text{or} \quad \left\| \begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array} \right\|$$

A determinant is usually written between single bars, as

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

This determinant only looks like an array. Really the symbol only stands for a single quantity, which is obtained by multiplying and adding the individual a_{ij} 's in the manner described in Section 10.54. The matrix, on the other hand, has no single numerical value but is instead the entire array. We shall be using a single letter or symbol to stand for a matrix, such as

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

When we do this, it is important to remember that A is not a number, and so does not act like a number; that is, it does not obey the ordinary laws of algebra.

Occasionally, we will be interested in the value of a determinant made up of exactly the same elements as some square matrix A . When we do we shall refer to it as the determinant of the matrix A .

A matrix of m rows and n columns is an m by n matrix. If $m = n$, the matrix is a square matrix of order m .

The sum of the diagonal elements of a square matrix is called the "trace" of the matrix, $\text{tr } A = a_{11} + a_{22} + \cdots + a_{nn}$.

If a matrix consists of a single column it is called a column matrix, or sometimes a column vector.

If the elements in the main diagonal of a square matrix are ones, and all the other elements are zeros, the matrix is called a unit-matrix, or identity matrix. Thus

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

is a unit matrix of order 3. Unit matrices of any order are usually denoted by the symbol I .

If all the elements are zero, the matrix is called a zero matrix.

Two matrices A and B are said to be equal if:

- (1) They have the same number of rows.
- (2) They have the same number of columns.
- (3) Each pair of corresponding elements are equal.

10.51 Addition and Subtraction of Matrices

The operations of addition and subtraction are defined for two matrices A and B if:

- (1) They have the same number of rows.
- (2) They have the same number of columns.

The sum of two matrices is the matrix obtained by adding corresponding pairs of elements. Thus, if

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}$$

then

$$A + B = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & a_{13} + b_{13} \\ a_{21} + b_{21} & a_{22} + b_{22} & a_{23} + b_{23} \\ a_{31} + b_{31} & a_{32} + b_{32} & a_{33} + b_{33} \end{pmatrix}$$

The difference $A - B$ is the matrix obtained by subtracting the elements of B from the corresponding elements of A .

$$A - B = \begin{pmatrix} a_{11} - b_{11} & a_{12} - b_{12} & a_{13} - b_{13} \\ a_{21} - b_{21} & a_{22} - b_{22} & a_{23} - b_{23} \\ a_{31} - b_{31} & a_{32} - b_{32} & a_{33} - b_{33} \end{pmatrix}$$

Example 1. Find $A + B$ and $A - B$, where

$$A = \begin{pmatrix} 3 & 0 & -2 \\ 1 & 3 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 2 & 1 & 2 \\ -1 & 3 & -2 \end{pmatrix}$$

SOLUTION:

$$A + B = \begin{pmatrix} 5 & 1 & 0 \\ 0 & 6 & -1 \end{pmatrix}, \quad A - B = \begin{pmatrix} 1 & -1 & -4 \\ 2 & 0 & 3 \end{pmatrix}$$

Example 2. Find $A + B$ and $A - B$, where

$$A = \begin{pmatrix} 3 & 0 & -2 \\ 1 & 3 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 2 & -1 \\ 1 & 3 \\ 2 & -2 \end{pmatrix}$$

Since there are not the same number of rows or columns in A and B , they cannot be added or subtracted. The symbols $A + B$ and $A - B$ are meaningless in this case.

Example 3. Given two matrices A and B , each with N columns and M rows, write FORTRAN statements which would form the sum, $C = A + B$.

The matrix A can be represented by a single subscripted variable $A(I,J)$, where I runs from 1 to M and J runs from 1 to N . The same is true for B and C . Then the required FORTRAN statements are

```
DO 20 I=1,M,
DO 20 J=1,N
20 C(I,J)=A(I,J)+B(I,J)
```

A total of $N \times M$ additions are required to obtain C .

As a direct extension of addition, it would be natural to be able to say

$$A + A = 2A$$

This leads to the definition of multiplication of a matrix by a constant as follows: A constant times a matrix is the matrix obtained by multiplying all elements of the original matrix by the constant.

10.52 Multiplication of Matrices

At first acquaintance, the operation of multiplication of two matrices seems to be defined in a most peculiar way. There are very good reasons for choosing to call this seemingly awkward process "multiplication," and these will appear shortly.

The product AB of two matrices, A and B , is defined only if the number of columns in A is equal to the number of rows in B . In all other cases the product is undefined. If the number of columns in A is equal to the number of rows in B , then A and B are said to be "conformable" in the order AB .

The product AB of two conformable matrices is itself a matrix, whose elements are found according to the following rule: The element in the i th row and the j th column of the product is the sum of the products by pairs of the elements of the i th row of A and j th column of B .

Example 1. If

$$A = \begin{pmatrix} 1 & 2 \\ 3 & -1 \end{pmatrix}, \quad B = \begin{pmatrix} 3 & -2 \\ 2 & 1 \end{pmatrix}$$

find AB .

Since A has 2 columns and B has 2 rows, A and B are conformable in the order AB , so the product is indeed defined. To find the element in the first row, first column of the product matrix, we take the first row of A , which is

$$1 \quad 2$$

and the first column of B , which is

$$\begin{matrix} 3 \\ 2 \end{matrix}$$

and form the sum of the products by pairs:

$$1 \times 3 + 2 \times 2 = 7$$

Hence 7 is the element in the first row, first column of the product.

In like manner, the element in the first row and second column of the product is obtained from combining the first row of A with the second column of B , thus:

$$1 \times (-2) + 2 \times 1 = 0$$

and for the second row, first column,

$$3 \times 3 + (-1) \times 2 = 7$$

and the second row, second column,

$$3 \times (-2) + (-1) \times 1 = -7$$

Hence the product is

$$\begin{pmatrix} 1 & 2 \\ 3 & -1 \end{pmatrix} \begin{pmatrix} 3 & -2 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 7 & 0 \\ 7 & -7 \end{pmatrix}$$

Example 2. If

$$A = \begin{pmatrix} 1 & 3 & 1 \\ -2 & 1 & -1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

find AB .

Since A has 3 columns and B has 3 rows, they are conformable in the order AB . We can expedite the process of finding the product by writing the two matrices side by side, and then going across a row of A and down a column of B forming products by pairs, thus:

$$\begin{pmatrix} 1 & 3 & 1 \\ -2 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \times 1 + 3 \times 2 + 1 \times 3 \\ -2 \times 1 + 1 \times 2 - 1 \times 3 \end{pmatrix} = \begin{pmatrix} 10 \\ -3 \end{pmatrix}$$

Example 3. For the matrices A and B of Example 2, find BA .

Since B has 1 column and A has 2 rows, they are not conformable in the order BA . The product BA is not defined!

Example 4. If

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}, \quad B = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1l} \\ b_{21} & b_{22} & \dots & b_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nl} \end{pmatrix}$$

and

$$AB = C$$

write a formula for finding c_{ij} , the element in the i th row and j th column of C .

The i th row of A is

$$a_{i1} \quad a_{i2} \quad \dots \quad a_{in}$$

and the j th column of B is

$$\begin{matrix} b_{1j} \\ b_{2j} \\ \vdots \\ b_{nj} \end{matrix}$$

and the sum of the products by pairs gives

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj}$$

or, in more abbreviated form,

$$c_{ij} = \sum_{k=1}^n a_{ik}b_{kj}$$

Example 5. Given matrix A with M rows and N columns and matrix B with N rows and L columns, write a set of FORTRAN statements which will form the product $C = AB$.

A suitable set of statements is

```
DO 10 I=1,M
DO 10 J=1,L
C(I,J)=0.
DO 10 K=1,N
10 C(I,J)=C(I,J)+A(I,K)*B(K,J)
```

We note that statement 10 is in three DO loops, and will be performed $N \times M \times L$ times, or $N \times M \times L$ multiplications are required to find the product matrix C .

Example 6. If

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

write the product Ax .

SOLUTION:

$$Ax = \begin{pmatrix} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \end{pmatrix}$$

Note that this product Ax is actually a column vector, having three elements.

Example 7. Write the system of linear equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned}$$

in matrix form.

From Example 6, if we define

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

then the left-hand sides of the equations above are just the three elements

of the column vector Ax . Now let us define the column vector

$$b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

We recall that two matrices are equal if and only if every pair of corresponding elements are equal. Thus, the statement

$$Ax = b$$

is a matrix equation. The expressions on each side of the equals sign are matrices. The equation means that

- (1) The first element of Ax , that is, $a_{11}x_1 + a_{12}x_2 + a_{13}x_3$, is equal to b_1 .
- (2) The second element of Ax , that is, $a_{21}x_1 + a_{22}x_2 + a_{23}x_3$, is equal to b_2 .
- (3) The third element of Ax , that is, $a_{31}x_1 + a_{32}x_2 + a_{33}x_3$, is equal to b_3 .

Hence the matrix equation

$$Ax = b$$

says exactly the same thing as the system of linear equations above.

We see from Examples 6 and 7 that any system of linear equations, with any number of unknowns, can be represented by a matrix equation

$$Ax = b$$

where A is a matrix and x and b are column vectors of the correct order. This simple expression is one of the several happy results of the seemingly odd definition of multiplication.

10.53 Laws of Matrix Algebra

We have defined three operations with matrices and have given them the names "addition," "subtraction," and "multiplication"—names we use in the ordinary algebra of numbers. Actually this is a little dangerous, since it suggests that these new matrix operations will obey the same rules as the ordinary arithmetic operations, and we really have no right to expect that they will do so.

The fundamental laws of ordinary algebra are the following:

- (1). Addition is *commutative*. $a + b = b + a$; that is, if we add b to a , or a to b , we will get the same result.

(2). Addition is *associative*. $(a + b) + c = a + (b + c)$; that is, if we add $a + b$, and then add c to this sum, we get the same result as if we add b and c first, and then add a to the sum.

(3). Multiplication is *distributive* with respect to addition. $a(b + c) = ab + ac$; that is, if we add b to c and then multiply by a , we get the same result as if we multiply a by b , multiply a by c , and then add the result.

(4). Multiplication is *commutative*. $ab = ba$; that is, if we multiply a by b or b by a , we get the same result.

(5). Multiplication is *associative*. $(ab)c = a(bc)$; that is, if we take the product ab and multiply by c we get the same answer as if we take the product bc and multiply by a .

When these laws for the algebra of numbers are investigated for matrices, it is found that they all hold *except* law 4, the commutative law for multiplication. As was seen in Examples 2 and 3, Section 10.52, it is possible to have two matrices whose product AB could be found but whose product BA was not even defined.

In summary, then, we can say that, in expressions involving sums, differences, and products of matrices, we can use the same laws for combining these operations as for ordinary numbers except that the order of any two matrices in a product cannot be reversed. In a matrix equation, we may add the same matrix to both sides or subtract the same matrix from both sides without changing the equality. We also may *multiply* both sides by the same matrix, provided that:

(1) The matrix is conformable with those by which it is to be multiplied.

(2) The order of multiplication is made the same on both sides of the equation.

Example 1. If A , B , and C are square matrices of order n , and if

$$A + B = C$$

solve for A .

Subtracting B from both sides, we have

$$A = C - B$$

Example 2. If A , B , C , and D are square matrices of order n , and if $A = B + C$, find AD and DA .

Multiplying the equation

$$A = B + C$$

on the right by D , we have

$$AD = (B + C)D = BD + CD$$

Multiplying the above equation on the left by D , we have

$$DA = D(B + C) = DB + DC$$

10.54 Determinants

The determinant of a square matrix A is defined to be the number obtained in the following manner: From the elements of A , we form all possible products containing exactly one element from each row and column in A . To each such term we assign a plus or minus sign in accordance with a rule to be stated shortly. The sum of these terms is the value of the determinant. The sign to be assigned to a term is determined by the following procedure. The factors in the term are arranged in order according to the row from which each factor was chosen:

$$a_{1k_1} a_{2k_2} a_{3k_3} \cdots a_{nk_n}$$

We then rearrange these factors so that they are in order according to the column from which each was chosen, that is, so that the subscripts k_1, k_2, \dots, k_n are in their natural order, and count the number of interchanges required to do this. We assign the term a plus sign if the number of interchanges was even and a minus sign if it was odd. For a 2 by 2 determinant, then,

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12}$$

For a 3 by 3 system,

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31}$$

It is clear that, by utilizing the programming methods of the earlier chapters, we can cause a computer to perform such calculations and provide the solution to a system of equations. It is not so obvious, but it can be shown that such a procedure is quite inefficient in machine time, particularly for systems involving a very large number of unknowns. According to the rule just stated for evaluating a determinant, an n by n determinant is the sum of $n!$ terms, each of which is the product of n numbers. If we were to calculate the value of a determinant by the most direct method, then, about $n \times n!$ multiplications would be required. For even a 10 by 10 determinant, several

million multiplications would be required, and for a 20 by 20 determinant, over 10^{18} multiplications would be needed. This would require over 100,000 years even on the fastest computers.

There is another method of evaluation of a determinant that is very much faster than the brute-force approach. If all elements on one row of a determinant are changed by adding or subtracting a constant multiple of the corresponding elements of another row, the value of the determinant is unchanged. By repeated application of this rule, we can reduce a determinant to a "triangular" form, in which all elements below the main diagonal are zero. For example,

$$\begin{vmatrix} b_{11} & b_{12} & b_{13} & \cdots & b_{1n} \\ 0 & b_{22} & b_{23} & \cdots & b_{2n} \\ 0 & 0 & b_{33} & \cdots & b_{3n} \\ 0 & 0 & 0 & b_{44} & \cdots & b_{4n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \cdots & b_{nn} \end{vmatrix}$$

The value of a determinant when written in this form turns out to be just the product of the diagonal elements, $b_{11}b_{22}b_{33} \cdots b_{nn}$, since all other terms formed in accordance with the definition of a determinant's value contain at least one factor whose value is zero. Hence, after a determinant is written in triangular form, only $n - 1$ multiplications are required to find its value.

The process is quite similar to that used in Section 10.2 to solve a system of linear equations by the elimination method. We start out with the array

$$\begin{matrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{matrix}$$

and perform the operations

$$a_{ij} - \frac{a_{ik} a_{kj}}{a_{kk}} \rightarrow a_{ij} \quad \text{for } i \text{ and } j = k + 1, k + 2, \dots, n$$

$$k = 1, \dots, n$$

Figure 10-8 is a flow chart of the process.

A calculation based on the flow chart, Figure 10-8, could run into trouble if a_{kk} ever becomes zero, since there is a division by this quantity. This problem can be avoided by taking the additional precaution of checking to see if a_{kk} is zero and if so interchanging two rows to obtain a nonzero value for a_{kk} . Since interchanging two rows in a determinant changes the sign of

the determinant's value, we must also change the signs of the elements in one of the rows to correct this.

There can also be accuracy problems associated with evaluating a determinant using the above flow chart, particularly for determinants of large order. These problems tend to be alleviated if the rows and columns are rearranged at each step so that a_{kk} is not only nonzero but is actually the largest element in absolute value. SUBROUTINE EXCH of Section 10.2

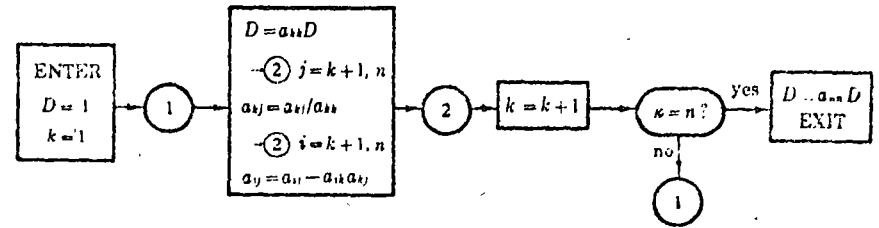


Figure 10-8: Evaluation of a determinant

provided this service for the elimination method, and with minor modifications it can be made to work in the present case. The main difference is that interchanging rows or columns in a determinant changes the sign of the determinant. We can correct for this by changing statement 32 to read

$$32 \ A(K,J) = -C$$

and statement 42 to read

$$42 \ A(I,K) = -C$$

We also need the dimension statement to read A(20,20) instead of A(20,21). We do not need the quantity ID as output, so we can eliminate the three statements following statement 42 and change the first statement to read

SUBROUTINE EXCH2(A,N,NN,K)

We changed the name as well, to ensure that the old routine of Section 10.2 is not used by mistake.

Another step which is useful to avoid undetected accuracy loss is in connection with the computation

$$a_{ij} = a_{ij} - a_{ik}a_{kj}$$

If the result of this subtraction is supposed to be zero, then this subtraction will be subject to the trouble mentioned many times earlier in the text, less

of accuracy caused by introduction of leading zeros. The method of protection against this trouble is the same one used in division of polynomials in Section 9.43. We check the result of the subtraction, and if the difference is much smaller than the numbers being subtracted, we set the difference equal to zero. The operation can be described by a section of flow chart (as in Figure 10-9) in which a_{ij} is set equal to zero if more than four significant figures have been lost in the subtraction.

The FORTRAN subroutine below evaluates the N th-order determinant

$$\begin{vmatrix} A(1,1) & A(1,2) & \cdots & A(1,N) \\ A(2,1) & A(2,2) & \cdots & A(2,N) \\ \vdots & \vdots & \ddots & \vdots \\ A(N,1) & A(N,2) & \cdots & A(N,N) \end{vmatrix}$$

for values of N to 20. In the first statement, the determinant is given the name AA, and the statements up to 100 redefine the elements so that the original determinant will not be destroyed during the calculation. State-

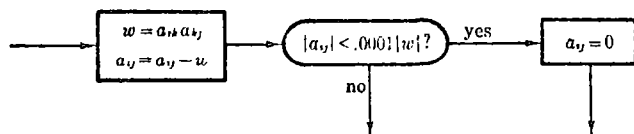


Figure 10-9 .

ment 1 calls EXCH2 to interchange rows and columns if necessary to move the largest element to location A(K,K). Statements 3 through 4 perform the actual calculation required in the main part of the flow chart, Figure 10-8.

```

SUBROUTINE DETERM(AA,N,D)
DIMENSION AA(20,20),A(20,20)
DO 100 I=1,N
DO 100 J=1,N
100 A(I,J)=AA(I,J)
D=1.
K=1
1 CALL EXCH2(A,N,N,K)
D=A(K,K)*D
IF(A(K,K))3,10,3
3 KK=K+1
DO 4 J=KK,N
A(K,J)=A(K,J)/A(K,K)
  
```

```

DO 4 I=KK,N
W=A(I,K)*A(K,J)
A(I,J)=A(I,J)-W
IF(ABS(A(I,J))- .0001*ABS(W))42,4,4
42 A(I,J)=0.
4 CONTINUE
K=KK
IF(K-N)1,9,10
9 D=A(N,N)*D
10 RETURN
END
  
```

10.55 Matrix Inversion

We have given definitions and rules for the addition, subtraction, and multiplication of matrices which parallel to some extent the rules of ordinary algebra. As yet we have not mentioned division, for the very good reason that division as such is not defined for matrices. There is another operation which serves a somewhat analogous purpose, however. That operation is the "inversion" of a matrix.

In ordinary algebra, b/a stands for the number which, when multiplied by a , gives b . Thus, if $ax = b$, we can say that $x = b/a$. Instead of treating division in this manner, we could define an "inverse" of a number as follows: For any number a , the inverse, a^{-1} , is that number which, when multiplied by a gives 1. Every nonzero number has a unique inverse; for example, the inverse of 2 is .5, and .5 is the *only* inverse of 2. Then if we have $ax = b$, we do not even have to have a process of division in order to find x , for we can multiply both sides of the equation by a^{-1} , giving

$$a^{-1}ax = a^{-1}b \quad \text{or} \quad x = a^{-1}b.$$

For square matrices, we define the inverse in a manner analogous to that above. For a square matrix A of order n , the inverse matrix, A^{-1} is that matrix which when multiplied by A gives the identity matrix of order n ; that is,

$$AA^{-1} = I$$

Example 1. Show that the inverse of

$$A^{-1} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}$$


```

200 A(I,J)=0.
    DO 300 I=1,N
300 A(I,N+1)=1.
    K=1
    1 CALL EXCH3(A,N,N2,K,1D)
    2 IF(A(K,K))3,999,3
    3 KK=K+1
      DO 4 J=KK,N2
        A(K,J)=A(K,J)/A(K,K)
      DO 4 I=1,N
        IF(K-I)41,4,41
41 W=A(I,K)*A(K,J)
  A(I,J)=A(I,J)-W
  IF(ABS(A(I,J))- .0001*ABS(W))42,4,4
42 A(I,J)=0.
4 CONTINUE
  K=KK
  IF(K-N)1,2,5
5 DO 10 J=1,N
  DO 10 I=1,N
  IF(1D(J)-1)10,8,10
8 DO 10 K=1,N
  AINV(I,K)=A(J,N+K)
10 CONTINUE
  RETURN
999 PRINT 1000
  RETURN
1000 FORMAT(19H MATRIX IS SINGULAR)
  END

```

— CAMBIARE
EGG. X COL.

In this subroutine the statements through 300 move the quantities to working storage to form the array depicted by (10-9). Statement 1 calls a version of SUBROUTINE EXCH given in Section 10.2. It is called EXCH3, to indicate that it must be a modified version of that subroutine with dimension statement changed to read

DIMENSION A(20,40)

Statements down to statement 4 parallel SUBROUTINE ELIM, except that the accuracy flag shown in Figure 10-9 has been inserted at statement 42. In the loops terminating on statement 10, instead of setting individual values of the X(I)'s, the subroutine sets entire rows of the inverse matrix AINV(I,K).

Once an inverse matrix has been obtained, an improved accuracy version can be obtained in a relatively straightforward manner. Let A be the matrix

to be inverted, and let D_1 be the approximate inverse produced by the above routine. Then, because of inaccuracies,

$$AD_1 \neq I$$

but instead

$$I - AD_1 = F_1$$

where F_1 is a matrix which, if D_1 was a reasonably good estimate, has small elements. If all the elements of F_1 are less than one in absolute value, then the matrix D_2 defined by

$$D_2 = D_1(I + F_1)$$

is an improved estimate of A^{-1} . If the error matrix $F_2 = I - AD_2$ still has elements which are too large, then the matrix D_3 defined by $D_3 = D_2(I + F_2)$ is a still better estimate, and so on. Thus repetition of a process involving some matrix multiplications can be used to improve the accuracy of the inverse to the extent desired, within the limits imposed by the usual problems of approximate arithmetic on computers.

EXERCISE 27

1. Given the following matrices

$$A = \begin{pmatrix} 2 & 2 \\ 1 & -1 \\ -2 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & -1 \\ 2 & 3 \end{pmatrix}, \quad C = \begin{pmatrix} 2 \\ 1 \\ -2 \end{pmatrix}$$

$$D = \begin{pmatrix} 4 & 1 & 3 \\ 2 & -1 & 1 \\ -3 & 2 & 1 \end{pmatrix}, \quad E = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad F = \begin{pmatrix} 1 & -1 & 2 \\ 2 & -3 & 1 \end{pmatrix}$$

evaluate the following expressions, or, if the expression is meaningless, so state.

- | | | |
|-------------|--------------|-------------|
| a. AB | b. DC | c. BE |
| d. BA | e. ABE | f. $DA + A$ |
| g. $FA + B$ | h. $FC + BE$ | i. $FDABE$ |
| j. $AF + D$ | | |

2. Using the method of Section 10.55, invert the following matrices.

a. $\begin{pmatrix} 3 & 2 \\ 4 & 3 \end{pmatrix}$ b. $\begin{pmatrix} 1 & -3 \\ 2 & 4 \end{pmatrix}$

c. $\begin{pmatrix} 2 & 3 & 1 \\ 1 & -1 & 2 \\ -3 & 1 & -1 \end{pmatrix}$ d. $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 2 \\ 0 & 4 & 3 \end{pmatrix}$

3. Find A^{-1} , then solve $Ax = b$ by multiplying both sides by A^{-1} , if

$$\text{a. } A = \begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix}, \quad b = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$\text{b. } A = \begin{pmatrix} 2 & 4 & -1 \\ -1 & -3 & 1 \\ 3 & -1 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 4 \\ -2 \\ 6 \end{pmatrix}$$

4. Write a FORTRAN subroutine INVIMP that will take the trial inverse obtained from MATINV and use the method described at the end of Section 10.55 to improve the inverse until all the elements of the error matrix F_n are less than .001 in absolute value.

10.6 OVERDETERMINED AND UNDERDETERMINED SYSTEMS OF LINEAR EQUATIONS

In several of the preceding sections, methods were discussed for solving systems of linear equations. In all these discussions it was assumed that there was a unique solution and that there were just as many equations as unknowns. Further, it was tacitly assumed that the equations were nonhomogeneous, that is, not all the constant terms were zero, and also that the determinant of the coefficients was not zero. With these conditions satisfied there is a unique solution. In many important cases, however, these conditions are not all satisfied—yet there may still be a unique solution, or there may be no solution or an infinite number of solutions. In this section we will discuss a method for finding which situation prevails and for completely describing the solutions when there is an infinite number of them.

10.61 Rank of a Matrix

As a tool for further study of systems of equations we will need the concept of rank of a matrix.

Definition. The rank of a matrix is the order of the highest-order nonvanishing determinant within the matrix.

By a "determinant within the matrix" we mean any determinant that can be made by crossing out rows or columns in the matrix.

Example 1. Find the rank of the matrix

$$\begin{pmatrix} -1 & 1 & 2 \\ -3 & 3 & 1 \end{pmatrix}$$

The largest-order determinant we can construct is second order, so the rank is 2 or less. To see if it is 2, we must check all second-order determinants. If we cross out the third column, we can construct the determinant

$$\begin{vmatrix} -1 & 1 \\ -3 & 3 \end{vmatrix}$$

which has the value zero. Since this one vanishes, we must check other second-order determinants. Crossing out the second column in the matrix, we obtain the determinant

$$\begin{vmatrix} -1 & 2 \\ -3 & 1 \end{vmatrix}$$

which has the value 5. Since there is a nonvanishing second-order determinant, the rank is 2.

Example 2. Find the rank of the matrix

$$\begin{pmatrix} 1 & 2 & 3 \\ -1 & -2 & -3 \\ 2 & 4 & 6 \end{pmatrix}$$

The largest-order determinant we can construct is third order, so the rank is 3 or less. The only third-order determinant is

$$\begin{vmatrix} 1 & 2 & 3 \\ -1 & -2 & -3 \\ 2 & 4 & 6 \end{vmatrix} = 0$$

so the rank is not 3. If we cross out the third row and third column, we have the determinant

$$\begin{vmatrix} 1 & 2 \\ -1 & -2 \end{vmatrix} = 0$$

Similarly, if we check all other second-order determinants, we find that they all vanish.

Hence the rank is less than 2. If we cross out the second and third rows, and the second and third columns, we can form the determinant $|1| = 1$. Since the highest-order nonvanishing determinant is first order, the rank of the matrix is 1.

It is seen from the above examples that finding the rank of a matrix is a straightforward process. For matrices of higher order, however, the process

as just demonstrated is extremely laborious, sometimes involving the evaluation of many determinants. Fortunately, however, a less laborious method is available, based on the following theorem:

Theorem 1. *The rank of a matrix is unchanged if any multiple of the elements of one row (or column) is added to the corresponding elements of another row (or column).*

This theorem means that we can proceed, just as in evaluating a determinant, to combine rows or columns to obtain zeros where we choose.

Example 3. Find the rank of

$$\begin{pmatrix} 1 & -1 & -1 & -2 \\ 2 & 1 & -2 & 2 \\ 4 & 3 & -4 & 6 \end{pmatrix}$$

Using Theorem 1, we may proceed as follows:

$$\begin{aligned} \text{rank} \begin{pmatrix} 1 & -1 & -1 & -2 \\ 2 & 1 & -2 & 2 \\ 4 & 3 & -4 & 6 \end{pmatrix} &= \text{rank} \begin{pmatrix} 1 & -1 & -1 & -2 \\ 0 & 3 & 0 & 6 \\ 4 & 3 & -4 & 6 \end{pmatrix} \begin{array}{l} \text{(twice first} \\ \text{row subtracted} \\ \text{from second)} \end{array} \\ &= \text{rank} \begin{pmatrix} 1 & -1 & -1 & -2 \\ 0 & 3 & 0 & 6 \\ 0 & 7 & 0 & 14 \end{pmatrix} \begin{array}{l} \text{(four times first} \\ \text{row subtracted} \\ \text{from third)} \end{array} \\ &= \text{rank} \begin{pmatrix} 1 & -1 & -1 & -2 \\ 0 & 3 & 0 & 6 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{array}{l} \text{(7/3 times second} \\ \text{row subtracted} \\ \text{from third)} \end{array} \end{aligned}$$

It is obvious in this last matrix all third-order determinants are zero, but at least one second-order determinant,

$$\begin{vmatrix} 1 & -1 \\ 0 & 3 \end{vmatrix}$$

is not zero. Hence the rank of the original matrix is 2.

Note that in the above example, we have *not* said the *matrices* obtained at each step are equal, but only that the *ranks* are equal. Each step has created a new matrix, one differing from the preceding in many respects, but having the rank in common.

It is seen that the method of determining rank as demonstrated in Example 3 is closely akin to the method of evaluation of a determinant given in Section

10.54. Minor modifications to the program given there will give a program for finding the rank of a matrix with no more effort than that involved in evaluating the largest determinant in the matrix.

The FORTRAN subroutine below finds the rank, K , of a matrix having N rows and M columns, where neither N nor M exceed 20.

```

SUBROUTINE MARANK(AA,N,M,K)
DIMENSION AA(20,20),A(20,20)
DO 100 I=1,N
DO 100 J=1,M
100 A(I,J)=AA(I,J)
K=1
1 CALL EXCH2(A,N,M,K)
IF(A(K,K))2,10,2
2 IF(K-N)3,11,11
3 IF(K-M)40,11,11
40 KK=K+1
DO 4 J=KK,M
A(K,J)=A(K,J)/A(K,K)
DO 4 I=KK,N
W=A(I,K)*A(K,J)
A(I,J)=A(I,J)-W
IF(ABS(A(I,J))- .0001*ABS(W))42,4,4
42 A(I,J)=0.
4 CONTINUE
K=KK
GO TO 1
10 K=K-1
11 RETURN
END

```

10.62 Consistent and Inconsistent Equations

A set of linear equations

$$a_{11}x_1 + \cdots + a_{1m}x_m = b_1$$

$$a_{21}x_1 + \cdots + a_{2m}x_m = b_2$$

$$\vdots$$

$$a_{n1}x_1 + \cdots + a_{nm}x_m = b_n$$

is said to be consistent if there exists at least one solution and inconsis-

if there is no solution. We are now in a position to give a criterion for determining whether a set of equations is consistent or inconsistent. We will refer to the matrix

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}$$

as the *coefficient* matrix, and to the matrix

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2m} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} & b_n \end{pmatrix}$$

as the *augmented* matrix. Then the following theorem applies:

Theorem 2. *A set of linear equations is consistent if and only if the coefficient matrix and augmented matrix have the same rank.*

Example 1. Determine if the following equations are consistent:

$$\begin{aligned} x + 3y &= 4 \\ 2x + 6y &= 2 \end{aligned}$$

The coefficient matrix is

$$\begin{pmatrix} 1 & 3 \\ 2 & 6 \end{pmatrix}$$

which has rank 1.

The augmented matrix is

$$\begin{pmatrix} 1 & 3 & 4 \\ 2 & 6 & 2 \end{pmatrix}$$

which has rank 2.

Hence the system is inconsistent.

Example 2. Determine if the following equations are consistent:

$$\begin{aligned} x + 2y &= 3 \\ 2x - y &= 2 \\ 3x + y &= 5 \end{aligned}$$

The coefficient matrix is

$$\begin{pmatrix} 1 & 2 \\ 2 & -1 \\ 3 & 1 \end{pmatrix}$$

which has rank 2.

The augmented matrix is

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & -1 & 2 \\ 3 & 1 & 5 \end{pmatrix}$$

which has rank 2.

Hence the equations are consistent, and there is a solution, despite the fact that there are more equations than unknowns! Upon closer scrutiny, it will be observed that the third equation is merely the sum of the first two.

The last example illustrates an important principle, that consistency or inconsistency cannot be ascertained merely from the numbers of equations and unknowns. A system with more equations than unknowns can be consistent, and a system with more unknowns than equations can be inconsistent. The subroutine for finding rank given in Section 10.62 is the tool needed to investigate consistency in the larger systems.

10.63 Linear Independence of Vectors

Consistent systems of linear equations may have infinitely many solutions. It is possible, however, to investigate these solutions systematically and to characterize them completely. To do so we need first the concept of linear dependence and independence. Consider the set of column vectors

$$\mathbf{u}_1 = \begin{pmatrix} u_{11} \\ u_{21} \\ \vdots \\ u_{n1} \end{pmatrix}, \quad \mathbf{u}_2 = \begin{pmatrix} u_{12} \\ u_{22} \\ \vdots \\ u_{n2} \end{pmatrix}, \quad \dots, \quad \mathbf{u}_r = \begin{pmatrix} u_{1r} \\ u_{2r} \\ \vdots \\ u_{nr} \end{pmatrix}$$

If c_1, c_2, \dots, c_r are any constants, the expression

$$c_1 \mathbf{u}_1 + c_2 \mathbf{u}_2 + \cdots + c_r \mathbf{u}_r$$

is called a "linear combination" of the vectors $\mathbf{u}_1, \dots, \mathbf{u}_r$. If there is some set of constants c_1, \dots, c_r , not all zero, such that

$$c_1 u_1 + c_2 u_2 + \cdots + c_r u_r = 0$$

then the vectors are said to be "linearly dependent." If, on the other hand, every linear combination of the vectors u_1, \dots, u_r is nonzero except for the case $c_1 = c_2 = \cdots = c_r = 0$, then the vectors are said to be "linearly independent."

Example 1. Are the vectors

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

linearly independent?

The sum

$$c_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + c_2 \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} c_2 \\ c_1 \end{pmatrix}$$

is zero only if both c_1 and c_2 are zero. Hence they are linearly independent.

Example 2. Are the vectors

$$\begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}, \quad \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}, \quad \text{and} \quad \begin{pmatrix} 3 \\ 0 \\ 3 \end{pmatrix}$$

linearly independent?

The sum

$$c_1 \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix} + c_2 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + c_3 \begin{pmatrix} 3 \\ 0 \\ 3 \end{pmatrix} = \begin{pmatrix} c_1 + 2c_2 + 3c_3 \\ -c_1 + c_2 \\ 2c_1 + c_2 + 3c_3 \end{pmatrix}$$

is zero if $c_1 = 1$, $c_2 = 1$, $c_3 = -1$. Hence the vectors are not linearly independent.

10.64 Complete Solution of Systems of Linear Equations

The following theorem gives a complete picture of the situation regarding solutions for systems of linear equations.

Theorem 3. Let $Ax = b$ be a consistent system having m unknowns, and let the rank of A be r . Then:

(1) If $r = m$, there is a unique solution vector x .

(2) If $r < m$, then there is at least one solution vector x . In addition, $m - r$ linearly independent vectors u_1, u_2, \dots, u_{m-r} can be found which are solutions to the set of homogeneous equations $Ax = 0$. The vector x plus any linear combination of these is also a solution of the given equation, and there are no other solutions. If $b = 0$, the vector x can be taken as $x = 0$.

Hereafter we will refer to the vector x described in this theorem as the *particular solution*.

A method of obtaining all these solutions in a systematic fashion is illustrated by the example below.

Example 1. Solve the system

$$\begin{pmatrix} 4 & 2 & -1 & 1 \\ 1 & -1 & 2 & -1 \\ 3 & 3 & -3 & 2 \\ 2 & -2 & 4 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 6 \\ 1 \\ 5 \\ 2 \end{pmatrix}$$

We will proceed as in the elimination method as illustrated in Section 10.2. Dividing the first equation by 4 and using it to eliminate x_1 from the remaining equations,

$$\begin{pmatrix} 1 & .5 & -.25 & .25 \\ 0 & -1.5 & 2.25 & -1.25 \\ 0 & 1.5 & -2.25 & 1.25 \\ 0 & -3 & 4.5 & -2.5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1.5 \\ -.5 \\ .5 \\ -1 \end{pmatrix}$$

Rearranging to make the largest element to be in the proper position,

$$\begin{pmatrix} 1 & -.25 & .5 & .25 \\ 0 & 4.5 & -3 & -2.5 \\ 0 & -2.25 & 1.5 & 1.25 \\ 0 & 2.25 & -1.5 & -1.25 \end{pmatrix} \begin{pmatrix} x_1 \\ x_3 \\ x_2 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1.5 \\ -1 \\ .5 \\ -.5 \end{pmatrix}$$

Dividing the second equation by 4.5 and using it to eliminate x_3 from the other equations,

$$\begin{pmatrix} 1 & 0 & 1/3 & 1/9 \\ 0 & 1 & -2/3 & -5/9 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_3 \\ x_2 \\ x_4 \end{pmatrix} = \begin{pmatrix} 13/9 \\ -2/9 \\ 0 \\ 0 \end{pmatrix}$$

At this point we see that the rank of A is 2 and that the system now has

two equations. If the system had been inconsistent, there would be more than two nonzero elements remaining on the right-hand side of the equation at this point.

Since there are four unknowns and the rank of A is 2, Theorem 2 tells us that the complete solution is made up of a particular solution and any linear combination of two linearly independent solution vectors.

We can find the particular solution by setting $x_2 = x_4 = 0$. Then the system becomes

$$\begin{aligned}x_1 &= 13/9 \\x_3 &= -2/9\end{aligned}$$

Hence the particular solution is

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 13/9 \\ 0 \\ -2/9 \\ 0 \end{pmatrix}$$

To find two linearly independent solution vectors, we take the homogeneous equation

$$\begin{pmatrix} 1 & 0 & 1/3 & 1/9 \\ 0 & 1 & -2/3 & -5/9 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_3 \\ x_2 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

and choose arbitrary values for x_2 and x_4 .

Taking $x_2 = 1, x_4 = 0$, we have

$$\begin{aligned}x_1 + 1/3 &= 0 \\x_3 - 2/3 &= 0\end{aligned}$$

which has the solution

$$x_1 = -1/3, \quad x_3 = 2/3$$

so one of the linearly independent vectors is

$$u_1 = \begin{pmatrix} -1/3 \\ 1 \\ 2/3 \\ 0 \end{pmatrix}$$

Taking $x_2 = 0, x_4 = 1$, we have

$$\begin{aligned}x_1 + 1/9 &= 0 \\x_3 - 5/9 &= 0\end{aligned}$$

which has the solution

$$x_1 = -1/9, \quad x_3 = 5/9$$

and so the other solution is

$$u_2 = \begin{pmatrix} -1/9 \\ 0 \\ 5/9 \\ 1 \end{pmatrix}$$

and the general solution is

$$x = \begin{pmatrix} 13/9 \\ 0 \\ -2/9 \\ 0 \end{pmatrix} + c_1 \begin{pmatrix} -1/3 \\ 1 \\ 2/3 \\ 0 \end{pmatrix} + c_2 \begin{pmatrix} -1/9 \\ 0 \\ 5/9 \\ 1 \end{pmatrix}$$

where c_1 and c_2 are arbitrary constants.

For convenience in organizing a computer solution, we note that these vectors (apart from a constant multiple of -1 in some cases) can be obtained from the last set of equations by the following somewhat artificial steps:

(1) Add -1 's down the last two columns of the diagonal of the coefficient matrix so that it becomes

$$\begin{pmatrix} 1 & 0 & 1/3 & 1/9 \\ 0 & 1 & -2/3 & -5/9 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

(2) Rearrange these last two columns and the column of constants as if they were ordered just as the x 's are:

$$\begin{pmatrix} x_1 \\ x_3 \\ x_2 \\ x_4 \end{pmatrix}$$

and needed to be correctly ordered. They become

$$\begin{pmatrix} 1/3 & 1/9 & 13/9 \\ -1 & 0 & 0 \\ -2/3 & -5/9 & -2/9 \\ 0 & -1 & 0 \end{pmatrix}$$

The column of constants has become the particular solution and the other two columns two linearly independent vectors that can be used to form the complete solution.

The method just demonstrated is a general one, and can be used for computer solution of larger systems. It requires only a few modifications and extensions of the elimination method given in Section 10.2.

The FORTRAN subroutine given below solves a system of N equations in M unknowns, where N and M are both 20 or less. Inputs are AA , the coefficient matrix; BB , the constant vector; and NI and M , the dimensions of the system. Outputs are: X , a particular solution vector, K , the number of linearly independent solution vectors for the homogeneous system, and U , a set of linearly independent solution vectors.

```

SUBROUTINE LINEQ(AA,NI,M,BB,X,K,U)
DIMENSION AA(20,20),BB(20),A(20,21),X(20),ID(20),U(20,20)
N=NI
MM=M+1
DO 100 I=1,N
A(I,MM)=BB(I)
DO 100 J=1,M
100 A(I,J)=AA(I,J)
K=1
IF(N-M)200,1,1
200 NP=N+1
N=M
DO 300 I=NP,M
DO 300 J=1,MM
300 A(I,J)=0.
K=1
1 CALL EXCH(A,M,MM,K,ID)
IF(A(K,K))2,5,2
2 KK=K+1
DO 3 J=KK,MM
A(K,J)=A(K,J)/A(K,K)
DO 3 I=1,N
IF(K-I)31,3,31
31 W=A(I,K)*A(K,J)
A(I,J)=A(I,J)-W
IF(ABS(A(I,J))- .0001*ABS(W))32,3,3

```

```

32 A(I,J)=0.
3 CONTINUE
K=KK
IF(K-M)1,2,7
5 DO 6 J=K,M
A(J,J)=-1.
DO 7 I=K,N
IF(A(I,MM))999,7,999
7 CONTINUE
DO 10 I=1,M
DO 10 J=1,M
IF(ID(J)-1)10,8,10
8 X(I)=A(J,MM)
IF(K-MM)9,10,10
9 KM=K-1
DO 10 IP=K,M
U(I,IP-KM)=A(J,IP)
10 CONTINUE
K=M-K
RETURN
999 PRINT 1000
RETURN
1000 FORMAT(27H EQUATIONS ARE INCONSISTENT)
END

```

10.7 EIGENVALUES AND EIGENVECTORS

A surprisingly large number of problems in physics and engineering can be reduced to the following mathematical problem: Given a square n th-order matrix A , find a nonzero vector x and a constant λ such that

$$Ax = \lambda x$$

That is, find a vector x such that Ax is simply a multiple of the vector x itself. We can rewrite this equation as

$$Ax - \lambda x = 0$$

or

$$(A - \lambda I)x = 0 \quad (10-10)$$

In this form, the equation appears as a set of homogeneous, linear equations

for x_1, x_2, \dots, x_n . The matrix of coefficients is $(A - \lambda I)$, and the augmented matrix is the same with a column of zeros added, so by Theorem 2 of Section 10.62 the equations are consistent. By Theorem 3 of Section 10.63 there is a unique solution if the rank of the coefficient matrix is n . We already know that solution; it is $x_1 = x_2 = \dots = x_n = 0$. Hence there is a nonzero vector \mathbf{x} only if the rank of $(A - \lambda I)$ is less than n . This will be true if

$$\det(A - \lambda I) = 0 \quad (10-11)$$

If this determinant is zero, then by Theorem 3 of Section 10.64 there are one or more linearly independent solution vectors that can be used to describe the complete solution. Thus we are interested in the values of λ for which

$$\begin{vmatrix} a_{11} - \lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} - \lambda & & a_{2n} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} - \lambda \end{vmatrix} = 0$$

In Section 10.54 it was stated that the value of a determinant could be obtained by forming all possible terms containing as factors exactly one element from each row and each column. If we were to attempt to do this with the determinant above, we would find that the various terms would contain different powers of λ . If we were to collect the terms having like powers, we would obtain an expression of the form

$$(-1)^n [\lambda^n - p_1 \lambda^{n-1} - p_2 \lambda^{n-2} - \cdots - p_n] \quad (10-12)$$

where the constants p_1, p_2, \dots, p_n are numbers resulting from some very complicated manipulations of the numbers a_{ij} in the determinant.

From Chapter 9, there are exactly n values of λ (not necessarily distinct) which will make (10-12) be equal to zero. These values are called the "eigenvalues" (or "characteristic roots," or "latent roots," or "proper values") of the matrix A . For any eigenvalue λ_i , the vector \mathbf{x} which satisfies equation (10-10) is called the "eigenvector" (or "characteristic vector," or "latent vector," or "proper vector") corresponding to λ_i . The polynomial (10-12) is called the "characteristic polynomial" of the matrix A , and the equation

$$\lambda^n - p_1 \lambda^{n-1} - p_2 \lambda^{n-2} - \cdots - p_n = 0 \quad (10-13)$$

is called the "characteristic equation."

Example 1. Find the eigenvalues and eigenvectors for the matrix

$$\begin{pmatrix} 1 & 3 \\ 2 & 2 \end{pmatrix}$$

To find the eigenvalues, we set

$$\begin{vmatrix} 1 - \lambda & 3 \\ 2 & 2 - \lambda \end{vmatrix} = 0$$

Expanding, we obtain the characteristic equation

$$(1 - \lambda)(2 - \lambda) - 6 = \lambda^2 - 3\lambda - 4 = 0$$

This factors into

$$(\lambda - 4)(\lambda + 1) = 0$$

so the eigenvalues are

$$\lambda_1 = 4, \quad \lambda_2 = -1$$

To find the eigenvector corresponding to λ_1 , we set

$$\begin{pmatrix} 1 - \lambda_1 & 3 \\ 2 & 2 - \lambda_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0$$

or

$$\begin{pmatrix} -3 & 3 \\ 2 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0$$

Since there are two unknowns and the coefficient matrix has rank 1, Theorem 3 of Section 10.63 tells us that these equations have one linearly independent vector solution U_1 , and that all other solutions are multiples of this one. We see by inspection that the vector

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

is a solution, and hence is an eigenvector corresponding to λ_1 . All other solutions are of the form

$$c_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

where c_1 is an arbitrary constant. Hence the eigenvector is really determined only up to an arbitrary constant multiple.

To find the eigenvector corresponding to λ_2 , we set

$$\begin{pmatrix} 1 - \lambda_2 & 3 \\ 2 & 2 - \lambda_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \mathbf{0}$$

or

$$\begin{pmatrix} 2 & 3 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \mathbf{0}$$

Again there is one linearly independent solution vector. We see by inspection that

$$\begin{pmatrix} 3 \\ -2 \end{pmatrix}$$

is a solution. All solutions are of the form

$$c_2 \begin{pmatrix} 3 \\ -2 \end{pmatrix}$$

where c_2 is an arbitrary constant.

Hence the eigenvalues are

$$4, \quad -1$$

and the corresponding eigenvectors are

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 3 \\ -2 \end{pmatrix}$$

(We ordinarily ignore the arbitrary constant multiple when writing an eigenvector.)

Example 2. Find the eigenvalues and eigenvectors for the matrix

$$\begin{pmatrix} 3 & 2 & 4 \\ 1 & 4 & 4 \\ -1 & -2 & -2 \end{pmatrix}$$

To determine the eigenvalues, we set

$$\begin{vmatrix} 3 - \lambda & 2 & 4 \\ 1 & 4 - \lambda & 4 \\ -1 & -2 & -2 - \lambda \end{vmatrix} = 0$$

Expanding, we obtain the characteristic equation

$$-\lambda^3 + 5\lambda^2 - 8\lambda + 4 = 0$$

which has the roots

$$\lambda_1 = 1, \quad \lambda_2 = \lambda_3 = 2$$

To find the eigenvector corresponding to λ_1 , we set

$$\begin{pmatrix} 3 - \lambda_1 & 2 & 4 \\ 1 & 4 - \lambda_1 & 4 \\ -1 & -2 & -2 - \lambda_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{0}$$

or

$$\begin{pmatrix} 2 & 2 & 4 \\ 1 & 3 & 4 \\ -1 & -2 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{0}$$

The coefficient matrix has rank 2, so this system has one linearly independent vector solution. If we solve by the method of Section 10.64, we find that the eigenvector is

$$\begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$$

To find the eigenvector corresponding to λ_2 , we set

$$\begin{pmatrix} 3 - \lambda_2 & 2 & 4 \\ 1 & 4 - \lambda_2 & 4 \\ -1 & -2 & -2 - \lambda_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{0}$$

or

$$\begin{pmatrix} 1 & 2 & 4 \\ 1 & 2 & 4 \\ -1 & -2 & -4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{0}$$

The coefficient matrix has rank 1, so this system has two linearly independent vector solutions. Solving by the method of Section 10.64, we find two linearly independent eigenvectors,

$$\begin{pmatrix} -4 \\ 0 \\ 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix}$$

The root λ_3 , being the same as λ_2 , has the same eigenvectors. Hence we have a single root, 1, with its eigenvector

$$\begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$$

and a double root, 2, with two eigenvectors:

$$\begin{pmatrix} -4 \\ 0 \\ 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix}$$

Example 3. Find the eigenvalues and eigenvectors for the matrix

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -8 & -12 & -6 \end{pmatrix}$$

We set

$$\begin{vmatrix} -\lambda & 1 & 0 \\ 0 & -\lambda & 1 \\ -8 & -12 & -6-\lambda \end{vmatrix} = 0$$

and obtain the characteristic equation

$$-\lambda^3 - 6\lambda^2 - 12\lambda - 8 = 0$$

which has the roots

$$\lambda_1 = -2, \quad \lambda_2 = -2, \quad \lambda_3 = -2$$

To find the eigenvectors, we set

$$\begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ -8 & -12 & -4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 0$$

Since the coefficient matrix has rank 2, there is only one linearly independent eigenvector. It turns out to be

$$\begin{pmatrix} 1 \\ -2 \\ 4 \end{pmatrix}$$

Since all roots are the same, we can obtain no more eigenvectors. Hence in this case we have a triple eigenvalue, -2 , and only one eigenvector (which might be considered an eigenvector of multiplicity 3):

$$\begin{pmatrix} 1 \\ -2 \\ 4 \end{pmatrix}$$

The above examples have illustrated all the possibilities concerning real eigenvalues and their corresponding eigenvectors. These possibilities can be summarized in the following theorem.

Theorem 4. An n th-order square matrix has n eigenvalues. If these are discrete, there is one eigenvector corresponding to each eigenvalue. If an eigenvalue is of multiplicity r , it may have from one to r linearly independent eigenvectors associated with it.

10.71 Program for Largest Eigenvalue and Eigenvector

Suppose that the matrix A has one eigenvalue λ which is larger than all others in absolute value, and y is any nonzero column vector conformable with A . Let the vectors y_1, y_2, \dots , be defined by

$$\begin{aligned} y_1 &= Ay_1 \\ y_2 &= Ay_1 \\ &\vdots \\ y_n &= Ay_{n-1} \end{aligned} \tag{10.14}$$

The vectors y_i defined in this manner can lead to the value of λ_1 and to x_1 , the eigenvalue corresponding to λ_1 . The method of obtaining the eigenvalue and eigenvector will be illustrated without proof.* In order to provide the illustration, let us first write a remote-terminal program to perform the computations indicated by expression (10-14). A suitable program is

* See, for example, J. G. Herriot, *Methods of Mathematical Analysis and Computation*, John Wiley & Sons, Inc., New York, 1963.

```

1  DIMENSION A(10,10),Y(10),YN(10)
2  PRINT, "INPUT N, TEN OR LESS"
3  INPUT, N
4  PRINT, "INPUT A(1,1)A(1,2),,,A(N,N)"
5  INPUT, ((A(I,J),J=1,N),I=1,N)
6  PRINT, "INPUT Y(1),Y(2),,,Y(N)"
7  INPUT, (Y(I),I=1,N)
8  1 DO 2 I=1,N
9  YN(I)=0
10 DO 2 J=1,N
11 2 YN(I)=YN(I)+A(I,J)*Y(J)
12 PRINT, (YN(I),I=1,N)
13 INPUT, Q
14 DO 3 I=1,N
15 3 Y(I)=YN(I)
16 GO TO 1
17 END

```

In this program, the statements at lines 2 through 7 allow the user to input an initial matrix A and vector y of order up to 10. The statements at lines 8 through 12 compute and print the vector $y_1 = Ay$. At line 13, the user is allowed to specify whether another step of the process is required. If the typed entry is the letter S, the program will terminate. If the entry is any number whatsoever, the program will cause y_1 to replace y , and will repeat lines 8 through 12, thereby computing and printing $y_2 = Ay_1$, and so on.

Example 1. Write all user inputs and machine responses for running the above program with the matrix

$$\begin{pmatrix} 1 & 3 \\ 2 & 2 \end{pmatrix} \text{ and the vector } \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

continuing until vectors through y_n have been generated.

The inputs and responses are

```

RUN
INPUT N, TEN OR LESS
? 2
INPUT A(1,1),A(1,2),,,A(N,N)
? 1,3,2,2
INPUT Y(1),Y(2),,,Y(N)
? 1,0
1.000000 2.000000
? 0
7.000000 6.000000

```

```

? 0
25.00000 26.00000
? 0
103.0000 102.0000
? 0
409.0000 410.0000
? 0
1639.000 1638.000
? 0
6553.000 6554.000
? 0
.2621500ES .2621400ES
? S
STOP

```

The above program was Example 1, Section 10.6, which had a largest eigenvalue of 4 and corresponding eigenvector of $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Looking at the vectors printed out above, we see that the vectors generated were

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 7 \\ 6 \end{pmatrix}, \begin{pmatrix} 25 \\ 26 \end{pmatrix}, \begin{pmatrix} 103 \\ 102 \end{pmatrix}, \begin{pmatrix} 409 \\ 410 \end{pmatrix}, \begin{pmatrix} 1639 \\ 1638 \end{pmatrix}, \begin{pmatrix} 6553 \\ 6554 \end{pmatrix}, \begin{pmatrix} 26215 \\ 26214 \end{pmatrix}$$

and that after the first few steps the components are always very nearly equal; that is, the vectors themselves are very nearly simple multiples of the vector $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Since eigenvectors are determined only up to a constant multiple, we can say that the vector y_n of (10-14) is actually approaching the eigenvector x . Now since

$$Ax = \lambda x$$

then if y_n is x , then y_{n+1} will be λx . We note without surprise, then, that in each of the vectors computed in the above example, the components are very nearly four times those of the preceding vector.

It appears, then, that the above program can be used almost directly to find the largest eigenvalue and corresponding eigenvector. Some improvement can be made by replacing the statement at line 15 by

```
15 3 Y(I)=YN(I)/YN(1)
```

This will serve to keep the components from growing at each stage, and further will cause the first component to approach the actual value of the eigenvalue. If this had been done for Example 1, the printouts would have been

1.000000 2.000000
 ? 0
 7.000000 6.000000
 ? 0
 3.571429 3.714286
 ? 0
 4.120000 4.080000
 ? 0
 3.970874 3.980583
 ? 0
 4.007335 4.004890
 ? 0
 3.998109 3.998596
 ? 0
 4.000458 4.000305
 ? S
 STOP

From these results, the eigenvalue 4 and the eigenvector $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ are apparent.

10.72 Complex Eigenvalues

From Chapter 9 it is known that the characteristic equation may have complex roots, occurring in conjugate pairs. In this case, the eigenvectors are also complex, and the equation

$$(A - \lambda I)x = 0$$

instead of being n equations in n unknowns, is really $2n$ equations in $2n$ unknowns, for both the real and imaginary part of x must satisfy the equation. Let

$$\lambda = \alpha + \beta i$$

be a complex eigenvalue, and let the eigenvector be

$$x = \begin{pmatrix} x_1 + y_1 i \\ x_2 + y_2 i \\ \vdots \\ x_n + y_n i \end{pmatrix}$$

If we substitute these in the above equation and separate real and imaginary parts, the result can be written in the form

10.7] Eigenvalues and Eigenvectors

$$\begin{pmatrix} a_{11} - \alpha & a_{12} & \cdots & a_{1n} & \beta & 0 & \cdots & 0 \\ a_{21} & a_{22} - \alpha & \cdots & a_{2n} & 0 & \beta & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} - \alpha & 0 & 0 & \cdots & \beta \\ -\beta & 0 & \cdots & 0 & a_{11} - \alpha & a_{12} & \cdots & a_{1n} \\ 0 & -\beta & \cdots & 0 & a_{21} & a_{22} - \alpha & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -\beta & a_{n1} & a_{n2} & \cdots & a_{nn} - \alpha \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = 0$$

These are $2n$ real equations in $2n$ real unknowns which can be solved for the x_i 's and y_i 's. The eigenvector corresponding to the complex conjugate of λ is the complex conjugate of the eigenvector for λ , so the process needs to be done only once for each pair of complex roots.

Example 1. Find the eigenvalues and eigenvectors of

$$\begin{pmatrix} -1 & -5 \\ 1 & 3 \end{pmatrix}$$

We set

$$\begin{vmatrix} -1 - \lambda & -5 \\ 1 & 3 - \lambda \end{vmatrix} = 0$$

and obtain the characteristic equation:

$$\lambda^2 - 2\lambda + 2 = 0$$

which has roots:

$$\lambda_1 = 1 + i, \quad \lambda_2 = 1 - i$$

To find the eigenvector corresponding to λ_1 , using the method described above, we write

$$\begin{pmatrix} -2 & -5 & 1 & 0 \\ 1 & 2 & 0 & 1 \\ -1 & 0 & -2 & -5 \\ 0 & -1 & 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{pmatrix} = 0$$

Applying the method of Section 13.34, this reduces to

$$\begin{pmatrix} 1 & 0 & -.2 & .4 \\ 0 & 1 & .4 & .2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \\ y_1 \\ x_1 \end{pmatrix} = 0$$

This has two linearly independent solution vectors

$$\begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ .2 \\ 1 \\ -.4 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 \\ -.4 \\ 0 \\ -.2 \end{pmatrix}$$

These vectors of themselves are not of interest to us, except to use the numbers in them to construct the complex of vectors

$$\begin{pmatrix} x_1 + y_1 i \\ x_2 + y_2 i \end{pmatrix} = \begin{pmatrix} i \\ .2 - .4i \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} x_1 + y_1 i \\ x_2 + y_2 i \end{pmatrix} = \begin{pmatrix} 1 \\ -.4 - .2i \end{pmatrix}$$

These two vectors are not linearly independent at all, for if we multiply the second by i , we obtain the first. Hence, we have really obtained only one eigenvector corresponding to the eigenvalue $1 + i$, and that is

$$\begin{pmatrix} i \\ .2 - .4i \end{pmatrix}$$

The eigenvector corresponding to $1 - i$ is the conjugate of this:

$$\begin{pmatrix} -i \\ .2 + .4i \end{pmatrix}$$

10.73 Determination of All Eigenvalues and Eigenvectors

The method described in Section 10.71 will provide the largest eigenvalue and corresponding eigenvector. Frequently it is necessary to find all eigen-

values and eigenvectors. From the discussions of Section 10.7, it is clear that this can be done by accomplishing the following three steps:

- (1) Find the characteristic polynomial.
- (2) Solve the characteristic equation for its roots.
- (3) Solve sets of linear equations for the eigenvectors.

Chapter 9 gave methods for solving polynomial equations, so we already have computer methods for step (2). Section 10.64 gave a computer method for solving systems of linear equations which is satisfactory for step (3). Hence the only thing really required is a computer method for generating the characteristic polynomial. In the examples above, we have used very small matrices and found the characteristic polynomial by brute-force expansion of the determinant, but this process is inefficient for large-order matrices. A more efficient method is the Leverrier-Faddeev method, which proceeds as follows:

$$\begin{aligned} \text{Let } A_1 &= A & \text{and } p_1 &= \text{tr } A \\ \text{Let } A_2 &= A(A_1 - p_1 I), & \text{and } p_2 &= (1/2) \text{tr } A_2 \\ \text{Let } A_3 &= A(A_2 - p_2 I), & \text{and } p_3 &= (1/3) \text{tr } A_3 \\ &\vdots & & \\ A_n &= A(A_{n-1} - p_{n-1} I) & \text{and } p_n &= (1/n) \text{tr } A_n \end{aligned}$$

The numbers p_1, p_2, \dots, p_n are the required coefficients in the characteristic equation

$$\lambda^n - p_1 \lambda^{n-1} - p_2 \lambda^{n-2} - \dots - p_n = 0$$

In addition, as a bonus side product of this process, it can be shown that the inverse of A is given by

$$A^{-1} = (1/p_n)(A_{n-1} - p_{n-1} I) \quad (10-15)$$

and also, as a sometimes helpful check,

$$A_n - p_n I = 0 \quad (10-16)$$

Example 1. Find the characteristic equation of

$$\begin{pmatrix} 1 & 3 & 2 \\ -2 & 1 & 1 \\ 1 & -2 & -1 \end{pmatrix}$$

Following the above procedure, we have

$$A_1 = \begin{pmatrix} 1 & 3 & 2 \\ -2 & 1 & 1 \\ 1 & -2 & -1 \end{pmatrix}, \quad p_1 = 1 + 1 - 1 = 1$$

$$A_2 = \begin{pmatrix} 1 & 3 & 2 \\ -2 & 1 & 1 \\ 1 & -2 & -1 \end{pmatrix} \begin{pmatrix} 0 & 3 & 2 \\ -2 & 0 & 1 \\ 1 & -2 & -2 \end{pmatrix} = \begin{pmatrix} -4 & -1 & 1 \\ -1 & -8 & -5 \\ 3 & 5 & 2 \end{pmatrix}$$

$$p_2 = (1/2)(-4 - 8 + 2) = -5$$

$$A_3 = \begin{pmatrix} 1 & 3 & 2 \\ -2 & 1 & 1 \\ 1 & -2 & -1 \end{pmatrix} \begin{pmatrix} 1 & -1 & 1 \\ -1 & -3 & -5 \\ 3 & 5 & 7 \end{pmatrix} = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{pmatrix}$$

$$p_3 = (1/3)(4 + 4 + 4) = 4$$

As a check, we see that

$$A_3 - p_3 I = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{pmatrix} - \begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{pmatrix} = 0$$

Hence the characteristic equation is

$$\lambda^3 - \lambda^2 + 5\lambda - 4 = 0$$

The flow chart, Figure 10-10, describes this process for an n th-order matrix. According to equation (10-16), the matrix A_n is simply the identity matrix multiplied by p_n , so only the first element of A_n need be calculated to give p_n . The value of p_n is, in fact, the determinant of A , so that if p_n is zero, the matrix is singular. If p_n is not zero, the inverse of A is easily calculated from equation (10-15), and the flow chart includes this calculation. The elements of A^{-1} are the last values obtained for f_{ij} .

The FORTRAN subroutine given below will generate coefficients in accordance with the flow chart, for matrices up to order 20. However, in order that the subscripts will match the notation in the subroutines of Chapter 9, the characteristic equation is written as

$$Q(1)\lambda^N + Q(2)\lambda^{N-1} + \dots + Q(N+1) = 0$$

The relationship between the p_k of the flow chart and the $Q(K)$ of the subroutine is given by

$$Q(1) = 1, \quad Q(K+1) = -p_k \quad \text{for } k = 1, \dots, n$$

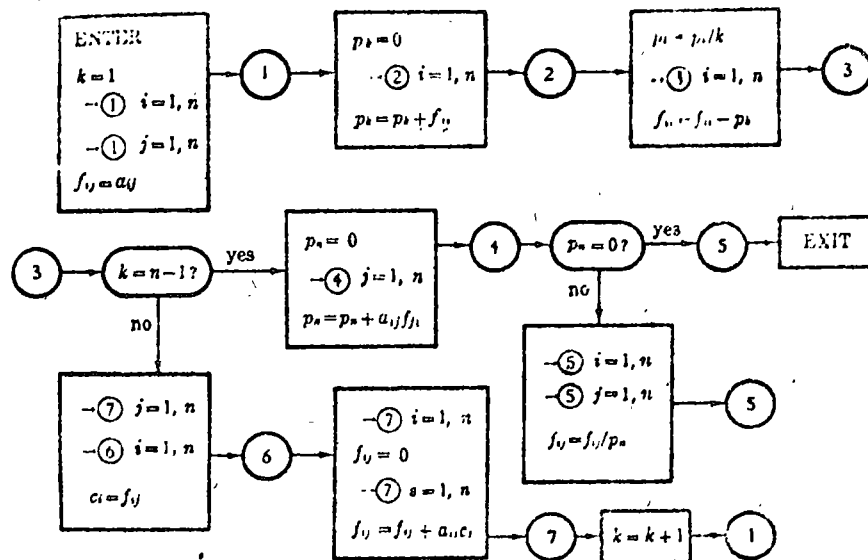


Figure 10-10: Generation of characteristic polynomial

As in the flow chart, the subscripted variable $F(I,J)$ is the inverse matrix unless $Q(N+1)$ happens to be zero.

```

SUBROUTINE CHAREQ(A,N,Q,F)
DIMENSION A(20,20),F(20,20),Q(21),C(20)
Q(1)=1.
K=1
DO 11 I=1,N
DO 11 J=1,N
11 F(I,J)=A(I,J)
1 CONTINUE
Q(K+1)=0.
DO 2 I=1,N
Q(K+1)=Q(K+1)+F(I,I)
2 CONTINUE
FK=K
Q(K+1)=-Q(K+1)/FK
DO 3 I=1,N
F(I,I)=F(I,I)+Q(K+1)
3 CONTINUE
IF(K-N+1)71,41,71
71 DO 7 J=1,N
DO 6 I=1,N
C(I)=F(I,J)
    
```

```

6 CONTINUE
DO 7 I=1,N
F(I,J)=0.
DO 7 IS=1,N
F(I,J)=F(I,J)+A(I,IS)*C(IS)
7 CONTINUE
K=K+1
GO TO 1
41 Q(N+1)=0.
DO 4 J=1,N
Q(N+1)=Q(N+1)-A(I,J)*F(J,1)
4 CONTINUE
IF(Q(N+1))51,5,51
51 DO 52 I=1,N
DO 52 J=1,N
52 F(I,J)=-F(I,J)/Q(N+1)
5 RETURN
END

```

With the above subroutine and subroutines of Chapter 9 and Section 10.6, eigenvalues and eigenvectors can be found in a systematic way. There are also methods which, under some conditions, can be used to find all eigenvalues directly from the matrix itself, without generating the characteristic equation first. These methods are available in the literature and will not be reported here.

EXERCISE 28

1. Determine the rank of the following matrices.

a. $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$

b. $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 6 \end{pmatrix}$

c. $\begin{pmatrix} 1 & 2 & 3 \\ -2 & 1 & -2 \\ -1 & 3 & 1 \end{pmatrix}$

d. $\begin{pmatrix} 2 & -1 & 3 & 4 \\ 1 & -2 & -2 & -1 \\ 0 & 3 & 7 & 6 \end{pmatrix}$

2. Determine whether the following systems are consistent or inconsistent.

a. $x + 2y + z = 4$
 $-2x - 4y - 2z = 3$

b. $x + 2y = 6$
 $x + 3y = 8$

c. $x + 3y = 7$
 $2x - y = 4$
 $4x + 5y = 18$

d. $x + 2y = 8$
 $3x - y = 2$
 $2x + y = 6$

3. Solve completely the following systems of equations.

a. $x + 2y = 0$
 $-2x - 4y = 0$

b. $x + y - z = 2$
 $x - y + z = 3$

c. $x + 3y - z = 4$
 $2x - y + 2z = 3$
 $3x + 2y + z = 7$

d. $x + 2y + z = 1$
 $2x - y + z = 2$
 $3x - y + 4z = 3$

- *4. Find all eigenvalues and eigenvectors for the following matrices.

a. $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

b. $\begin{pmatrix} 1 & 3 \\ -2 & -4 \end{pmatrix}$

c. $\begin{pmatrix} 1 & 2 \\ -2 & -3 \end{pmatrix}$

d. $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix}$

e. $\begin{pmatrix} 2 & 1 & 2 \\ -1 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$

5. Find the number of multiplications required to find the rank of a 10 by 15 matrix using SUBROUTINE MARANK of Section 10.61, if the rank turns out to be 8.
6. Write a program that will input a system of linear equations up to 20 by 20, call subroutine LINEQ of Section 10.64 to obtain all solutions, and print the result.
7. Write a program that will input a square matrix up to 19 by 19, call subroutine CHAREQ of Section 10.73 to obtain the characteristic equation, call the appropriate subroutine from Chapter 9 to find the largest real root, call subroutine LINEQ of Section 10.64 to find the corresponding eigenvector, and print the result.

I N D I C E

the problems of the third chapter are characterized by sets of simultaneous ordinary differential equations with prescribed *initial conditions*, the problems of the fourth and fifth chapters are characterized by ordinary or partial differential equations with closed *boundary conditions*, and the problems of the sixth chapter are characterized by partial differential equations with open boundary conditions. This survey of numerical procedures thus amounts to a catalogue of practical methods for the solution of algebraic, ordinary differential, and partial differential equations. In Chaps. 1, 3, 4, and 6 both linear and nonlinear problems are considered. The discussion of eigenvalue problems in Chaps. 2 and 5 is limited to linear systems.

All six chapters have the same structure. At the beginning of each chapter several representative problems are presented. These serve to identify the class of problems under consideration. The process of formulating a mathematical model is illustrated for each of these problems.

Before leaving these formulations they are each cast into dimensionless form. This is an extremely useful organizational tool¹ of the analyst. In connection with numerical calculations it removes all unnecessary symbols, leaving the basic problem in its simplest form.

Then, before surveying numerical procedures applicable to this class of problems, a brief résumé of the classical mathematical theory is given. A complete mathematical development has not been attempted but an effort has been made to describe clearly the properties of the well-behaved or regular system. The possibilities of irregular behavior are hinted at by means of simple counterexamples. Enough theory is presented to provide a background for the explanation of the success (and limitations) of the numerical procedures which follow.

After these preliminaries the actual survey of numerical procedures begins. Illustrative examples are drawn from the problems formulated at the beginning of the chapter. At the end of each section there is a set of exercises for the reader. A few of these are of the nature of drill problems but the majority represent interesting extensions or alternative developments of the text material. Answers or hints for the solution are given in most cases.

The numerical procedures described here are those which in the judgment of the author are of most potential interest to the engineering analyst. Methods for both hand and machine computation are given.

Throughout the text there are references to books and papers having direct bearing on the matter at hand. For the reader's convenience a number of selected general references are grouped together in the Bibliography at the end of the book.

¹ See, for example, H. L. Langhaar, "Dimensional Analysis and Theory of Models," John Wiley & Sons, Inc., New York, 1951.

**EQUILIBRIUM PROBLEMS IN SYSTEMS
WITH A FINITE NUMBER OF DEGREES OF FREEDOM**

The state of a physical system can often be described with adequate precision by giving the magnitudes of a finite number of state variables. This chapter deals with numerical procedures for determining steady states of such systems. The chapter begins with a preliminary examination of several particular problems. The general problem of this type is then formulated mathematically as a set of simultaneous algebraic equations. There is a review of the classical results from the theory of such systems, including a discussion of the relationship of extremum principles to equilibrium problems. Numerical procedures, both exact and approximate, are then described and illustrated by applying them to the particular problems set up at the beginning of the chapter.

1-1. Particular Examples

We begin with an assortment of examples of how mathematical formulations are set up for particular physical problems. The examples are taken from a variety of fields and, in general, have been chosen for their simplicity despite the fact that the really significant contributions of numerical procedures occur in problems of extended complexity.

It is generally recognized that the most difficult step in the whole process of engineering analysis is that in which a mathematical model is substituted for a real physical system. It is here that judgment, experience, and ingenuity of the highest order are required of the analyst. It is here that the really gross approximations and simplifications are made. In this text the basic structure of the various physical problem types and the corresponding mathematical models is emphasized.

The general equilibrium problem in a lumped-parameter system has the following structure: The given system is made up by interconnecting a number of simple elements. The equilibrium or steady-state requirements for each individual element are known. As examples we have the stress-strain law for elastic elements, Ohm's law for electrical resistances, and the pressure-flow relation for hydraulic resistances. In addition to satisfying the equilibrium requirements of the individual elements it is

also necessary to satisfy certain interconnection requirements. Thus in elastic systems we must have geometric fit and balance of forces at all joints; in electric networks we must satisfy both of Kirchhoff's laws; and in hydraulic networks we must have conservation of flow and uniqueness of pressure at every interconnection. The over-all equilibrium problem then consists in finding the state of a system which simultaneously satisfies the equilibrium requirements of the individual elements together with the interconnection requirements.

To serve as concrete illustrations of this general statement and to provide illustrative examples for the numerical procedures which follow, we here consider the following five particular equilibrium problems:

- 1-1. Elastic spring system.
- 1-2. D-c network.
- 1-3. A-c network.
- 1-4. Continuous beam.
- 1-5. Hydraulic network.

In each case the problem is cast into nondimensional form, with particular data assumed, in preparation for numerical solution. In most cases complementary forms of the problem are considered. The first four problems are linear, while the fifth represents an example of a practically important nonlinear problem.

Problem 1-1. Elastic Spring System. In Fig. 1-1 a system of four

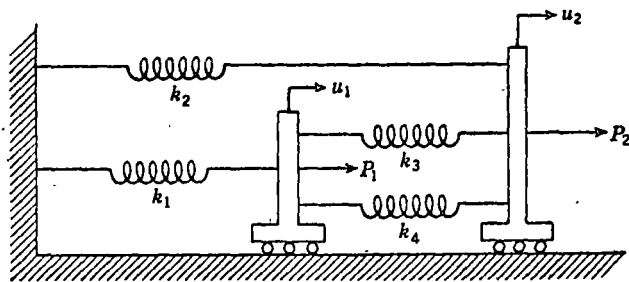


FIG. 1-1. Elastic system of interconnected springs subjected to loads P_1 and P_2 .

linear springs is shown. Assume that when P_1 and P_2 are zero then u_1 and u_2 are both zero and that all springs are in their natural positions. The problem here is to find the displacements u_1 and u_2 and the forces f_1, f_2, f_3 , and f_4 in the four springs when the loads P_1 and P_2 are applied. The fundamental requirements are:

1. Spring force = k (spring elongation) for each spring.
2. Forces should balance on each movable cart.
3. Spring elongations should be compatible with the displacements of the carts.

A standard method of solution is to choose unknown variables in such

a way that requirement 3 above is automatically satisfied. In our problem this is done by taking u_1 and u_2 as unknowns and expressing the spring elongations in terms of them (e.g., elongation of spring 4 is $u_2 - u_1$). Next the spring forces are expressed in terms of u_1 and u_2 by introducing the spring constants. Finally, writing the force-balance conditions for each cart gives us the following equations for u_1 and u_2 :

$$\begin{aligned} k_1 u_1 - k_3(u_2 - u_1) - k_4(u_2 - u_1) &= P_1 \\ k_2 u_2 + k_3(u_2 - u_1) + k_4(u_2 - u_1) &= P_2 \end{aligned} \quad (1-1)$$

A complete solution of our problem would require the solution of these simultaneous equations. We stop at this point, however, since we are here concerned only with the formulation of the problem. Summarizing, we limited ourselves to geometrically compatible states as soon as we took u_1 and u_2 as unknowns; requiring that force balance should also hold gave us (1-1).

A complementary method of solution for the same problem is to choose unknown variables in such a way that requirement 2 above is automatically satisfied. This may be done by taking the spring forces f_2 and f_3 as unknown and expressing the other spring forces, f_1 and f_4 , in terms of them by means of the force-balance conditions.

$$\begin{aligned} f_4 &= P_2 - f_2 - f_3 \\ f_1 &= P_1 + f_3 + f_4 = P_1 + P_2 - f_2 \end{aligned} \quad (1-2)$$

Next the spring elongations are expressed in terms of f_2 and f_3 by introducing the spring constants. Finally we obtain the following equations for f_2 and f_3 by requiring that the spring elongations be compatible with unique displacements of the carts:

$$\begin{aligned} \frac{f_2}{k_2} &= \frac{P_1 + P_2 - f_2}{k_1} + \frac{P_2 - f_2 - f_3}{k_4} \\ \frac{f_3}{k_3} &= \frac{P_2 - f_2 - f_3}{k_4} \end{aligned} \quad (1-3)$$

The second of these expresses the fact that the elongations of springs 3 and 4 should be the same. The first expresses the fact that the elongation of spring 2 must be the same as the sum of the elongations of springs 1 and 4. Again a complete solution would require the simultaneous solution of (1-3), but we stop at this point. Reiterating our logic, we limited ourselves to self-balancing states when we took f_2 and f_3 as unknowns and used (1-2) for the other forces. Among these self-balancing states the true state is selected by (1-3), which requires that the spring elongations should be compatible with the given interconnections of the system.

For future use we now specialize the above problem to the case where

$$\begin{aligned} k_1 &= 3k \\ k_2 &= 2k \\ k_3 &= k \\ k_4 &= k \end{aligned} \quad \begin{aligned} P_1 &= P \\ P_2 &= 2P \end{aligned} \quad (1-4)$$

Substituting these values in (1-1) and (1-3), we obtain

$$\begin{aligned} 5ku_1 - 2ku_2 &= P \\ -2ku_1 + 4ku_2 &= 2P \end{aligned} \quad (1-5)$$

as the equations for the displacements and

$$\begin{aligned} \frac{1}{3}f_2 + f_3 &= 3P \\ f_2 + 2f_3 &= 2P \end{aligned} \quad (1-6)$$

as the complementary equations for the forces. These formulations can be simplified even further by introducing dimensionless variables. If we define the nondimensional displacements

$$x_1 = \frac{u_1}{P/k} \quad x_2 = \frac{u_2}{P/k} \quad (1-7)$$

the displacement equations (1-5) can be written in the following form:

$$\begin{aligned} 5x_1 - 2x_2 &= 1 \\ -2x_1 + 4x_2 &= 2 \end{aligned} \quad (1-8)$$

Similarly, in terms of the nondimensional forces

$$y_1 = \frac{f_2}{P} \quad y_3 = \frac{f_3}{P} \quad (1-9)$$

the force equations (1-6) become

$$\begin{aligned} \frac{1}{3}y_1 + y_2 &= 3 \\ y_1 + 2y_2 &= 2 \end{aligned} \quad (1-10)$$

Problem 1-2. D-C Network. We consider the problem of determining the voltages and currents in the network shown in Fig. 1-2. The resistances and battery emfs are given in the figure in terms of R and E . The equilibrium or steady-state conditions are Ohm's law for each individual resistor plus the interconnection requirements which are the two laws of Kirchhoff.¹ We can obtain complementary formulations of the problem in the following manner: If we represent the state of the sys-

¹ See, for example, C. L. Dawes, "Electrical Engineering," 3d ed., vol. I, McGraw-Hill Book Company, Inc., New York, 1937, p. 72.

tem by a set of independent currents such that Kirchhoff's first law is automatically satisfied, we then obtain equations for determining these currents by requiring that the second law be satisfied. Alternatively if the state of the system is represented by a set of independent voltages such that Kirchhoff's second law is automatically satisfied, equations can then be obtained for determining these voltages by requiring the satisfaction of the first law.

In accordance with the first procedure the state of the system is represented by the three loop currents I_1 , I_2 , and I_3 . The net current flow into any junction is always zero for any values of I_1 , I_2 , and I_3 . Ohm's law together with the requirement that the net voltage drop in any closed loop should vanish yields the following equations:

$$\begin{aligned} 2E - RI_1 - 4R(I_1 - I_2) &= 0 \\ -RI_2 - 5R(I_2 - I_3) - 4R(I_2 - I_1) &= 0 \\ -RI_3 - 5R(I_3 - I_2) - E &= 0 \end{aligned} \quad (1-11)$$

When the currents which satisfy (1-11) are found, any desired network emf is easily obtained by an elementary application of Ohm's law.

Following the second procedure, the state of the system can be represented by the potentials e_1 and e_2 of the nodes A and B with respect to G . This ensures that the voltage drop around any closed loop vanishes. The requirement that there should be no net current flow into the nodes A and B results in the following equations:

$$\begin{aligned} \frac{2E - e_1}{R} - \frac{e_1}{4R} + \frac{e_2 - e_1}{R} &= 0 \\ \frac{E - e_2}{R} - \frac{e_2}{5R} + \frac{e_1 - e_2}{R} &= 0 \end{aligned} \quad (1-12)$$

When the voltages e_1 and e_2 which satisfy (1-12) have been found, any desired network current may be obtained by a simple application of Ohm's law.

The complete solution can thus be obtained by solving either (1-11) or (1-12). Note that here the number of degrees of freedom is not the same for the two analyses. Before leaving this problem, we cast the equations into nondimensional form. Dimensionless currents and volt-

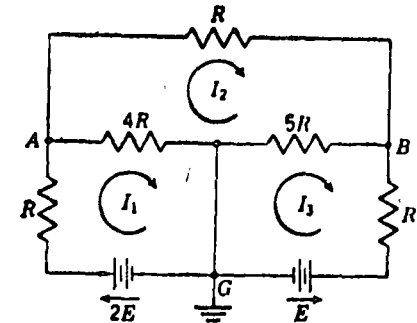


FIG. 1-2. Network of resistors and batteries.

variables are defined as follows:

$$\begin{aligned} x_1 &= \frac{I_1}{E/R} & x_2 &= \frac{I_2}{E/R} & x_3 &= \frac{I_3}{E/R} \\ y_1 &= \frac{e_1}{E} & y_2 &= \frac{e_2}{E} \end{aligned} \quad (1-13)$$

The current equations (1-11) then become

$$\begin{aligned} 5x_1 - 4x_2 &= 2 \\ -4x_1 + 10x_2 - 5x_3 &= 0 \\ -5x_2 + 6x_3 &= -1 \end{aligned} \quad (1-14)$$

while the voltage equations (1-12) take the following form:

$$\begin{aligned} 2.25y_1 - y_2 &= 2 \\ -y_1 + 2.20y_2 &= 1 \end{aligned} \quad (1-15)$$

These last two sets of equations constitute complementary dimensionless formulations of Prob. 1-2.

Problem 1-3. A-C Network. The equilibrium problem here is to determine the steady-state currents in the network of Fig. 1-3. The impedances of the branches at the frequency of the voltage source are indicated in the usual complex notation in terms of R . Complementary formulations of this problem can be obtained in the same manner as in Prob. 1-2. We consider here only the equations for the currents. If we take I_1 and I_2 as the state variables, Kirchhoff's first law is automatically satisfied and the second law yields the following equations:

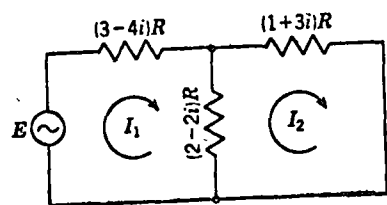


FIG. 1-3. Network of impedances connected to alternating-voltage source.

$$\begin{aligned} E - (3 - 4i)RI_1 - (2 - 2i)R(I_1 - I_2) &= 0 \\ -(2 - 2i)R(I_2 - I_1) - (1 + 3i)RI_2 &= 0 \end{aligned} \quad (1-16)$$

A nondimensional formulation is obtained by introducing the dimensionless variables

$$I'_1 = \frac{I_1}{E/R} \quad I'_2 = \frac{I_2}{E/R} \quad (1-17)$$

into (1-16) as follows:

$$\begin{aligned} (5 - 6i)I'_1 - (2 - 2i)I'_2 &= 1 \\ -(2 - 2i)I'_1 + (3 + i)I'_2 &= 0 \end{aligned} \quad (1-18)$$

The quantities I'_1 and I'_2 are expected to be complex. Although procedures exist for the direct solution of sets of equations such as (1-18), it is sometimes useful to trans-

¹ See, for example, C. L. Dawes, "A Course in Electrical Engineering," 4th ed., vol. II, McGraw-Hill Book Company, Inc., New York, 1947, p. 70. The symbol i stands for the imaginary unit $\sqrt{-1}$.

form the complex equations into their real equivalents. To illustrate this as for the present example, we define the real quantities x_1, \dots, x_4 as follows:

$$\begin{aligned} I'_1 &= x_1 + ix_2 \\ I'_2 &= x_3 + ix_4 \end{aligned} \quad (1-19)$$

When these are substituted in (1-18), each equation can be separated into two: one obtained from the real terms and one from the imaginary terms. We thus obtain the following four real equations, which are equivalent to the two complex equations of (1-18):

$$\begin{aligned} 5x_1 + 6x_2 - 2x_3 - 2x_4 &= 1 \\ 6x_1 - 5x_2 - 2x_3 + 2x_4 &= 0 \\ -2x_1 - 2x_2 + 3x_3 - x_4 &= 0 \\ -2x_1 + 2x_2 - x_3 - 3x_4 &= 0 \end{aligned} \quad (1-20)$$

Problem 1-4. Continuous Beam. In Fig. 1-4 a uniform elastic beam is shown. It is simply supported at $A, B,$ and C and clamped at D . Equilibrium problems for such systems consist in determining the bend-

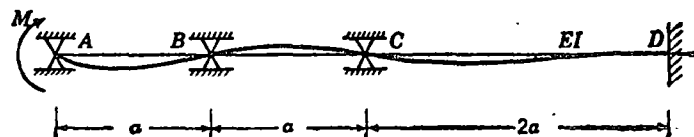


FIG. 1-4. Continuous beam freely supported at $A, B,$ and C , clamped at D , and subjected to external moment M applied at A .

ing moments and deflections resulting from assigned loads. We consider the particular problem of Fig. 1-4, where the load is the single moment M applied at A . The flexural stiffness of the beam is EI , and the span lengths are given in terms of a .

This system may be treated as a lumped parameter system by considering each span as a single element. The total equilibrium problem then involves satisfying the elastic requirements within each span, together with the interconnection requirements at the joints. These interconnection requirements are that adjacent spans should have the same inclination and the same bending moment at their common junction. The internal elastic requirements for a single span are one stage more complicated than the corresponding single-element relations in the foregoing examples. Here each span is itself a two-degree-of-freedom system described by two geometric quantities (the inclinations at the ends) and by two force quantities (the bending moments at the ends). The relations between these which represent the elastic requirements¹ are shown in Fig. 1-5. Clockwise angles have been called positive. Bending moments which tend to stretch the bottom fibers and compress the top fibers have been called positive. A formulation of the equilibrium

¹ See, for example, L. C. Maugh, "Statically Indeterminate Structures," John Wiley & Sons, Inc., New York, 1946, p. 49.

problem may be obtained by using either inclinations or bending moments to represent the state of the system. Thus a set of independent angles which satisfy the compatibility requirements might be chosen. With the aid of the elastic relations bending moments could then be expressed in terms of these angles, and finally, by writing the conditions for moment

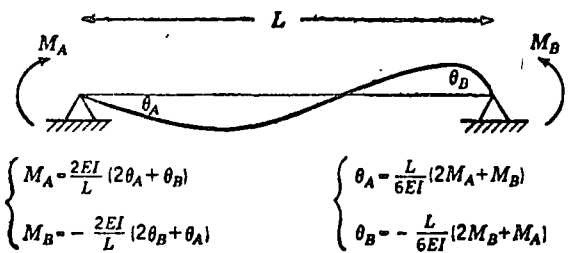


FIG. 1-5. Elastic relationships for a span whose ends are restrained from translation and which is subjected to end moments.

balance, a set of equations for determining the angles would be obtained. Alternatively a set of independent bending moments which satisfy the requirements of moment balance could be used to represent the state of the system. The compatibility requirements together with the elastic relations would then furnish equations for determining these moments.

Adopting the former procedure, the state of the system of Fig. 1-4 can

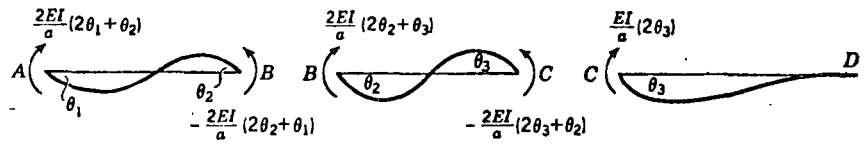


FIG. 1-6. Representation of the beam of Fig. 1-4 in terms of the displacements θ_1 , θ_2 , and θ_3 .

be represented by the clockwise inclinations of the beam at A, B, and C. These angles are denoted by θ_1 , θ_2 , and θ_3 , respectively. Making use of the elastic relations of Fig. 1-5, the terminal bending moments in each span are as indicated in Fig. 1-6.

Governing equations for the angles are now obtained by writing the conditions for moment balance at the supports A, B, and C.

$$\begin{aligned}
 M &= \frac{2EI}{a} (2\theta_1 + \theta_2) \\
 -\frac{2EI}{a} (2\theta_2 + \theta_1) + M_B - \frac{2EI}{a} (2\theta_3 + \theta_2) &= 0 \\
 -\frac{2EI}{a} (2\theta_3 + \theta_2) + M_C - \frac{EI}{a} (2\theta_3) &= 0
 \end{aligned}
 \tag{1-21}$$

These may be cast into nondimensional form by introducing the following dimensionless inclinations:

$$x_1 = \frac{\theta_1}{Ma/2EI} \quad x_2 = \frac{\theta_2}{Ma/2EI} \quad x_3 = \frac{\theta_3}{Ma/2EI} \tag{1-22}$$

We thus obtain the following formulation of the equilibrium problem:

$$\begin{aligned}
 2x_1 + x_2 &= +1 \\
 x_1 + 4x_2 + x_3 &= 0 \\
 x_2 + 3x_3 &= 0
 \end{aligned}
 \tag{1-23}$$

A complementary formulation may be obtained in terms of the bending moments M_1 , M_2 , and M_3 at B, C, and D, respectively, in the beam of Fig. 1-4. It is left as an exercise for the reader to show that in terms of the dimensionless moments

$$y_1 = \frac{M_1}{M} \quad y_2 = \frac{M_2}{M} \quad y_3 = \frac{M_3}{M} \tag{1-24}$$

the governing equations are as follows:

$$\begin{aligned}
 4y_1 + y_2 &= -1 \\
 y_1 + 6y_2 + 2y_3 &= 0 \\
 2y_2 + 4y_3 &= 0
 \end{aligned}
 \tag{1-25}$$

Problem 1-5. Hydraulic Network. We consider the problem of determining the steady flow of an incompressible fluid in a network of branched pipes under the assumption that the pressure drop in a single

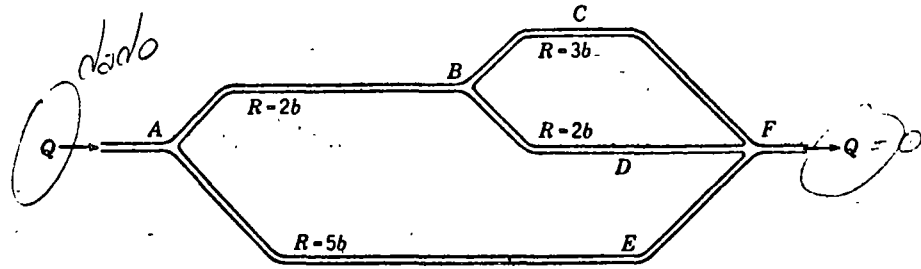


FIG. 1-7. Schematic diagram of hydraulic network passing a total flow Q.

branch is proportional to the square of the rate of flow through that branch. Figure 1-7 shows the plan of a particular pipe network. The total rate of flow, in at A and out at F, is Q. For a single branch the

pressure drop in the direction of flow is given¹ by the following resistance law,

$$\Delta p = Rq^2 \quad (1-26)$$

where q is rate of flow through the branch and R is a resistance coefficient. The resistance coefficient of each branch in Fig. 1-7 is given in terms of b .

The equilibrium problem consists in determining the pressure and flow distribution in the steady state. To make the problem definite, we assume that Q is given and that the pressure at F is zero. The governing requirements are that the pressure at each junction should be single-valued, that the rate of flow into any junction should equal the rate of flow out of that junction, and that in each separate branch the resistance law (1-26) should be satisfied. A formulation of the problem can be made in terms of either junction pressures or branch flow rates. Thus the state of the system can be represented by p_1 and p_2 , the pressures at A and B , respectively. In terms of these the flow rates in the individual branches are given by (1-26).

$$\begin{aligned} q_{AB} &= \left(\frac{p_1 - p_2}{2b} \right)^{\frac{1}{2}} \\ q_{BCF} &= \left(\frac{p_2}{3b} \right)^{\frac{1}{2}} \\ q_{BDF} &= \left(\frac{p_2}{2b} \right)^{\frac{1}{2}} \\ q_{AEF} &= \left(\frac{p_1}{5b} \right)^{\frac{1}{2}} \end{aligned} \quad (1-27)$$

The requirement of continuity of flow at the junctions A and B provides the following governing equations:

$$\begin{aligned} Q &= \left(\frac{p_1 - p_2}{2b} \right)^{\frac{1}{2}} + \left(\frac{p_1}{5b} \right)^{\frac{1}{2}} \\ \left(\frac{p_1 - p_2}{2b} \right)^{\frac{1}{2}} &= \left(\frac{p_2}{3b} \right)^{\frac{1}{2}} + \left(\frac{p_2}{2b} \right)^{\frac{1}{2}} \end{aligned} \quad (1-28)$$

A nondimensional formulation may be obtained by introducing dimensionless pressures

$$x_1 = \frac{p_1}{bQ^2} \quad x_2 = \frac{p_2}{bQ^2} \quad (1-29)$$

¹ See, for example, H. W. King, C. O. Wisler, and J. G. Woodburn, "Hydraulics," John Wiley & Sons, Inc., New York, 1948, 5th ed., p. 220. Strictly speaking we should consider Δp and q as directed quantities and write $\Delta p = [\text{sign}(q)]Rq^2$. If we use (1-26), it is incumbent on us to check that all pressure drops are actually in the direction of flow in any proposed solution.

In terms of these (1-28) may be cast into the following form:

$$\begin{aligned} 0.4472x_1^{\frac{1}{2}} + 0.7071(x_1 - x_2)^{\frac{1}{2}} &= 1 \\ 0.7071(x_1 - x_2)^{\frac{1}{2}} - 1.2845x_2^{\frac{1}{2}} &= 0 \end{aligned} \quad (1-30)$$

A complementary formulation may be obtained in terms of branch flow rates. Continuity of flow will be preserved in Fig. 1-7 if the flow rates q_1 and q_2 in the branches AB and BCF , respectively, are independent provided the flow rates in the remaining branches are taken as follows:

$$\begin{aligned} q_{BDF} &= q_1 - q_2 \\ q_{AEF} &= Q - q_1 \end{aligned} \quad (1-31)$$

With the aid of (1-26) the requirement of single-valued pressures at A and B leads to the following governing equations:

$$\begin{aligned} 2bq_1^2 + 2b(q_1 - q_2)^2 &= 5b(Q - q_1)^2 \\ 3bq_2^2 &= 2b(q_1 - q_2)^2 \end{aligned} \quad (1-32)$$

Introducing the dimensionless flow rates

$$y_1 = \frac{q_1}{Q} \quad y_2 = \frac{q_2}{Q} \quad (1-33)$$

we obtain a nondimensional formulation as follows:

$$\begin{aligned} 10y_1 - y_1^2 - 4y_1y_2 + 2y_2^2 &= 5 \\ +2y_1^2 - 4y_1y_2 - y_2^2 &= 0 \end{aligned} \quad (1-34)$$

EXERCISES

1-1. The lengths and cross-sectional areas of the bars of a plane pinned truss are indicated in Fig. 1-8. The bars are joined by frictionless pins, and each one satisfies Hooke's law, $f/A = E\delta/L$, where f is the tensile force and δ is the elongation. The

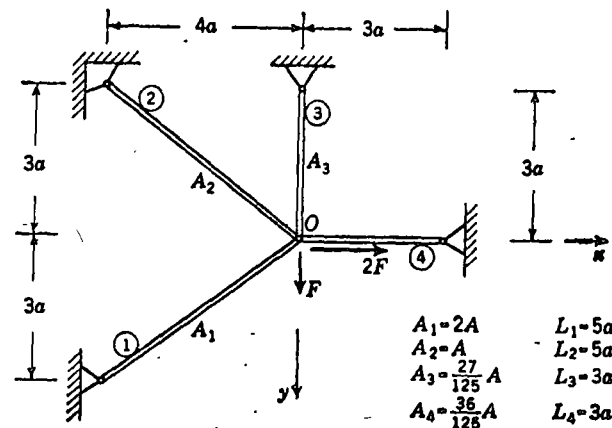


FIG. 1-8. Exercise 1-1.

EIGENVALUE PROBLEMS FOR SYSTEMS
WITH A FINITE NUMBER OF DEGREES OF FREEDOM

Equilibrium problems involve the determination of system configurations under prescribed loading conditions. An eigenvalue problem may also involve the determination of system configurations, but of greater importance is the determination of the critical loading conditions under which these configurations are possible. A parameter which describes such a critical condition is called an eigenvalue. As examples we have the natural frequencies in oscillating systems and the buckling loads in elastic-stability problems.

We consider only *linear* eigenvalue problems. Matrix notation is used because it facilitates the theoretical discussion and because it provides a useful system for laying out the actual computations. The necessary rules are briefly reviewed in Sec. 2-2.

2-1. Particular Examples

Two examples are used to illustrate the formulation of eigenvalue problems from physical systems:

- 2-1. Three-mass vibrating system.
- 2-2. Buckling of a structure.

In both cases the formulations are left in ordinary algebraic form. Matrix formulations will be given in Sec. 2-2.

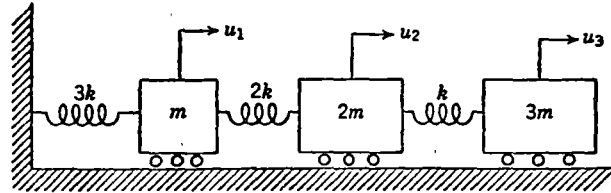


FIG. 2-1. Vibrational system with three degrees of freedom.

Problem 2-1. Three-mass Vibrating System. The system is shown in Fig. 2-1. The displacements of the three masses from the unstrained configuration are measured by u_1 , u_2 , and u_3 . The equations of motion may be written by imagining the system disturbed from equilibrium and

applying Newton's second law to each mass. Neglecting friction, we obtain

$$\begin{aligned} -3ku_1 + 2k(u_2 - u_1) &= m \frac{d^2 u_1}{dt^2} \\ -2k(u_2 - u_1) + k(u_3 - u_2) &= 2m \frac{d^2 u_2}{dt^2} \\ -k(u_3 - u_2) &= 3m \frac{d^2 u_3}{dt^2} \end{aligned} \quad (2-1)$$

For a natural vibration we would have

$$\begin{aligned} u_1 &= x_1 \sin(\omega t + \varphi) \\ u_2 &= x_2 \sin(\omega t + \varphi) \\ u_3 &= x_3 \sin(\omega t + \varphi) \end{aligned} \quad (2-2)$$

where x_1 , x_2 , and x_3 represent the amplitudes of vibration, ω is the natural frequency, and φ is a phase angle. If we substitute (2-2) into (2-1) and set

$$\frac{m\omega^2}{k} = \lambda \quad (2-3)$$

we find the following equations as the conditions for determining the amplitudes and frequency:

$$\begin{aligned} 5x_1 - 2x_2 &= \lambda(x_1) \\ -2x_1 + 3x_2 - x_3 &= \lambda(2x_2) \\ -x_2 + x_3 &= \lambda(3x_3) \end{aligned} \quad (2-4)$$

The parameter λ is a dimensionless measure of the frequency. An eigenvalue is a value of λ for which there are nonzero amplitudes which satisfy (2-4). A configuration of amplitudes which meets these requirements is called a natural mode. The corresponding frequency is called a natural frequency. A complete solution would involve finding all the natural frequencies and their associated natural modes. In technical problems it may not be of interest to obtain the complete solution. Sometimes only the lowest natural frequency is desired; sometimes just the lowest frequency and the corresponding mode or just the two lowest frequencies are desired.

Problem 2-2. Buckling of a Structure. A system of rigid weightless links hinged together and supported by springs is shown in Fig. 2-2a. In this position all three links are exactly vertical, and there is no force in any of the springs. We consider the stability of this system when subjected to a vertical load P applied at B . For small loads the three links will remain vertical, moving down as a unit against the springs k_1 . For large loads the links will buckle; that is, B and C will undergo transverse displacements as shown in Fig. 2-2b. Our problem is to determine

the stability limit for the vertical position. We want to know the value of P for which an equilibrium position with transverse displacements first becomes possible.

To obtain a quantitative analysis, we assume that the desired critical buckling load is holding the system in equilibrium and find the equi-

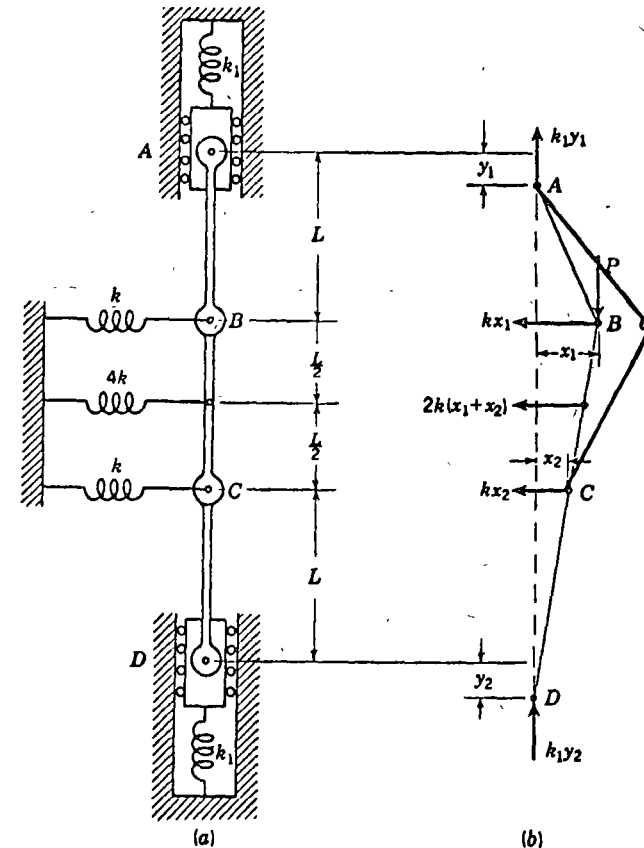


Fig. 2-2. Buckling of a system of spring-supported rigid links.

librium conditions by applying the principle of minimum potential energy. A geometrically compatible state can be represented by arbitrary (small) values of y_1 , x_1 , and x_2 if we take y_2 as

$$y_2 = y_1 - \frac{x_1^2}{2L} - \frac{(x_1 - x_2)^2}{2L} - \frac{x_2^2}{2L} \quad (2-5)$$

The usual small-angle approximations, $1 - \cos \theta \approx \frac{1}{2}\theta^2$ and $\sin \theta \approx \theta$, have been made here. By adding the strain energy of the springs to the

potential energy of the load P we have the total potential energy

$$\Phi = \frac{1}{2}k_1y_1^2 + \frac{1}{2}kx_1^2 + \frac{1}{2}k(x_1 + x_2)^2 + \frac{1}{2}kx_2^2 + \frac{1}{2}k_1y_2^2 - P\left(y_1 - \frac{x_1^2}{2L}\right) \quad (2-6)$$

where y_2 is understood to take the value (2-5). The equilibrium equations are the conditions for stationary potential energy.

$$\begin{aligned} \frac{\partial \Phi}{\partial y_1} &= k_1y_1 + k_1y_2 - P = 0 \\ \frac{\partial \Phi}{\partial x_1} &= kx_1 + k(x_1 + x_2) + k_1y_2\left(-\frac{x_1}{L} - \frac{x_1 - x_2}{L}\right) + \frac{Px_1}{L} = 0 \quad (2-7) \\ \frac{\partial \Phi}{\partial x_2} &= k(x_1 + x_2) + kx_2 + k_1y_2\left(\frac{x_1 - x_2}{L} - \frac{x_2}{L}\right) = 0 \end{aligned}$$

One solution of this system is $x_1 = x_2 = 0$ and

$$y_2 = y_1 = \frac{P}{2k_1} \quad (2-8)$$

which is obtained from (2-5) and the first of (2-7). This is the unbuckled equilibrium position.

If x_1 and x_2 do not vanish, we obtain

$$\begin{aligned} y_1 &= \frac{P}{2k_1} + \frac{1}{4L}[x_1^2 + (x_1 - x_2)^2 + x_2^2] \\ y_2 &= \frac{P}{2k_1} - \frac{1}{4L}[x_1^2 + (x_1 - x_2)^2 + x_2^2] \end{aligned} \quad (2-9)$$

by solving (2-5) and the first of (2-7). We next insert the second of (2-9) into the last two relations of (2-7) to get a pair of simultaneous equations in x_1 and x_2 . These equations contain linear and cubic terms. Since we are interested in the first appearance of buckling, we need consider only such small values of x_1 and x_2 that the cubic terms may be neglected in comparison with the linear terms. The linearized equations for x_1 and x_2 then appear as follows:

$$\begin{aligned} kx_1 + k(x_1 + x_2) + \frac{P}{2L}(-2x_1 + x_2) + \frac{Px_1}{L} &= 0 \\ k(x_1 + x_2) + kx_2 + \frac{P}{2L}(x_1 - 2x_2) &= 0 \end{aligned} \quad (2-10)$$

By introducing the dimensionless parameter

$$\lambda = \frac{2kL}{P} \quad (2-11)$$

we obtain

$$\begin{aligned} -x_2 &= \lambda(2x_1 + x_2) \\ -x_1 + 2x_2 &= \lambda(x_1 + 2x_2) \end{aligned} \quad (2-12)$$

as our formulation of the eigenvalue problem.

An eigenvalue is a value of λ for which the equations permit nonvanishing displacements. Such a configuration of displacements is called a buckling mode.

A complete solution of an eigenvalue problem involves finding all possible eigenvalues with their associated modes. In technical buckling problems a complete solution is not of interest. Very often the magnitude of the smallest buckling load is all that is required. Sometimes the corresponding buckling mode is of interest in order to assist in the design of stiffening reinforcement.

The present system has the interesting feature that if the load P is reversed (i.e., applied vertically upward) there is still the possibility of buckling. In such cases both the smallest positive and smallest negative buckling loads are of practical interest.

EXERCISES

2-1. Show that the eigenvalue problem for determining the natural frequencies and modes of torsional vibration of the system of Fig. 2-3 may be formulated as follows:

$$\begin{aligned} x_1 - x_2 &= \lambda x_1 \\ -x_1 + 2x_2 - x_3 &= \lambda x_2 \\ -x_2 + 2x_3 - x_4 &= \lambda x_3 \\ -x_3 + \frac{3}{2}x_4 - \frac{1}{2}x_5 &= \lambda x_4 \\ -\frac{1}{2}x_4 + \frac{1}{2}x_5 &= 4\lambda x_5 \end{aligned}$$

where $\lambda = \omega^2 J/k$ and k is the torsional stiffness of a shaft and J is the moment of

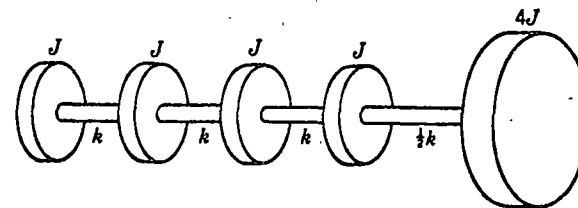


FIG. 2-3. Exercise 2-1.

inertia of a disk. The system is so supported on frictionless bearings that it is free to rotate without any bending of the shafts.

2-2. At resonance let the currents in Fig. 2-4 be

$$\begin{aligned} I_1 &= x_1 \sin(\omega t + \varphi) \\ I_2 &= x_2 \sin(\omega t + \varphi) \end{aligned}$$



centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



METODOS NUMERICOS Y APLICACIONES CON LA COMPUTADORA
DIGITAL

TEMA 4 : RAICES DE FUNCIONES TRASCEDENTES Y
POLINOMIOS.

ABRIL, 1978.

4. RAICES DE FUNCIONES TRASCENDENTES Y POLINOMIOS

4.1 Introducción

Todo modelo matemático de un sistema físico involucra el planteamiento de funciones, ya sean varias o una sola.

Existen dos tipos básicos de ecuaciones o funciones:

- trascendentes ($e^{-x} - \text{sen}3x = 0$)
- polinomiales ($x^4 - 3x^3 + 10x^2 + 1 = 0$)

Un problema frecuente es la obtención de las raíces de dichas funciones o dicho de otra manera los valores de la variable independiente que satisfacen la ecuación. Otro problema usual es el determinar la intersección de dos funciones que representan conceptos distintos (Vgr.: costos e ingresos), esto último equivale a encontrar la solución de una sola función formada por la resta de las dos originales.

Existen diversos métodos para la solución de tales ecuaciones y tendrán características particulares dependiendo del tipo de función, trascendente o polinomial, que se avoquen a resolver.

Para la solución de funciones trascendentes algunos de los métodos disponibles son: aproximaciones sucesivas, partición de intervalos, Newton-Raphson, Newton de segundo orden, Von Mises, etc.

Para la solución de funciones polinomiales: Newton-Raphson, Newton de segundo orden, Lin-Bairstow, Graeffe.

De los métodos anteriores solo se tratará el de Newton-Raphson para funciones trascendentes y el de Lin-Bairstow para funciones algebraicas.

4.2 Funciones Trascendentes

4.2.1 Objeto

Obtener la solución de funciones del tipo:

$$y = f(x) = 0 \quad (4.1)$$

por el método de Newton Raphson.

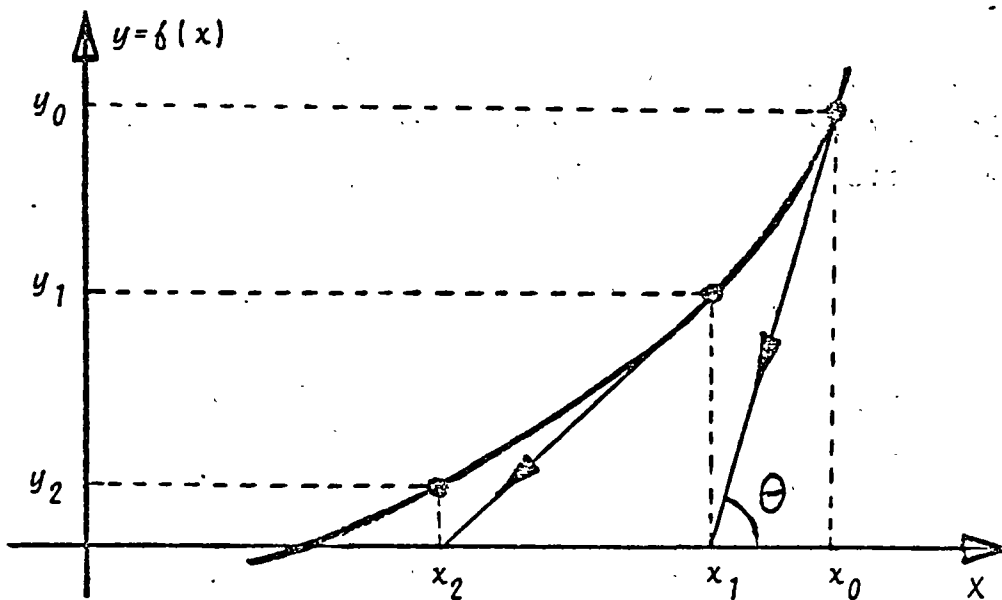
4.2.2 Método

Dada la curva correspondiente a $y=f(x)=0$ se requiere un punto inicial de arranque (x_0, y_0) , a partir de dicho punto se traza una recta tangente a la curva y la intersección de la recta con el eje "x" dará la nueva solución aproximada (x_1, y_1) ; el método se repite sucesivamente hasta que:

$$\begin{cases} |x_{n+1} - x_n| < \epsilon \\ |y_n| < \delta \end{cases} \quad (4.2)$$

donde ϵ y δ son arbitrariamente pequeñas.

Analíticamente se tendrá:



$$\left. \frac{df(x)}{dx} \right|_{x=x_0} = \operatorname{tg} \theta = \frac{f(x_0) - 0}{x_0 - x_1} \quad (4.3)$$

por lo que:

$$(x_0 - x_1) \left. \frac{d}{dx} f(x) \right|_{x=x_0} = f(x_0) \quad (4.4)$$

de donde se obtiene:

$$\begin{aligned} x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} * \\ &\vdots \\ x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} \end{aligned} \quad (4.5)$$

$$* \quad f'(x_0) = \left. \frac{df(x)}{dx} \right|_{x_0}$$

para garantizar la convergencia del método se requiere:

- x_0 esté cercano a la raíz
- $f'(x_0)$ no debe ser muy próxima a cero
- $f''(x_0)$ no debe ser excesivamente grande

El método solo permite detectar las raíces reales de la función y ésta última debe ser continua y diferenciable en una vecindad de x .

4.2.3 Descripción del programa

a) Subrutinas requeridas:

Ninguna.

b) Descripción de las variables:

G(X)	función de la cual se desean obtener sus raíces
DERG(X)	derivada de la función G
EPS	criterio de convergencia
N	máximo número de iteraciones a efectuar
L	contador de iteraciones
XV	valor viejo de la aproximación de la raíz
XN	nueva aproximación de la raíz

c) Dimensiones:

No utiliza proposición DIMENSION

d) Formatos para los datos de entrada:

SEC. TARJETAS	FORMATO	INFORMACION
1	(I5, 2F10.0)N, EPS, XV	
----- otros paquetes de datos (opcional) -----		
n		TARJETA EN BLANCO, cuando no se den nuevas condiciones de arranque.

e) Diagrama de bloques:

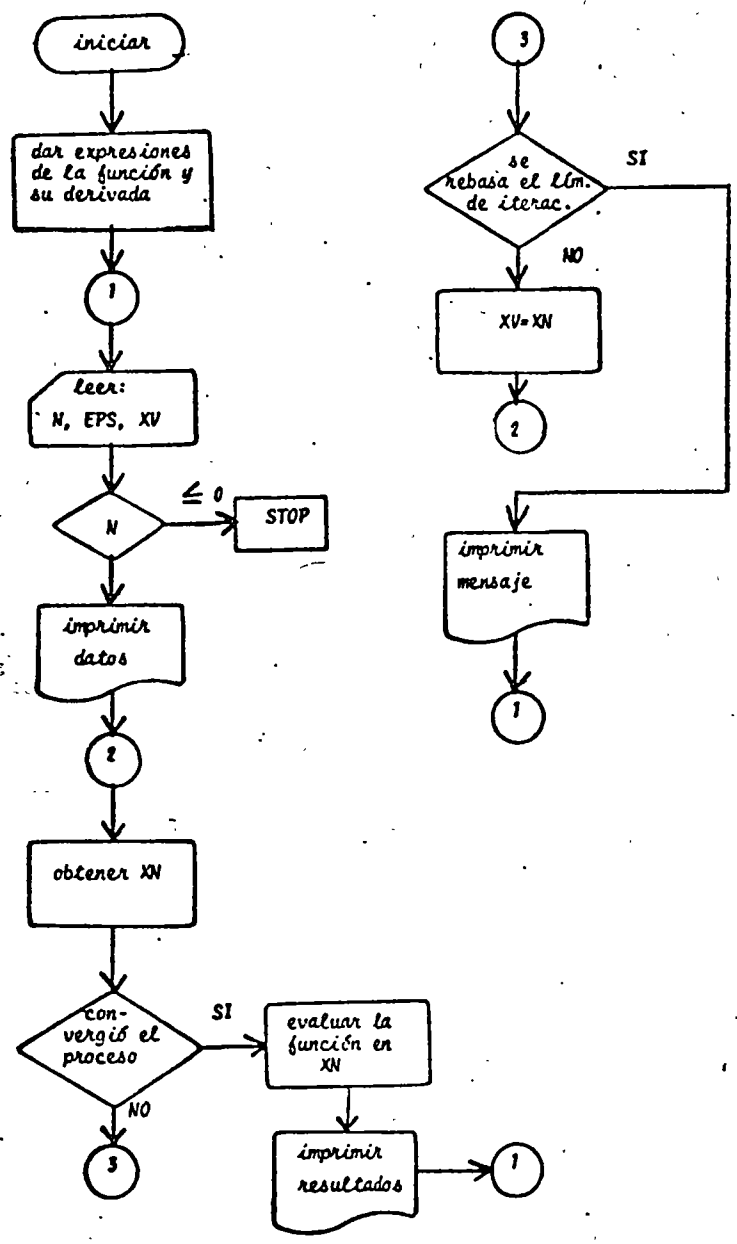


Fig. 4.1 Diagrama de bloques para el programa

6) Listado:

```

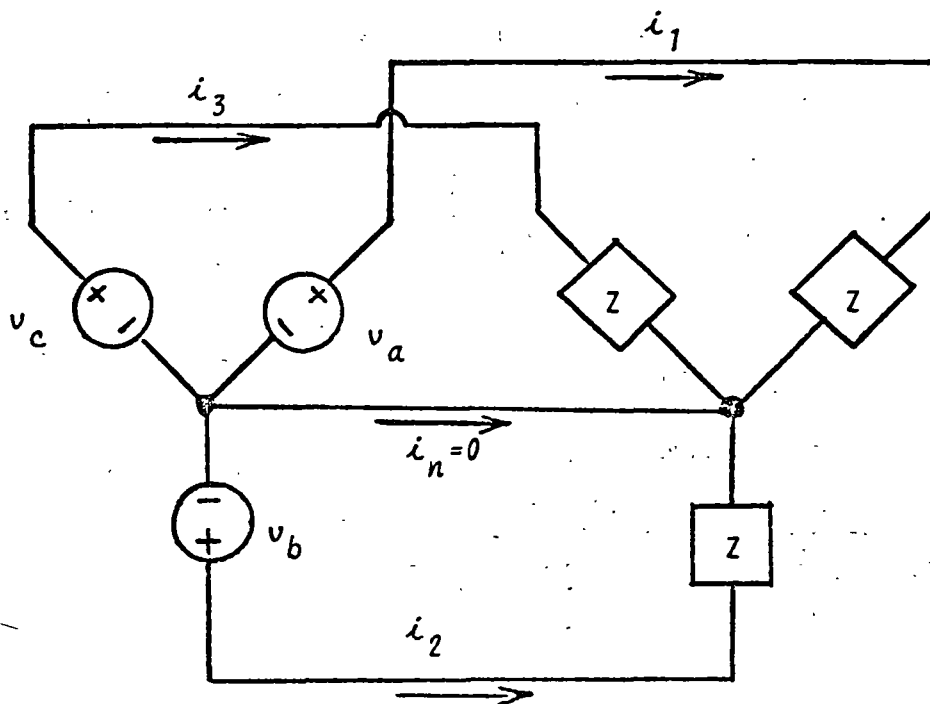
C   PROGRAMA PARA CALCULAR LAS RAICES DE UNA ECUACION POR EL METODO DE
C   NEWTON-RAPHSON
C   SIGNIFICADO DE LAS VARIABLES :
C   G= FUNCION DE LA CUAL SE OBTENDRAN LAS RAICES.
C   DERG= DERIVADA DE G.
C   EPS= CRITERIO DE CONVERGENCIA.
C   N= ITERACIONES A EFECTUAR.
C   I= CONTADOR DE ITERACIONES.
C   XV= VALOR VIEJO DE LA RAIZ , ORIGINALMENTE ES EL VALOR DE ARRANQUE
C   XN= NUEVA APROXIMACION DE LA RAIZ.
C   FUNCION G Y SU DERIVADA
-----
G(X)=SIN(377.*X - 1.88146)
DERG(X)=377.*COS(377.*X - 1.88146)
C   LECTURA DE DATOS
1   READ(5,200) N,EPS,XV
   IF(N)2,2,3
2   CALL EXIT
C   IMPRIMIR DATOS
3   WRITE(6,220) N,EPS,XV
   DO 4 I=1,N
   XN=XV-C(XV)/DERG(XV)
   IF(ABS(XV-XN).LT.EPS) GO TO 5
   XV=XN
4   CONTINUE
   XV=G(XV)
   WRITE(6,260) I,XN,XV
   GO TO 1
5   XN=G(XV)
   WRITE(6,240) I,XN,XV
   GO TO 1
C   FORMATOS DE LECTURA E IMPRESION
200  FORMAT(15,2F10.0)
220  FORMAT(14I,5(//),10X,' MAXIMO NUMERO DE ITERACIONES = ',15,3(//),10X
     1,' CRITERIO DE CONVERGENCIA = ',1PE15.8,3(//),10X,' VALOR INICIAL
     2DE LA RAIZ = ',1PE15.8)
240  FORMAT(10(//),10X,' ITERACIONES EFECTUADAS = ',15,3(//),10X,' LA RAI
     1Z OBTENIDA = ',1PE15.8,3(//),10X,' VALOR DE LA FUNCION = ',1PE15.8)
260  FORMAT(10(//),10X,' EL METODO NO CONVERGE ',3(//),10X,' NUMERO DE IT
     1ERACIONES EFECTUADAS = ',15,3(//),10X,' ULTIMA APROXIMACION DE LA RA
     2IZ = ',1PE15.8,3(//),10X,' VALOR DE LA FUNCION = ',1PE15.8)
     END

```

Fig. 4.2 Listado del programa

4.2.4 Ejemplo

Para el sistema trifásico balanceado mostrado:



Determine un instante de tiempo para el cual $i_2(t) = 0$ si $Z = 5.6 + j1.8$ y $v_a(t) = 141.4 \text{sen}(377t + 30^\circ)$. Considere secuencia de fase abc.

*SOLUCION

Por los datos se tiene: $v_b(t) = 141.4 \text{sen}(377t - 90^\circ)$ y del diagrama unifilar:

$$i_2(t) = 24 \text{sen}(377t - 107.8^\circ) \text{ A}$$

por lo que se requiere un valor de "t" tal que $\text{sen}(377t + 107.8^\circ) = 0$

TABLA 4.1 Datos del problema del ejemplo 3.2.4

$$N = 500$$

$$\text{EPS} = 0.0000001$$

$$\text{XV} = 2.5$$

$$G(X) = \text{sen}(377X - 107.8^\circ)$$

$$\text{DERG}(X) = 377 \text{cos}(377X - 107.8^\circ)$$

TABLA 4.2 Resultados del problema del ejemplo 3.2.4

MAXIMO NUMERO DE ITERACIONES =	500
CRITERIO DE CONVERGENCIA =	1.00000000E-06
VALOR INICIAL DE LA RAIZ =	2.50000000E+00
ITERACIONES EFECTUADAS =	6
LA RAIZ OBTENIDA =	2.4826544E+00
VALOR DE LA FUNCION =	4.75867426E-09

4.3 Funciones polinomiales

4.3.1 Objeto

Resolver ecuaciones del tipo:

$$P(X) = a_n X^n + a_{n-1} X^{n-1} + \dots + a_1 X + a_0 = 0 \quad (4.6)$$

mediante el método de Lin-Bairstow.

4.3.2 Método

El método de Lin-Bairstow permite obtener las raíces reales y complejas de un polinomio de grado "n" mediante su factorización en polinomios de segundo grado, dado que el teorema -- fundamental del álgebra afirma que todo polinomio se puede factorizar mediante binomios del tipo $(X - a)$ donde a es una raíz de $P(X)$, es decir:

$$P(X) = Q(X) (X - a) \quad (4.7)$$

Para todo polinomio de grado "n" se cumple que:

- tiene "n" raíces ya sea reales o complejas, únicas o repetidas
- las raíces complejas siempre aparecen en pares conjugados
- la regla de los signos de Descartes es aplicable al polinomio

A continuación se efectúa una breve descripción del método.

Dado el polinomio:

$$P(X) = X^n + A_1 X^{n-1} + \dots + A_n = 0 \quad (4.8)$$

se descompone en la siguiente forma:

$$P(X) = (X^2 + pX + q) Q(X) + RX + S \quad (4.9)$$

donde:

$$Q(X) = X^{n-2} + B_1 X^{n-3} + \dots + B_{n-3} X + B_{n-2} \quad (4.10)$$

Por un teorema del Algebra, el factor $(X^2 + pX + q)$ será raíz de $P(X)$ solo si $R = 0 = S$, la última igualdad es el objetivo a cumplir.

Agrupando términos en (4.9) e igualando términos de igual potencia en (4.9) y (4.8) se tiene:

$$A_k = B_k + pB_{k-1} + qB_{k-2}, \quad k = 1, \dots, n-2 \quad (4.11)$$

$$A_{n-1} = R + pB_{n-2} + qB_{n-3} \quad (4.12)$$

$$A_n = S + qB_{n-2} \quad (4.13)$$

$$B_0 = 1$$

$$B_{-1} = 0$$

de las ecuaciones (4.12) y (4.13) se tiene:

$$R = A_{n-1} - pB_{n-2} - qB_{n-3} = B_{n-1} \quad (4.14)$$

$$S = A_n - qB_{n-2} = B_n + pB_{n-1} \quad (4.15)$$

de las ecuaciones (4.11) se obtiene B_i en función de p y q por lo que:

$$R = f_1(p, q) = 0 \quad (4.16)$$

$$S = f_2(p, q) = 0 \quad (4.17)$$

lo cual es un sistema de ecuaciones no lineales que se resuelve por el método de Newton Raphson para Δp y Δq dando valores iniciales de p y q :

$$p_{k+1} = p_k + \Delta p \quad (4.18)$$

$$q_{k+1} = q_k + \Delta q \quad (4.19)$$

Dado un criterio de convergencia ϵ , la solución de (4.16) y (4.17) se tendrá cuando:

$$|p_{k+1} - p_k| < \epsilon \quad (4.20)$$

$$|q_{k+1} - q_k| < \epsilon \quad (4.21)$$

Al aplicar el método de Newton-Raphson a 4.16 y 4.17 se tendrá:

$$B_{n-1} + \frac{\partial B_{n-1}}{\partial p} \Delta p + \frac{\partial B_{n-1}}{\partial q} \Delta q = 0 \quad (4.22)$$

$$B_n + \left(\frac{\partial B_n}{\partial p} + B_{n-1} \right) \Delta p + \frac{\partial B_n}{\partial q} \Delta q = 0 \quad (4.23)$$

de la ecuación (4.11)

$$\frac{\partial B_k}{\partial p} = -B_{k-1} - p \frac{\partial B_{k-1}}{\partial p} - q \frac{\partial B_{k-2}}{\partial p} \quad (4.24)$$

$$\frac{\partial B_k}{\partial q} = -B_{k-2} - p \frac{\partial B_{k-1}}{\partial q} - q \frac{\partial B_{k-2}}{\partial q} \quad (4.25)$$

$$\frac{\partial B_{-1}}{\partial p} = \frac{\partial B_{-1}}{\partial q} = \frac{\partial B_0}{\partial p} = \frac{\partial B_0}{\partial q} = 0 \quad (4.26)$$

de 4.9 el polinomio $Q(X)$ puede factorizarse a su vez:

$$Q(X) = (X^2 + pX + q) (X^{n-4} + C_1 X^{n-5} + \dots + C_{n-5} X + C_{n-4}) + R^* y + S^* \quad (4.27)$$

desarrollando el mismo proceso se llega a:

$$\begin{aligned} C_k &= B_k - pC_{k-1} - qC_{k-2} \\ C_{-1} &= 0 \\ C_0 &= 1 \end{aligned} \quad (4.28)$$

comparando 4.28 con 4.24:

$$\frac{\partial B_k}{\partial p} = -C_{k-1} \quad (4.29)$$

$$\frac{\partial B_k}{\partial q} = -C_{k-2} \quad (4.30)$$

sustituyendo 4.29 y 4.30 en 4.22 y 4.23:

$$C_{n-2} \Delta p + C_{n-3} \Delta q = B_{n-1} \quad (4.31)$$

$$- \left(\frac{\partial B_n}{\partial p} + B_{n-1} \right) \Delta p + C_{n-2} \Delta q = B_n \quad (4.32)$$

$$\text{haciendo } - \left(\frac{\partial B_n}{\partial p} + B_{n-1} \right) = \overline{C_{n-1}} = C_{n-1} - B_{n-1} \quad (4.33)$$

se obtiene la expresión del sistema de ecuaciones no lineal:

$$C_{n-2} \Delta p + C_{n-3} \Delta q = B_{n-1} \quad (4.34)$$

$$\overline{C_{n-1}} \Delta p + C_{n-2} \Delta q = B_n \quad (4.35)$$

Los pasos a seguir en la computadora son:

- ① Dar valores iniciales de p y q .

- ② Evaluar B_k , $k = 1, \dots, n$
- ③ Evaluar C_k , $k = 1, \dots, n-1$
- ④ Evaluar $C_{n-1} = C_{n-1} - B_{n-1}$
- ⑤ Resolver el sistema de ecuaciones (4.34) y (4.35)
- ⑥ Obtener Δp y Δq por iteraciones hasta que $|\Delta p| < \epsilon$
y $|\Delta q| < \epsilon$
- ⑦ Obtener los valores de p y q mediante:

$$p_k = p_{k-1} + \Delta p$$

$$q_k = q_{k-1} + \Delta q$$

- ⑧ Con los valores obtenidos de p y q resolver el factor cuadrático

- ⑨ Obtener el polinomio reducido y regresar a ① hasta obtener todas las raíces.

Si los valores iniciales de p y q son cercanos a los valores verdaderos el método siempre converge.

Si los valores iniciales son aleatorios puede no haber convergencia por lo que se requiere dar un máximo número de iteraciones.

4.3.3 Descripción del Programa

a) Subrutinas requeridas:

SUBROUTINE RIR00, esta subrutina obtiene las raíces del polinomio y en caso de no existir convergencia imprime las únicas raíces encontradas. El programa principal lee los coeficientes del polinomio e imprime resultados.

b) Descripción de las variables:

Para la subrutina RIR00:

A(I)

Coeficientes del polinomio

NC

grado del polinomio

EPS

criterio de convergencia para Δp y Δq

LMAX

máximo número de iteraciones

PZERO

valor inicial de p

QZERO	valor inicial de q
RTREA(I)	parte real de las raíces
RTIMA(I)	parte imaginaria de las raíces
NUM	parámetro que indica si se encontraron o no todas las raíces
L	contador de iteraciones
J	contador de raíces encontradas
M	decrementador del grado del polinomio
B(I)	coeficientes del polinomio de grado NC-2
C(I)	coeficientes del polinomio de grado NC-4
DELP	Δp
DELQ	Δq
P	p
Q	q
RADTR	discriminante de la ecuación cuadrática
Para el programa principal:	
A(I)	coeficientes del polinomio
NC	grado del polinomio
EPS	criterio de convergencia
LMAX	máximo número de iteraciones
PZERO	valor inicial de p
QZERO	valor inicial de q
RTREA(I)	parte real de las raíces
RTIMA(I)	parte imaginaria de las raíces
NUM	parámetro que indica si se encontraron o no todas las raíces
CEREQ	criterio para determinar cuándo una --- raíz es nula

c) Dimensiones:

La proposición COMMON del programa principal y la subrutina, así como la proposición DIMENSION de la subrutina deberán modificarse si $NC > 20$

d) Formatos para los datos de entrada:

SEC.	TARJETAS	FORMATO	INFORMACION
1		(I2)	NC
2		(8F10.0)	A; el coeficiente del -- término que da el grado
.			

del polinomio debe ser unitario y no se proporciona como dato, los otros coeficientes se dan a partir del coeficiente del término de grado $NC-1$

 otros paquetes de datos (opcional)

n

TARJETA EN BLANCO, al finalizar toda la información.

e) Diagrama de bloques:

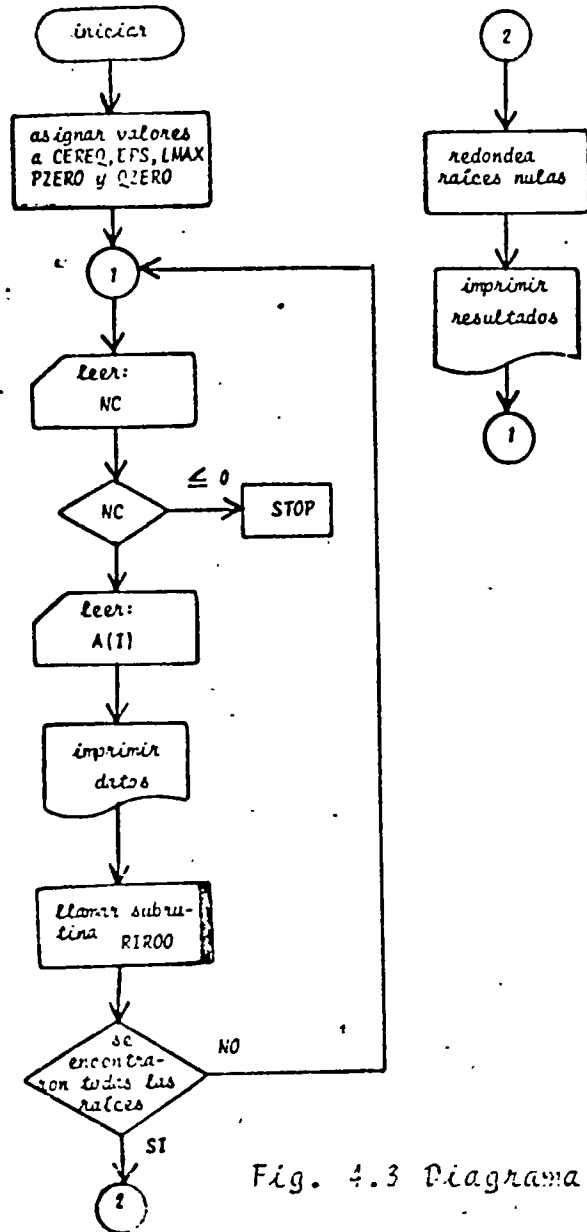


Fig. 4.3 Diagrama de bloques del programa principal

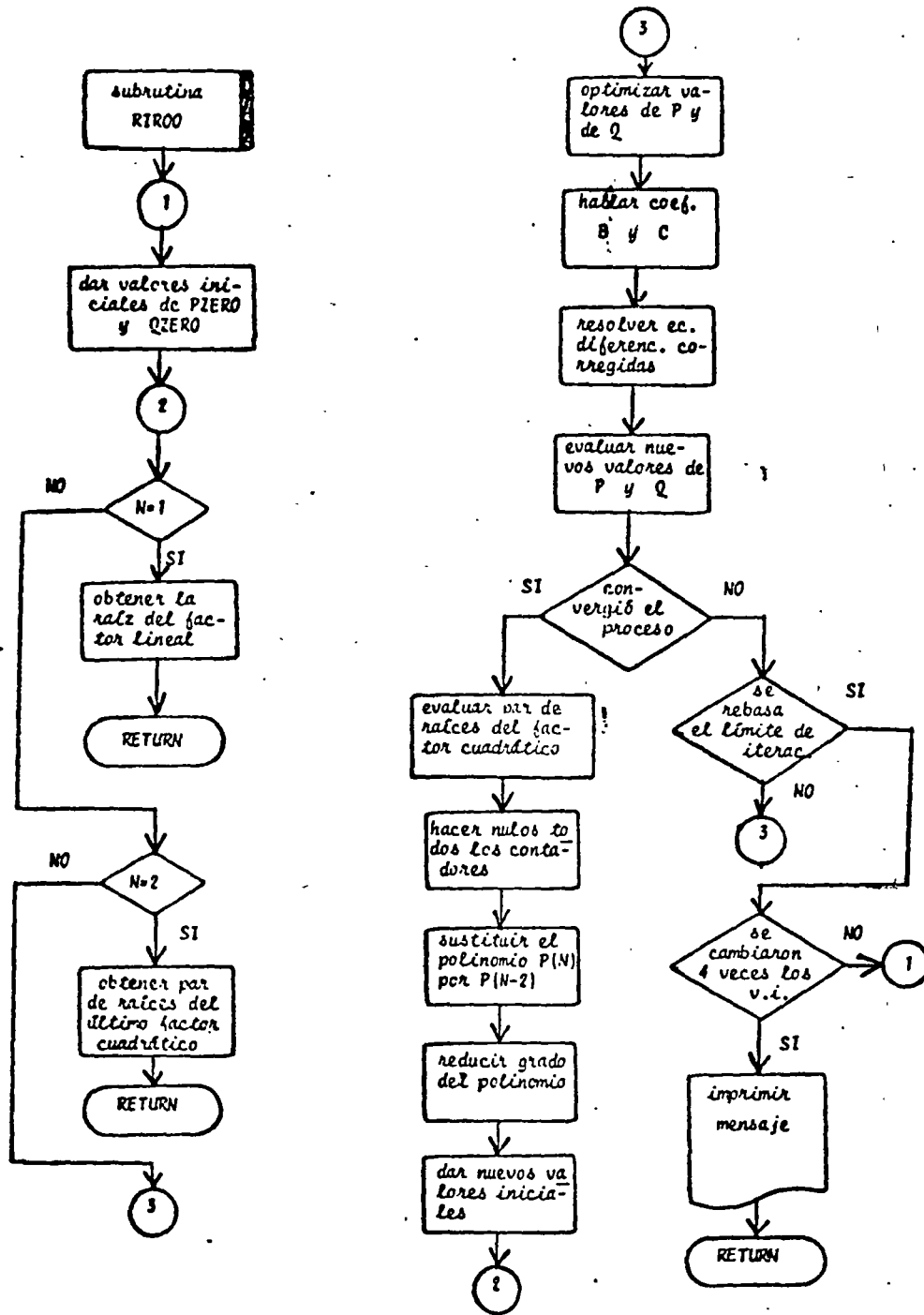


Fig. 4.4 Diagrama de bloques de la subrutina RIR00


```

C PUNTO DE PARTIDA PARA LA OPTIMIZACION DE LOS VALORES DE P Y DE Q
C DETERMINACION DE LOS COEFICIENTES B Y C
7 B(1)=A(1) = P
  B(2)=A(2) = P*C(1) = Q
  DO 3 K=3,N
8 B(K)=A(K) = P*A(K-1) = Q*B(K-2)
  N1=N-1
  C(1)=B(1) = P
  C(2)=B(2) = P*C(1) = Q
  C(3)=B(3) = P*C(2) = Q*C(1)
  IF(N,CO,3) GO TO 91
  DO 9 K=3,N1
9 C(K)=B(K) = P*C(K-1) = Q*C(K-2)
91 CIB=C(N1)-C(N1)
C SE RESUELVEN LAS ECUACIONES DIFERENCIALES CORREGIDAS
IF(N1,IE,31) GO TO 10
DENOM=(C(N-2))**2 = CIB
GO TO 11
10 DENOM=(C(N-2))**2 = CIB*C(N-3)
11 IF(DENOM.EQ.0.0) GO TO 14
IF(N1,IE,3) GO TO 12
DELP=(B(N-1)*C(N-2) - R(N))/DENOM
GO TO 13
12 DELP=(R(N-1)*C(N-2) - R(N)+C(N-3))/DENOM
13 DELQ=(R(N)-C(N-2) - R(N-1)*CIB)/DELP
C EVALUAR VALORES INCREMENTADOS DE P Y DE Q
P=P + DELP
Q=Q + DELQ
C SE PUEDE CONVERGENCIA
IF(ABS(DELP),GT,EPS) GO TO 30
IF(ABS(DELP),LE,EPS) GO TO 18
C REVISAR ITERACIONES EFECTUADAS
30 IF(L,GT,LMAX) GO TO 14
  L=L+1
  GO TO 7
14 NUM=NUM+1
  IF(N1,GT,4) GO TO 15
  GO TO (25,26,27,28),NUM
15 J1=J-1
  IF(J1,LE,0) GO TO 17
C IMPRIMIR UNICAS PAICES ENCONTRADAS
  WRITE(IH,50)
  DO 16 J=1,J1
16 WRITE(IH,200) RTREA(K),RTIMA(K)
  GO TO 24
17 WRITE(JH,150)
  GO TO 24
C OBTENER PAR DE RAICES CORRESPONDIENTES AL FACTOR CUADRATICO
18 RADTR=P+Q = 4,0*Q
  IF(RADTR) 19,20,21
19 RAD=SQRT(-RADTR)
  RTREA(J)=-P/2,Q
  RTREA(J+1)=-P/2,0
  RTIMA(J)=RAD/2,Q
  RTIMA(J+1)=RAD/2,0
  GO TO 22
20 RTREA(J)=-P/2,0
  RTREA(J+1)=-P/2,0
  RTIMA(J)=0,J
  RTIMA(J+1)=0,0
  GO TO 22
21 RAD=SQRT(PADTR)
  RTREA(J)=(-P+RAD)/2,Q
  RTREA(J+1)=(-P-RAD)/2,Q
  RTIMA(J)=0,0
  RTIMA(J+1)=0,0
C REDUCIR ORDEN DEL POLINOMIO Y CAMBIAR COEFICIENTES
22 M=M-1
  J=J+2
  NUM=0
  PZERO=0,
  QZERO=0,
  DE="=C-2="
  DO 23 K=1,M+1
23 A(K)=B(K)
  GO TO 1
C FORMATOS DE IMPRESION
50 FORMAT(//,3X, 'NO SE ENCONTRO LA SOLUCION COMPLETA, LA UNICA
150 FORMAT(//,15X, 'ENCONTRADAS DE LA ECUACION FUERON',//,41X,
200 FORMAT(//,17X, 'PARTE REAL',//,17X, 'PARTE IMAGINARIA')
150 FORMAT(//,15X, 'NO SE ENCONTRO LAS RAICES DE LA ECUACION')
200 FORMAT(//,13X,E15.7,15X,E15.7)
24 RETURN
END

```

Fig. 4.6 Listado de la subrutina R1R00

4.3.4 Ejemplo

Para un sistema lineal e invariable con el tiempo la función de transferencia $H(S)$ (relación entrada-salida en el dominio de la frecuencia) está dada por:

$$H(S) = \frac{S^3 + 6S^2 + 3S + 1}{S^5 + 8S^4 + 6S^3 + 3S^2 + 2S + 2} *$$

Se sabe que las raíces del polinomio del denominador (polos del sistema) representan las frecuencias naturales del sistema. Determine dichas frecuencias naturales.

* SOLUCION

TABLA 4.3 Datos para el problema del ejemplo 4.3.4

$$NC = 5$$

$$P(X) = X^5 + 8X^4 + 6X^3 + 3X^2 + 2X + 2$$

TABLA 4.4 Resultados para el problema 4.3.4 empleando el programa de computadora.

EL GRADO DEL POLINOMIO ES						5
LOS COEFICIENTES DEL POLINOMIO SON						
1.00	.800E+01	.600E+01	.300E+01	.200E+01	.200E+01	
LAS RAICES DEL POLINOMIO SON						
I	PARTE REAL		PARTE IMAGINARIA			
1	.26355E+00		.60717E+00			
2	.26355E+00		-.60719E+00			
3	-.65247E+00		.45425E+00			
4	-.65247E+00		-.45425E+00			
5	-.72222E+01		0.			

* $S = a + j\omega$

4.4 Bibliografía

1. CARNAHAN B., LUTHER H., WILKES J., "Applied Numerical Methods". New York: John Wiley and Sons Inc., 1969. pp. 141-209.
2. HAMMING Richard, "Numerical Methods for Scientists -- and Engineers". New York: Mc Graw Hill Book Co., -- 1962. pp. 351-359.
3. JAMES M., SMITH G., WOLFORD J., "Applied Numerical -- Methods for Digital Computation with FORTRAN". ---- Scranton Penn: International Textbook Co., 1967. pp. 127-183.
4. KUO S. Shan, "Computer Applications of Numerical ---- Methods". Reading Mass.: Addison Wesley Co., 1972. pp. 101-127, 400-404.
5. OLIVERA S. Antonio, "Apuntes de Métodos Numéricos". - México.: Facultad de Ingeniería, UNAM. 1972. pp. 3.1-3.44



centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



METODOS NUMERICOS Y APLICACIONES CON LA COMPUTADORA
DIGITAL

TEMA 5 : I N T E R P O L A C I O N .

ABRIL, 1978.

Palacio de Minería

Calle de Tacuba 5,

primer piso.

México 1, D. F.

INTERPOLATION

3.1 INTRODUCTION

Interpolation lies at the heart of classical numerical analysis. There are two main reasons for this. The first is that in hand computation there is continual need to look up the value of a function in a table. In order to find the value of the function at nontabulated arguments, it is necessary to interpolate. Moreover, the highly accurate tables at small increments of the argument that we take for granted today are mostly of comparatively recent origin. Therefore, classical numerical analysts developed an extremely sophisticated group of interpolation methods. Today the need to interpolate arises comparatively seldom; for example, on digital computers we almost always generate directly the value of a function rather than interpolate in a table of values (see Chap. 7). And when the need to interpolate in a table does arise, the small increments in the arguments in most tables mean that quite simple techniques (e.g., linear or quadratic interpolation) will usually suffice. Thus, while every numerical analyst must know how to interpolate, he will seldom, if ever, have use for the more sophisticated interpolation techniques.

Why then commence the main body of this book with a chapter on interpolation? The answer to this question is provided by the second of the reasons mentioned at the beginning of this section. This is that interpolation formulas are the starting points in the derivations of many methods in other areas of numerical analysis. Almost all the classical methods of numerical differentiation, numerical quadrature, and numerical integration are directly derivable from interpolation formulas. While

modern numerical analysis does not rely so heavily on interpolation formulas in these areas, their importance and usefulness are still great, as we shall see in Chaps. 4 and 5. Thus then is ample motivation for treating interpolation at the outset of this book.

Because we are especially interested in digital computer applications, our approach to interpolation will differ substantially from that in most texts. In particular, we shall not emphasize interpolation formulas based on difference techniques since these are seldom used on computers. Nevertheless, we shall not ignore finite differences because of their great usefulness in hand computation and, even on digital computers, for certain applications (see Secs. 4.3 and 4.15-1).

Suppose we have a function $f(x)$ which is known (perhaps along with certain of its derivatives) at a set of points. These points will hereafter be called the *tabular points* because interpolation so often takes place in a table of functional values. The object of interpolation is to *estimate* values of the function at nontabular points and—at least—to *bound* the error between the estimated and true values. Our approach will be to approximate $f(x)$ by a function $y(x)$ which, at the tabular points, has the same values as $f(x)$ (and perhaps the given derivative values, if any). Thus, in the language of the previous chapter, we shall be using exact approximations. In this chapter we shall consider only the case where $y(x)$ is a polynomial. In the last section of Ch. p. 6 we shall consider the case in which $y(x)$ is a linear combination of trigonometric functions.

In Sec. 2.2-1 the form of the general interpolation operator was given as

$$L[f(x)] = f(x) + \sum_{j=1}^n \sum_{i=0}^m A_{ji}(x) f^{(i)}(a_j) \quad (3.1-1)$$

Except in Sec. 3.8 we shall be concerned only with the case $m = 0$; that is, we shall use only the functional values of $f(x)$. When $m = 0$ we replace $A_{0j}(x)$ by $l_j(x)$ in order to conform to standard notation. Thus (3.1-1) becomes

$$L[f(x)] = f(x) - \sum_{j=1}^n l_j(x) f(a_j) \quad (3.1-2)$$

Our object is to determine the $l_j(x)$ so that

$$L[f(a_j)] = 0 \quad j = 1, \dots, n \quad (3.1-3)$$

independent of the function $f(x)$. In general, however, for nontabular points

$$L[f(x)] = E(x) \quad (3.1-4)$$

That is,

$$f(x) = \sum_{j=1}^n l_j(x) f(a_j) + E(x) = y(x) + E(x) \quad (3.1-5)$$

where $y(x)$ is our approximation to $f(x)$, and $E(x)$ is the error in the approximation. In the notation of the previous chapter, the operator $L[f(x)]$ is obtained by replacing $f(x)$ in (3.1-2) by its approximation

$$y(x) = \sum_{j=1}^n l_j(x) f(a_j) \quad (3.1-6)$$

In terms of $y(x)$ and $E(x)$ the requirement (3.1-3) becomes

$$E(a_j) = f(a_j) - y(a_j) = 0 \quad j = 1, \dots, n \quad (3.1-7)$$

Our two aims then are to determine the $l_j(x)$ so that (3.1-7) is satisfied and to find a representation for $E(x)$ which will enable us to estimate or at least bound the error for values of $x \neq a_j, j = 1, \dots, n$.

3.2 LAGRANGIAN INTERPOLATION

In this section we consider the case where there are no restrictions on the spacing of the tabular points. In Sec. 3.3 we shall then consider the case of equally spaced abscissas. Even in the general situation we consider here, however, the determination of the polynomials $l_j(x)$ is straightforward. Since we wish the error at the tabular points to be zero independent of $f(x)$, it follows using (3.1-5) that

$$l_j(a_k) = \delta_{jk} \quad j, k = 1, \dots, n \quad (3.2-1)$$

where δ_{jk} is the Kronecker delta.† Since $l_j(x)$ is to be a polynomial, this requires that it have a factor

$$(x - a_1)(x - a_2) \cdots (x - a_{j-1})(x - a_{j+1}) \cdots (x - a_n) \quad (3.2-2)$$

and since $l_j(a_j) = 1$ we may write

$$l_j(x) = \frac{(x - a_1) \cdots (x - a_{j-1})(x - a_{j+1}) \cdots (x - a_n)}{(a_j - a_1) \cdots (a_j - a_{j-1})(a_j - a_{j+1}) \cdots (a_j - a_n)} \quad (3.2-3)$$

Note that there are other possible polynomial representations of $l_j(x)$, but (3.2-3) is the only possible polynomial of degree $n - 1$ and no polynomial of lesser degree is possible (why?). It is notationally convenient to write

† $\delta_{jk} = 0$ unless $j = k$, in which case $\delta_{jk} = 1$.

$l_j(x)$ as

$$l_j(x) = \frac{p_n(x)}{(x - a_j)p'_n(a_j)} \quad p'_n(a_j) = \left. \frac{dp_n}{dx} \right|_{x=a_j} \quad (3.2-4)$$

where

$$p_n(x) = \prod_{i=1}^n (x - a_i) \quad (3.2-5)$$

To find an expression for $E(x)$, we consider the function

$$F(z) = f(z) - y(z) - [f(x) - y(x)][p_n(z)/p_n(x)] \quad (3.2-6)$$

with $y(x)$ as in (3.1-6). The function $F(z)$ as a function of z has $n + 1$ zeros at the points a_1, \dots, a_n and x [assume for now that x in (3.2-6) is not one of the tabular points]. Therefore, by applying Rolle's theorem n times

$$F^{(n)}(z) = f^{(n)}(z) - y^{(n)}(z) - [f(x) - y(x)][n!/p_n(x)] \quad (3.2-7)$$

has at least one zero in the interval spanned by a_1, \dots, a_n and x . Calling this zero $z = \xi$ and noting that $y^{(n)}(z) = 0$ since $l_j(z)$ is a polynomial of degree $n - 1$, we have

$$0 = F^{(n)}(\xi) = f^{(n)}(\xi) - [f(x) - y(x)][n!/p_n(x)] \quad (3.2-8)$$

from which, using (3.1-5), it follows that

$$E(x) = [p_n(x)/n!]f^{(n)}(\xi) \quad (3.2-9)$$

where ξ , which is an unknown function of x , lies in the interval spanned by a_1, \dots, a_n and x . Although x in (3.2-6) was restricted to be a non-tabular point, $E(x)$, as given by (3.2-9), holds for both tabular and non-tabular points (why?).

Equation (3.1-5) with the $l_j(x)$ given by (3.2-4) and $E(x)$ by (3.2-9) is called the *Lagrangian interpolation formula*. When $n = 2$, $y(x)$ is the familiar formula for linear interpolation [cf. (2.2-4)]

$$y(x) = \frac{x - a_2}{a_1 - a_2} f(a_1) + \frac{x - a_1}{a_2 - a_1} f(a_2) \quad (3.2-10)$$

The polynomials $l_j(x)$ are called *Lagrangian interpolation polynomials*. Our derivation of the Lagrangian formula has been equivalent to finding that polynomial of degree $n - 1$ which passes through the points $[a_j, f(a_j)]$, $j = 1, \dots, n$ {4}. Therefore, as we would expect, (3.2-9) indicates that this formula is *exact* [i.e., $E(x) = 0$ for all x] for polynomials of degree $n - 1$ or less. In general, an interpolation formula which is

exact for polynomials of degree r is said to have an order of accuracy r or to be of order r .

The use of the Lagrangian interpolation formula is straightforward. To estimate $f(x)$ at a nontabular point, we merely compute $y(x)$ as given by (3.1-6) using (3.2-1) and (3.2-5) to compute the polynomials $l_j(x)$. If we can estimate or bound the n th derivative of $f(x)$, then the error can be estimated or bounded using (3.2-9).

Example 3.1 Let $f(x) = \ln x$. Given the table of values

x :	.40	.50	.70	.80
$\ln x$:	-.916291	-.693147	-.356675	-.223144

estimate the value of $\ln .60$.

With $a_1 = .40$, $a_2 = .50$, $a_3 = .70$, and $a_4 = .80$, we calculate from (3.2-4)

$$l_1(.60) = \frac{-1}{4!} \quad l_2(.60) = \frac{2}{4!}$$

$$l_3(.60) = \frac{2}{3!} \quad l_4(.60) = \frac{-1}{4!}$$

and from (3.1-6) we get the approximation

$$\ln .60 \approx -.509975$$

The true value is $\ln .60 = -.510826$. From (3.2-9) we get

$$E(.60) = \frac{p_4(.60)}{4!} \left(\frac{-6}{\xi^4} \right) = \frac{-.0004}{4} \frac{1}{\xi^4}$$

In the interval $(.4, .8)$, $\frac{10^4}{4096} < \frac{1}{\xi^4} < \frac{10^4}{256}$ so that

$$\frac{1}{4096} < |E(.60)| < \frac{1}{256}$$

and indeed the difference between the approximate and true values lies within this error.

3.3 INTERPOLATION AT EQUAL INTERVALS

In most applications of interpolation, the tabular points are equally spaced. For this reason it is worthwhile to consider the simplifications of the Lagrangian formula that can be made in this case.

3.3-1 Lagrangian interpolation at equal intervals

Let the equal spacing be h so that

$$a_{j+1} - a_j = h \quad j = 1, \dots, n-1 \quad (3.3-1)$$

For reasons of symmetry and computational convenience, it is common to take n odd and let

$$x = a_r + hm \quad (3.3-2)$$

where $r = (n+1)/2$. Thus $m = 0$ corresponds to the center of the

Table 3.1 Values of the Lagrangian interpolation polynomials for $n = 5$ ($x = a_r + hm$)

m	$l_1(m)$	$l_2(m)$	$l_3(m)$	$l_4(m)$	$l_5(m)$	
0	.0000	.0000	1.0000	.0000	.0000	0
.2	.0144	-.1056	.9504	.1584	-.0176	-.2
.4	.0224	-.1536	.8064	.3584	-.0336	-.4
.6	.0224	-.1456	.5824	.5824	-.0416	-.6
.8	.0144	-.0896	.3024	.8064	-.0336	-.8
1.0	.0000	.0000	.0000	1.0000	.0000	-1.0
1.2	-.0176	.1024	-.2816	1.1264	.0704	-1.2
1.4	-.0336	.1904	-.4896	1.1424	.1904	-1.4
1.6	-.0416	.2304	-.5616	.9984	.3711	-1.6
1.8	-.0336	.1824	-.4256	.6384	.6384	-1.8
2.0	.0000	.0000	.0000	.0000	1.0000	-2.0
	$l_1(m)$	$l_2(m)$	$l_3(m)$	$l_4(m)$	$l_5(m)$	m

interval spanned by the tabular points. Using (3.3-2), $p_n(x)$ and $l_j(x)$ can be expressed as functions of m . In particular, from (3.2-3) it follows that $l_j(m)$ is independent of h and can thus be tabulated as a function of m . Using (3.3-2) the Lagrangian interpolation formula becomes, writing $f(a_r + hm)$ as $f(m)$,

$$f(m) = \sum_{j=1}^n l_j(m) f(a_j) + [h^n p_n(m)/n!] f^{(n)}(\xi) \quad (3.3-3)$$

where

$$p_n(m) = (m-r+1)(m-r+2) \cdots m(m+1) \cdots (m+r-1) \quad (3.3-4)$$

Table 3.1 is a short tabulation of the Lagrangian interpolation polynomials $l_j(m)$ for $n = 5$. Clearly, when m and n are such that the $l_j(m)$ are tabulated, the use of (3.3-3) is quite straightforward on a desk calculator. On a digital computer, it will seldom be convenient to store such a table but rather will be easier to generate the values of $l_j(m)$ using (3.2-4).

Example 3.2 Using the same data as in Example 3.1 plus the true value of $\ln .60$, estimate the value of $\ln .54$.

We have $h = .1$; using Table 3.1 with $m = -.6$, we get from (3.3-3)

$$\ln .54 \approx -.0416 \ln .40 + .5824 \ln .50 + .5824 \ln .60 - .1456 \ln .70 + .0224 \ln .80$$

$$= -.616143$$

whereas the true value is $-.616186$.

When the values of $l_j(m)$ are not tabulated, then, for hand computa-

tion, instead of (3.3-3) it is preferable to use the finite-difference interpolation formulas which we shall discuss in Sec. 3.4. Before proceeding to discuss finite differences, however, we emphasize that *there is one and only one polynomial of degree $n - 1$ that takes on the values of $f(x)$ at the n tabular points* (why?). In what follows, we shall write interpolation formulas in a form very different from (3.1-5) or (3.3-3). But as long as these formulas involve polynomials passing through the same n tabular points, they will be identical to the Lagrangian interpolation formula.

3.3-2 Finite differences

In textbooks on classical numerical analysis, the calculus of finite differences and the interpolation, differentiation, and integration formulas based on it were always of central importance. This is because, for work on desk calculators, finite differences are a wonderfully convenient tool. Aside from their advantages for hand computation, there are certain special applications for which finite differences are invaluable (see Sec. 3.3-2-3). Also they are used extensively—although generally in a quite simple form—in the numerical solution of partial differential equations and boundary-value problems of ordinary differential equations on digital computers (see Sec. 4.3).

3.3-2-1 Definitions

As in Sec. 3.3-1 let the interval between successive tabular points be h . Then we define:

1. The k th forward difference of $f(x)$ as

$$\begin{aligned} \Delta^k f(x) &= \Delta^{k-1} f(x+h) - \Delta^{k-1} f(x) \quad k = 1, 2, \dots \\ \Delta^0 f(x) &= f(x) \end{aligned} \tag{3.3-5}$$

Thus, for example,

$$\begin{aligned} \Delta^1 f(x) &\equiv \Delta f(x) = f(x+h) - f(x) \tag{3.3-6} \\ \Delta^2 f(x) &= \Delta f(x+h) - \Delta f(x) = f(x+2h) - 2f(x+h) + f(x) \tag{3.3-7} \end{aligned}$$

In fact, it should be clear from this definition that any order difference can be written as a linear combination of functional values as in (3.3-6) and (3.3-7). The general form of this linear combination, whose derivation we leave to a problem [8], is

$$\Delta^j f(x) = \sum_{k=0}^j (-1)^{j-k} \binom{j}{k} f(x+kh) \tag{3.3-8}$$

where the binomial coefficient $\binom{j}{k} = \frac{j!}{k!(j-k)!}$

2. The k th backward difference as

$$\begin{aligned} \nabla^k f(x) &= \nabla^{k-1} f(x) - \nabla^{k-1} f(x-h) \quad k = 1, 2, \dots \\ \nabla^0 f(x) &= f(x) \end{aligned} \tag{3.3-9}$$

3. The k th central difference as

$$\begin{aligned} \delta^k f(x) &= \delta^{k-1} f(x + \frac{1}{2}h) - \delta^{k-1} f(x - \frac{1}{2}h) \\ \delta^0 f(x) &= f(x) \end{aligned} \quad k = 1, 2, \dots \tag{3.3-10}$$

Note that if x is a tabular point then only even central differences involve tabular points (why?).

A property of differences that we shall have use for later is that the first difference of a polynomial of degree n is a polynomial of degree $n - 1$ [9]. Therefore, the n th difference of a polynomial of degree n is a constant, and the $(n + 1)$ st difference is identically zero. The properties of finite differences and the formulas based upon them may be derived by operational calculus using the difference operators Δ , ∇ , and δ ; we leave a consideration of this approach to a problem [10].

3.3-2-2 The lozenge diagram

In the remainder of this section, we shall denote $\Delta^j f(a_k)$ by $\Delta^j f_k$ with a corresponding notation for backward and central differences. Furthermore, we shall change our previous notation slightly and let the tabular points have both positive and negative subscripts. When we calculate differences, it is convenient to set up a difference table as in Fig. 3.1 in

a_{-4}	f_{-4}		$\Delta^2 f_{-4}$	$\Delta^3 f_{-4}$	$\Delta^4 f_{-4}$	
		Δf_{-4}				$\Delta^5 f_{-4}$
a_{-3}	f_{-3}	Δf_{-3}	$\Delta^2 f_{-3}$	$\Delta^3 f_{-3}$	$\Delta^4 f_{-3}$	$\Delta^5 f_{-3}$
			Δf_{-2}	$\Delta^2 f_{-2}$	$\Delta^3 f_{-2}$	$\Delta^4 f_{-2}$
a_{-2}	f_{-2}	Δf_{-2}	$\Delta^2 f_{-2}$	$\Delta^3 f_{-2}$	$\Delta^4 f_{-2}$	$\Delta^5 f_{-2}$
			Δf_{-1}	$\Delta^2 f_{-1}$	$\Delta^3 f_{-1}$	$\Delta^4 f_{-1}$
a_{-1}	f_{-1}	Δf_{-1}	$\Delta^2 f_{-1}$	$\Delta^3 f_{-1}$	$\Delta^4 f_{-1}$	$\Delta^5 f_{-1}$
			Δf_0	$\Delta^2 f_0$	$\Delta^3 f_0$	$\Delta^4 f_0$
a_0	f_0	Δf_0	$\Delta^2 f_0$	$\Delta^3 f_0$	$\Delta^4 f_0$	$\Delta^5 f_0$
			Δf_1	$\Delta^2 f_1$	$\Delta^3 f_1$	$\Delta^4 f_1$
a_1	f_1	Δf_1	$\Delta^2 f_1$	$\Delta^3 f_1$	$\Delta^4 f_1$	$\Delta^5 f_1$
			Δf_2	$\Delta^2 f_2$	$\Delta^3 f_2$	$\Delta^4 f_2$
a_2	f_2	Δf_2	$\Delta^2 f_2$	$\Delta^3 f_2$	$\Delta^4 f_2$	$\Delta^5 f_2$
			Δf_3	$\Delta^2 f_3$	$\Delta^3 f_3$	$\Delta^4 f_3$
a_3	f_3	Δf_3	$\Delta^2 f_3$	$\Delta^3 f_3$	$\Delta^4 f_3$	$\Delta^5 f_3$
			Δf_4	$\Delta^2 f_4$	$\Delta^3 f_4$	$\Delta^4 f_4$
a_4	f_4	Δf_4	$\Delta^2 f_4$	$\Delta^3 f_4$	$\Delta^4 f_4$	$\Delta^5 f_4$

Fig. 3.1 Forward difference table.

BIBLIOTECA DE ING. I
 DEPARTAMENTO DE ENGENHARIA
 SISTEMA DE BIBLIOTECA

which each entry after the second column is the difference of the two immediately to its left. The use of forward differences in the table is arbitrary; backward differences could just as easily have been used (but not central differences—why?).

Example 3.3 Using the data of Example 3.2 with one point added at either end, compute the difference table.

The result is

x	$\ln x$	Δ	Δ^2	Δ^3	Δ^4	Δ^5	Δ^6
.30	-1.203973	.287682					
.40	-.916291	.223144	-.064538				
.50	-.693147	.182321	-.040823	.023715			
.60	-.510826	.154151	-.028170	.012653	-.005959		
.70	-.356675	.133531	-.020620	.007550	-.002425		
.80	-.223144	.117783	-.015748	.004872			
.90	-.105361						

If to Fig. 3.1 we add connecting lines and binomial coefficients† as in Fig. 3.2, we can use this modified difference table called a *lozenge* or *Fraser diagram* to generate most of the interesting finite-difference interpolation formulas. To generate such an interpolation formula, we proceed as follows:

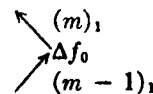
1. Start at an entry in the first (functional value) column and proceed along *any* path in the lozenge diagram (i.e., if a segment terminates on a difference, the path may be continued along any of the other three paths leading from the difference). End the path at any difference.
2. Then construct the formula by
 - (i) writing down the functional value at which the path started and then
 - (ii) for every left to right segment in the path *add* a term consisting of the difference on which the segment *terminates* multiplied by the binomial coefficient directly *below* this difference, if the slope of the segment is positive, and directly *above*, if the slope of the segment is negative, and

$$\dagger (m+k)_n = \frac{(m+k)(m+k-1)\cdots(m+k-n+1)}{n!} = \binom{m+k}{n}. \text{ In}$$

this section we let m be such that $x = a_0 + hm$ [cf. (3.3-2)].

- (iib) for every right to left segment subtract a term consisting of the difference at which the segment *originates* multiplied by the binomial coefficient directly *below* this difference, if the slope of the segment is positive (i.e., if the segment goes downward and to the left), and directly *above*, if the slope is negative.

These rules imply that, if at a given difference we change direction from right to left to left to right, this difference does not appear in the interpolation formula. As an example of the opposite situation, the path



gives rise to the terms

$$(m-1)_1 \Delta f_0 - (m)_1 \Delta f_0$$

For example, starting at f_0 , proceeding along lines sloping downward to the right and terminating with the n th difference, we get, writing

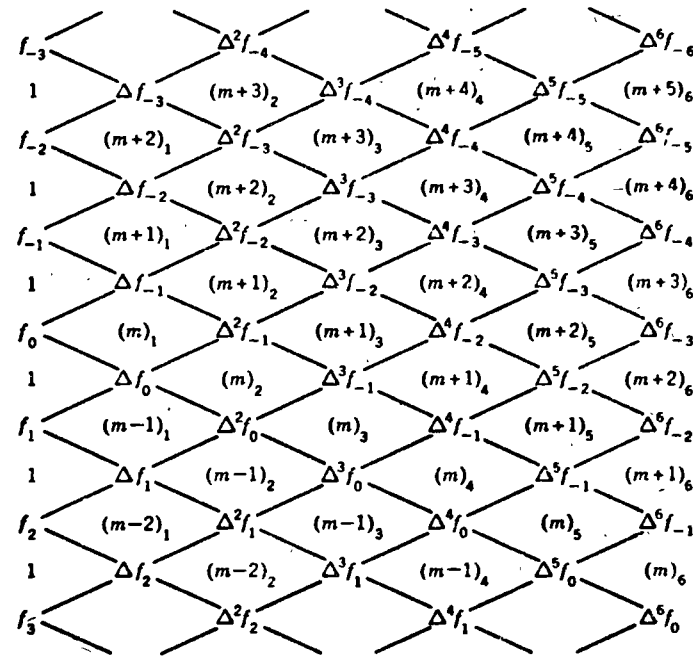


Fig. 3.2 The lozenge diagram.

$y(a_0 + hm)$ as $y(m)$,

$$y(m) = f_0 + (m)_1 \Delta f_0 + (m)_2 \Delta^2 f_0 + \cdots + (m)_n \Delta^n f_0 = \sum_{j=0}^n (m)_j \Delta^j f_0 \quad (3.3-11)$$

This formula is called *Newton's forward formula* and will be discussed in more detail in Sec. 3.4.

The value of the procedure outlined above is contained in the statement that any formula derived by this procedure which terminates with an n th difference is *algebraically equivalent* to an equal-interval Lagrangian formula which uses the tabular points involved in the terminating difference. [For example, the n th difference in (3.3-11) involves the points a_0, \dots, a_n ; see (3.3-8).] The proof of this assertion requires that we show that

1. At least one formula has this property. In Sec. 3.4 we shall prove that Newton's forward formula has the desired property.
2. All formulas which terminate with the same difference no matter by what path they reach that difference are algebraically equivalent. We leave the proof of this to a problem {11}.

3.3-2-3 Error propagation in difference tables. Table checking

Suppose one of the entries in the second column of Fig. 3.1 is in error by ϵ . We ask the question: How will this error propagate through the difference table? To answer this question, it is sufficient to consider the auxiliary table shown in Fig. 3.3 in which all functional values are zero except for a value of ϵ corresponding to that functional value in error (why is this

f	Δ	Δ^2	Δ^3	Δ^4	Δ^5
0		0		0	
0	0		0		ϵ
0	0	0		ϵ	
0	ϵ		ϵ		-5ϵ
ϵ		-2ϵ		6ϵ	
0	$-e$		3ϵ		-10ϵ
0	0	ϵ		-4ϵ	
0	0	0	$-\epsilon$		5ϵ
0	0	0	0	ϵ	
0	0	0	0	0	$-\epsilon$

Fig. 3.3 Error propagation in difference tables.

sufficient?) Note the binomial-coefficient pattern in each column. This error-propagation pattern is the basis of the method to be described below for using differences to check the correctness of entries in a table.

When a table of a mathematical function is compiled, it is clearly very important that every entry be correct (i.e., be correctly rounded). The rationale behind the method we are about to present is, first, that tabulated mathematical functions are locally smooth and, second, that generally some quite low order difference will be nearly zero. The latter is equivalent to saying that the coefficients of the Taylor-series expansion of the function are all small except for the first few (why?). We proceed as follows:

1. Difference the table. If, at any stage of the differencing, one group of values disrupts a smooth pattern (e.g., monotonicity), this probably indicates an error. Continue differencing until such a pattern appears or all differences are nearly zero (see remarks below on roundoff).
2. If a disrupting pattern is found and the deviation from smoothness follows the binomial-coefficient pattern of Fig. 3.3, then a single error has been detected and is easily corrected.
3. If the pattern is not binomial, then it may be that two or more errors are present and the patterns have overlapped. In this case some ingenuity is required to untangle the patterns; see {12}.

It is important to note that roundoff errors propagating through a table can also disrupt a difference pattern. The worst case of such propagation is shown in Fig. 3.4. Here ϵ is the magnitude of the maximum

f	Δ	Δ^2	Δ^3	Δ^4	Δ^5
ϵ		-4ϵ		16ϵ	
$-\epsilon$	-2ϵ		8ϵ		-32ϵ
ϵ	2ϵ	4ϵ		-16ϵ	
ϵ		-4ϵ		16ϵ	
$-\epsilon$	-2ϵ		8ϵ		-32ϵ
ϵ	2ϵ	4ϵ		-16ϵ	
ϵ		-4ϵ		16ϵ	
$-\epsilon$	-2ϵ		8ϵ		-32ϵ
ϵ	2ϵ	4ϵ		-16ϵ	
ϵ		-4ϵ		16ϵ	

Fig. 3.4 Propagation of roundoff error.

roundoff error in the functional values. Then in the n th difference, the worst possible error is $2^n \epsilon$. In checking tables by differencing, it is important, therefore, to distinguish between irregularities in the difference pattern due to errors and those due to roundoff [13].

Example 3.4 Find the error in the difference table

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$
1	1			
2	4	3		
3	9	5	2	0
4	16	7	2	-1
5	24	8	1	3
6	36	12	4	-3
7	49	13	1	1
8	64	15	2	0
9	81	17	2	

A disrupting pattern is noticeable in the second difference and the binomial pattern stands out clearly in the third difference. Thus there is an error $\epsilon = -1$ in the entry for $x = 5$ which should, of course, be 25 since $f(x) = x^2$ for all the other entries.

3.4 FINITE-DIFFERENCE INTERPOLATION FORMULAS

We prove first that Eq. (3.3-11), Newton's forward formula, is algebraically equivalent to the Lagrangian interpolation formula at equal intervals for the $n + 1$ points a_0, \dots, a_n . Since $(m)_n$ is a polynomial of degree n in m , it is sufficient to prove that $y(i)$ in (3.3-11) equals f_i , $i = 0, \dots, n$ for then $y(m)$ would be the unique polynomial of degree n passing through the $n + 1$ points f_i . Using (3.3-8) in (3.3-11), we get

$$y(i) = \sum_{j=0}^n (i)_j \Delta^j f_0 = \sum_{j=0}^n \sum_{k=0}^j (-1)^{j-k} (i)_j \binom{j}{k} f_k$$

$$= \sum_{k=0}^n \sum_{j=k}^n (-1)^{j-k} (i)_j \binom{j}{k} f_k \quad i = 0, \dots, n \quad (3.4-1)$$

The coefficient of f_i in $y(i)$ is then given by

$$\sum_{j=r}^n (-1)^{j-r} (i)_j \binom{j}{r} \quad (3.4-2)$$

For $r > i$ this coefficient is zero since $(i)_j = 0$ if $i < j$. When $r = i$, the only nonzero term in (3.4-2) is that for $j = r$ and equals 1. When $r < i$, (3.4-2) may be written

$$\sum_{j=r}^i (-1)^{j-r} \binom{i}{j} \binom{j}{r} \quad (3.4-3)$$

which, by suitable manipulation [15], can be shown to vanish. Thus, the right-hand side of (3.4-1) is just f_i , which completes the proof.

Using the lozenge diagram, we may generate the following interpolation formulas:

1. *Newton's backward formula.* Starting at f_0 and proceeding along lines sloping upward and to the right, we get

$$y(m) = f_0 + m \Delta f_{-1} + (m+1)_2 \Delta^2 f_{-2} + \dots + (m+n-1)_n \Delta^n f_{-n} \quad (3.4-4)$$

which is equivalent to a Lagrangian formula using the points $a_0, a_{-1}, \dots, a_{-n}$. This formula is, in fact, more conveniently expressed in terms of backward differences [16].

2. *Gauss's forward formula.* Here we proceed in a zigzag, downward and to the right, then upward and to the right, then downward and to the right, etc. The result is

$$y(m) = f_0 + m \Delta f_0 + (m)_2 \Delta^2 f_{-1} + (m+1)_3 \Delta^3 f_{-1} + (m+1)_4 \Delta^4 f_{-2} + \dots \quad (3.4-5)$$

If terminated with a difference of order $2r$, (3.4-5) is equivalent to a Lagrangian formula using the points $a_0, a_{\pm 1}, \dots, a_{\pm r}$, but if terminated with the difference of order $2r + 1$, the point a_{r+1} must be added to the above.

3. *Gauss's backward formula.* Here we proceed as in Gauss's forward formula except that the first step is upward and to the right. The formula is

$$y(m) = f_0 + m \Delta f_{-1} + (m+1)_2 \Delta^2 f_{-1} + (m+1)_3 \Delta^3 f_{-2} + (m+2)_4 \Delta^4 f_{-2} \dots \quad (3.4-6)$$

Both Gaussian formulas are conveniently expressed in terms of central differences [16].

Because of the result stated in Sec. 3.3-2-2, each of these formulas is

algebraically equivalent to the Lagrangian formula which uses the same tabular points. The errors in these formulas are given, therefore, by (3.2-9). In the next section we shall indicate why it is useful to be able to express the same interpolation formula in a number of different forms.

If we take the mean of Gauss's forward and backward formulas as given by (3.4-5) and (3.4-6), we get *Stirling's interpolation formula*

$$y(m) = f_0 + (m/2)(\Delta f_0 + \Delta f_{-1}) + \frac{1}{2}[(m)_2 \Delta^2 f_{-1} + (m+1)_2 \Delta^2 f_{-2}] + \dots \quad (3.4-7)$$

Terminated with a difference of order $2r$, this formula is also equivalent to a Lagrangian formula since both Gaussian formulas use the points $a_0, a_{\pm 1}, \dots, a_{\pm r}$, but, if terminated with an odd difference (say, $2r+1$), this is no longer true because one Gaussian formula uses the above points plus a_{r+1} and the other uses a_{-r-1} . In the latter case Stirling's formula uses $2r+3$ points but has an accuracy of only order $2r+1$. Stirling's formula can be conveniently expressed in terms of central differences [16].

Bessel's interpolation formula is the mean of the Gaussian forward formula given by (3.4-5) and a Gaussian backward formula launched not from f_0 but from f_1 . It has the form

$$y(m) = \frac{1}{2}(f_0 + f_1) + (m - \frac{1}{2})\Delta f_0 + \frac{1}{2}(m)_2(\Delta^2 f_0 + \Delta^2 f_1) + \dots \quad (3.4-8)$$

Note that, when (3.4-6) is modified to consider launching from f_1 , m must be replaced by $m-1$ so that the origin of m is still at a_0 . Analogously to the case with Stirling's formula, Bessel's formula terminated with an odd difference is equivalent to a Lagrangian formula but terminated with an even difference it is not (why?). Bessel's formula is also conveniently written using central differences [16].

Some other interpolation formulas which can be obtained by manipulating the ones derived in this section are considered in a problem [17].

3.5 THE USE OF INTERPOLATION FORMULAS

With the exception of Stirling's formula terminated with an odd difference and Bessel's formula terminated with an even difference, all the interpolation formulas we have derived are algebraically equivalent over the same set of tabular points. For equally spaced data, the ease with which difference tables can be generated makes the finite-difference interpolation formulas more convenient than the Lagrangian formula for hand computation. To get some insight into which of the finite-difference formulas to use in a given application (why does it matter if they are all equivalent?), let us consider interpolation in a table of values.

One of the great advantages of the finite-difference interpolation formulas is the ease with which added terms of the formula may be used merely by calculating higher differences in the table of Fig. 3.1. For example, if we add the value of $\ln x$ at 1.0 to the table of Example 3.3, we may calculate a new row of differences and thereby get a difference of order 7 in the table. Commonly, we do not know a priori how many terms in a given interpolation formula will be sufficient to achieve the accuracy we desire. Therefore, we generally add terms to the formula by computing higher differences until the contribution of the added terms is so small that the number of decimal places of interest to us has stabilized. [If by use of (3.2-9) we can bound the error all well and good; often, however, it will be difficult to estimate, much less bound, the derivative term in (3.2-9).] It is desirable then to use that interpolation formula which gives the best results at every stage of the computation.

Consider the problem of estimating $\ln .65$ using the data in the difference table of Example 3.3. Suppose that a priori we do not know how many differences will be required to obtain the accuracy we need.† If all the data in the table will be required to achieve the desired accuracy, then it makes no appreciable difference which finite-difference interpolation formula we use because all will be algebraically equivalent. But if it is possible that a sufficiently accurate result can be obtained using fewer than six differences, we should choose our interpolation formula with some care.

Let us compare the use of Newton's backward formula and Gauss's forward formula. If we may need to use all the data in the table, then the Newtonian formula must use $x_0 = .9$ (i.e., $m = -2.5$) and the Gaussian formula must use $x_0 = .6$ ($m = .5$). But while these two formulas will be algebraically equivalent if terms through the sixth difference are used, for smaller numbers of terms, they will not be equivalent [20]. Therefore, which should we choose?

This question is most easily answered by considering the error term (3.2-9). The only term in the error that we can control is the $p_n(x)$ term. To minimize the magnitude of $p_n(x)$, we should choose the tabular points so that the value of x at which we wish to interpolate is as near as possible to the center of the interval spanned by the tabular points (why?). Therefore, the answer to our question is that the Gaussian formula is to be preferred in the above example because, when the number of differences used is small, it more nearly satisfies the condition above than the Newtonian formula.

From the above it follows that Newton's backward formula has its chief value when we wish to interpolate near the end of a table, for in this

† For such a simple function as this, we could, of course, estimate the error using (3.2-9).

case there would not be a sufficient number of differences available for the Gaussian formula. For example, to estimate $\ln .85$ using the data of Example 3.3, if we used Gauss's forward formula with $x_0 = .8$, then we could only use the terms through the second difference [21]. Similarly, Newton's forward formula is chiefly valuable near the beginning of a table. But, when there are a substantial number of tabular points available on either side of the interpolation point, a Gaussian formula is more desirable than either Newtonian formula. In particular, Stirling's formula (which is just the average of two Gaussian formulas) terminated with an even difference (so that it is equivalent to a Lagrangian formula) is useful when m can be chosen near zero; similarly, Bessel's formula terminated with an odd difference is useful when m is near $\frac{1}{2}$. The justification for these conclusions is considered in [21].

Example 3.5 Use Newton's backward formula with $x_0 = .9$ and Gauss's forward formula with $x_0 = .6$ to find an estimate of $\ln .65$.

Using (3.4-4), (3.4-5), and the data of Example 3.3, we may construct the following table:

Number of differences used	Newton's backward formula ($m = -2.5$)	Gauss's forward formula ($m = .5$)
0	-.105361	-.510826
1	-.399819	-.433751
2	-.429346	-.430229
3	-.430869	-.430701
4	-.430762	-.430821
5	-.430791	-.430792
6	-.430774	-.430775

The true value is $\ln .65 = -.430783$. As we expect, when the number of differences is small, the Gaussian formula is more accurate than the Newtonian formula, although both give the same value, except for roundoff, when all six differences are used. (Why is the Newtonian formula more accurate when four differences are used?) Using (3.2-9) we may verify that the error at every stage is within expected bounds [20].

The reader may well think that any desired degree of accuracy could be achieved merely by increasing the number of terms used in any interpolation formula (finite difference or Lagrangian).† In fact, interpolation series formed by letting the number of tabular points go to infinity are generally only asymptotically convergent; that is, as we add more points

† We ignore here the fact that the growth of roundoff error with higher differences limits the accuracy attainable with finite-difference interpolation formulas.

the error first decreases and then at some point starts to increase and grow without bound †. One reason for this eventual divergence of interpolation series is connected with the fact that the n th derivative of all but some entire functions (functions with no singularities in the complex plane) eventually grows without bound as n increases (see Sec. 4.11). Even for entire functions, however, the interpolation series may fail to converge [22]. We note that, in practice, the desired degree of accuracy in interpolation can almost always be achieved. That is, the asymptotic convergence is generally very good indeed.

3.6 ITERATED INTERPOLATION

An important advantage of finite-difference interpolation formulas over the Lagrangian formula would seem to be the property of the former that enables a term to be added to them merely by adding one tabular point and computing an additional row of differences. As we demonstrated in Example 3.5, this enables us to generate a sequence of *interpolants* each one involving one more tabular point than the previous one. Therefore, the convergence of the interpolation procedure can be tested easily. But suppose, given the Lagrangian interpolation formula using n points, we wish to add one point to get higher accuracy. A look at (3.2-4) indicates that, even if we have saved the values of $p'_n(a_j)$, $j = 1, \dots, n$, each $l_j(x)$, $j = 1, \dots, n$ requires some recalculation, and we must also calculate $l_{n+1}(x)$. Our purpose in this section is to show how this seeming disadvantage of the Lagrangian formula can be overcome. We shall do this by using *iterated interpolation* in which a sequence of interpolants in the Lagrangian context is generated without the need for substantial recalculation of coefficients when going from n to $n + 1$ points.

Denote by $y_{n_1, \dots, n_n}(x)$ the Lagrangian interpolation formula which uses the points a_{n_1}, \dots, a_{n_n} which we do not require to be equally spaced. Then in particular we may write

$$y_{1,2,\dots,n}(x) = \frac{1}{a_n - a_{n-1}} \left| \begin{array}{cc} y_{1,2,\dots,n-1}(x) & a_{n-1} - x \\ y_{1,2,\dots,n-2,n}(x) & a_n - x \end{array} \right| \quad (3.6-1)$$

This equation can be verified by noting that the right-hand side, which is a polynomial of degree $n - 1$, takes on the values $f(a_i)$ at the points a_i , $i = 1, \dots, n$. Equation (3.6-1) then indicates how a Lagrangian formula of order n can be generated from lower-order formulas. By use of the following table, we may generalize the result of (3.6-1) to achieve our

† The classic example of this behavior is the very well-behaved function $f(x) = 1/(1 + x^2)$ which is considered by Steffensen (1950, pp. 35-38).

object [note that $y_i(x) = f(a_i)$]:

$$\begin{array}{ccccccc} a_1 & a_1 - x & y_1(x) & & & & \\ a_2 & a_2 - x & y_2(x) & y_{1,2}(x) & & & \\ a_3 & a_3 - x & y_3(x) & y_{1,3}(x) & y_{1,2,3}(x) & & \\ \dots & \dots & \dots & \dots & \dots & \dots & \\ a_n & a_n - x & y_n(x) & y_{1,n}(x) & y_{1,2,n}(x) & \dots & y_{1,2,3,\dots,n}(x) \end{array} \quad (3.6-2)$$

The entries in each column of the table can be generated from the entries in the previous column by analogy with (3.6-1). For example

$$y_{1,2,n}(x) = \frac{1}{a_n - a_2} \begin{vmatrix} y_{1,2}(x) & a_2 - x \\ y_{1,n}(x) & a_n - x \end{vmatrix} \quad (3.6-3)$$

The entries on the diagonal in (3.6-2) are just what we were seeking. They form a sequence of Lagrangian interpolants each of which incorporates one more tabular point than the previous one. Further, since each entry in (3.6-2) is calculated using a formula analogous to (3.6-1), the process is easily mechanized. Iterated interpolation is, thus, well suited to digital-computer application and, for points not equally spaced, is also convenient for hand computation.

Example 3.6 Use iterated interpolation to calculate $\ln .54$ using the data of Example 3.2.

Corresponding to the table (3.6-2), we get

a_i	$a_i - x$	$y_i(x)$				
.40	-.14	-.916291				
.50	-.04	-.693147	-.603889			
.60	.06	-.510826	-.632466	-.615320		
.70	.16	-.356675	-.655137	-.614139	-.616029	
.80	.26	-.223144	-.672690	-.613196	-.615957	-.616144

We are not bound to use the natural order of the points as above. Consider instead iterated interpolation using the ordering below

a_i	$a_i - x$	$y_i(x)$				
.50	-.04	-.693147				
.60	.06	-.510826	-.620219			
.40	-.14	-.916291	-.603889	-.615320		
.70	.16	-.356675	-.625853	-.616839	-.616029	
.80	.26	-.223144	-.630480	-.617141	-.615957	-.616144

The difference in the two final values from the result of Example 3.2 is the result of roundoff since the same five points were used in all computations. Note, however, that the first interpolant (-.620219) in the second calculation is substantially more accurate than that in the first calculation (-.603889). This occurs because in the second computation we arranged the data so that the magnitudes in the $(a_i - x)$ column would be increasing. In this way the value of $p_n(x)$ in the error term is minimized at every stage (cf. the discussion of Sec. 3.5). Therefore, if we order the tabular points so that the magnitudes in the $a_i - x$ column are increasing, then each interpolant tends to be the best possible ["tends" because the value of the derivative in the error may be greater when $p_n(x)$ is smaller]. In this way the convergence of the

interpolation (as judged by the difference in two successive interpolants or by the stabilization of a certain number of decimal places) will tend to be most rapid. In this example only the first interpolant is improved because only the tabular point at .40 is out of the best possible order.

3.7 INVERSE INTERPOLATION

In Chap. 8 we shall be concerned with the solution of the general nonlinear equation $f(x) = 0$. One of our basic tools in the solution of this equation will be inverse interpolation, which we shall now consider briefly. The solution of $f(x) = 0$ is one example of the common numerical problem of finding the zero of a function. Another case where this occurs is in the numerical integration of an ordinary differential equation (see Chap. 5) when we would like to know that value of the independent variable for which the dependent variable (i.e., the solution of the differential equation) is zero. Inverse interpolation provides us with a straightforward and powerful way to find such zeros of functions.†

Let the function whose zero (or zeros) we wish to find be $y = f(x)$ and suppose it is tabulated at a series of points (which need not necessarily be equally spaced) so that we have

$$\begin{array}{ccccccc} x: & x_1 & x_2 & \dots & x_n & & \\ y = f(x): & f(x_1) & f(x_2) & \dots & f(x_n) & & \end{array} \quad (3.7-1)$$

Now let us suppose that on the interval $[x_1, x_n]$, $f(x)$ satisfies the conditions of the inverse-function theorem [i.e., in particular that $f'(x) \neq 0$] so that we may write $x = g(y)$ where g is the function inverse to f . Therefore, finding the value of $g(0)$ is equivalent to finding a zero of $f(x)$. To estimate $g(0)$ we first write the table (3.7-1) as

$$\begin{array}{ccccccc} y: & f(x_1) & f(x_2) & \dots & f(x_n) & & \\ x = g(y): & x_1 & x_2 & \dots & x_n & & \end{array} \quad (3.7-2)$$

Now in the context of interpolation let $f(x_1), \dots, f(x_n)$ be the tabular points of the independent variable y (not equally spaced in general) and let x_1, \dots, x_n be the functional values at these points. Then, if we use a Lagrangian interpolation formula to approximate $g(y)$ by a polynomial and then interpolate at the point $y = 0$, we get the desired approximation to $\alpha = g(0)$.

Example 3.7 Given the data

$x:$.1	.2	.3	.4	.5
$f(x):$.70010	.40160	.10810	-.17440	-.43750

find an approximate value of the zero of $f(x)$ between .3 and .4.

† We note in passing that even the Newton-Raphson method for the solution of $f(x) = 0$ can be considered to be an application of inverse interpolation; see Chap. 8.

Our approach will be to use iterated interpolation. We therefore first arrange the data in order of increasing magnitude of $f(x)$ (cf. Example 3.6) and then use the technique of the previous section to generate the table

$f(x_i)$	$f(x_i) - 0$	x_i					
.10810	.10810	3					
-.17440	-.17440	4	.33827				
.40160	.40160	2	.33683	.33783			
-.43750	-.43750	5	.33963	.33737	.33761		
.70010	.70010	1	.33652	.33792	.33771	.33765	

The data for $f(x)$ are in fact values of the function $x^4 - 3x + 1$, which has a zero .33767 correctly rounded to five places.

Expressed in terms of $g(y)$ the error in inverse Lagrangian interpolation is

$$E(y) = [p_n(y)/n!]g^{(n)}(\xi) \quad (3.7-3)$$

Now derivatives of g can be expressed in terms of f , and although this relation is not simple (see [1], Chap. 8), there is a power of $f'(x)$ in the denominator of each derivative of $g(y)$; e.g.,

$$g'(y) = 1/f'(x) \quad g''(y) = -f''(x)/[f'(x)]^3$$

Therefore, although we can carry through the process of inverse interpolation even if $f'(r)$ vanishes in $[x_1, x_n]$, we would expect the accuracy to be very poor in this case. When $f'(x)$ vanishes near the zero we can, however, often find the zero by an iterative process involving linear inverse interpolation [30].

3.8 HERMITE INTERPOLATION

In this section we consider the case $m = 1$ in (3.1-1). In particular, we suppose that the first derivative as well as the function is known at r of the n tabular points. In place of (3.1-5) we have then

$$f(x) = \sum_{j=1}^n h_j(x)f(a_j) + \sum_{j=1}^r \bar{h}_j(x)f'(a_j) + E(x) = y(x) + E(x) \quad (3.8-1)$$

where now the approximation $y(x)$ is given by

$$y(x) = \sum_{j=1}^n h_j(x)f(a_j) + \sum_{j=1}^r \bar{h}_j(x)f'(a_j) \quad (3.8-2)$$

and $h_j(x)$ and $\bar{h}_j(x)$ are both polynomials. Again using the criterion of exact approximation, we require that the error term $E(x)$ be such that

$$\begin{aligned} E(a_j) &= 0 & j &= 1, \dots, n \\ E'(a_j) &= 0 & j &= 1, \dots, r \end{aligned} \quad (3.8-3)$$

In analogy with Eq. (3.2-4) in the Lagrangian case, this leads to the following conditions that must be satisfied by the $h_j(x)$ and $\bar{h}_j(x)$.

$$\begin{aligned} h_j(a_k) &= \delta_{jk} & j, k &= 1, \dots, n \\ \bar{h}_j(a_k) &= 0 & j &= 1, \dots, r; k = 1, \dots, n \\ h'_j(a_k) &= 0 & j &= 1, \dots, n; k = 1, \dots, r \\ \bar{h}'_j(a_k) &= \delta_{jk} & j, k &= 1, \dots, r \end{aligned} \quad (3.8-4)$$

Since there are $n + r$ conditions to satisfy in (3.8-3), we expect that $y(x)$ will have to be a polynomial of degree $n + r - 1$ [i.e., we shall approximate $f(x)$ by a polynomial of degree $n + r - 1$ passing through $f(a_j)$, $j = 1, \dots, n$ and having derivatives $f'(a_j)$, $j = 1, \dots, r$]. In deriving the $h_j(x)$ and $\bar{h}_j(x)$, we shall use the notation

$$\begin{aligned} p_n(x) &= (x - a_1) \cdots (x - a_n) \\ p_r(x) &= (x - a_1) \cdots (x - a_r) \\ l_{jn}(x) &= \frac{p_n(x)}{(x - a_j)p'_n(a_j)} & j &= 1, \dots, n \\ l_{jr}(x) &= \frac{p_r(x)}{(x - a_j)p'_r(a_j)} & j &= 1, \dots, r \end{aligned} \quad (3.8-5)$$

To satisfy the conditions on $h_j(x)$, we set

$$h_j(x) = \begin{cases} l_j(x)l_{jn}(x)l_{jr}(x) & j = 1, \dots, r \\ l_{jn}(x)[p_r(x)/p_r(a_j)] & j = r + 1, \dots, n \end{cases} \quad (3.8-6)$$

where $l_j(x)$ is a linear polynomial so that $h_j(x)$ is of degree $n + r - 1$. As given by (3.8-6), $h_j(x)$ satisfies all the conditions of (3.8-4) except $h_j(a_j) = 1$, $j = 1, \dots, r$ and $h'_j(a_j) = 0$, $j = 1, \dots, r$. To satisfy these we must have

$$\begin{aligned} l_j(a_j) &= 1 & j &= 1, \dots, r \\ l'_j(a_j) + l'_{jn}(a_j) + l'_{jr}(a_j) &= 0 & j &= 1, \dots, r \end{aligned} \quad (3.8-7)$$

Similarly, if we set

$$\bar{h}_j(x) = s_j(x)l_{jr}(x)l_{jn}(x) \quad j = 1, \dots, r \quad (3.8-8)$$

with $s_j(x)$ a linear polynomial we must have

$$\begin{aligned} s_j(a_j) &= 0 & j &= 1, \dots, r \\ s'_j(a_j) &= 1 & j &= 1, \dots, r \end{aligned} \quad (3.8-9)$$

in order to satisfy (3.8-4). Linear functions satisfying (3.8-7) and (3.8-9) are easily found to be [31]

$$l_j(x) = 1 - (x - a_j)[l'_{jn}(a_j) + l'_{jr}(a_j)] \quad s_j(x) = x - a_j \quad (3.8-10)$$

This completes the determination of $h_j(x)$ and $\bar{h}_j(x)$.

To find $E(x)$ we proceed in a manner similar to the Lagrangian case. Let

$$F(z) = f(z) - y(z) - [f(x) - y(x)] \frac{p_n(z)p_r(z)}{p_n(x)p_r(x)} \quad (2.8-11)$$

with x not one of the tabular points. This function has $n + r + 1$ zeros (double zeros at a_1, \dots, a_r , single zeros at a_{r+1}, \dots, a_n and x) so that by a generalization of Rolle's theorem [36], there exists a ξ in the interval spanned by a_1, \dots, a_n and x such that

$$0 = F^{(n+r)}(\xi) = f^{(n+r)}(\xi) - [f(x) - y(x)] \frac{(n+r)!}{p_n(x)p_r(x)} \quad (3.8-12)$$

Thus

$$E(x) = \frac{p_n(x)p_r(x)}{(n+r)!} f^{(n+r)}(\xi) \quad (3.8-13)$$

This relation also is correct if x is one of the tabular points (why?).

The interpolation formula (3.8-1) then becomes

$$f(x) = \sum_{j=1}^n h_j(x)f(a_j) + \sum_{j=1}^r \bar{h}_j(x)f'(a_j) + \frac{p_n(x)p_r(x)}{(n+r)!} f^{(n+r)}(\xi) \quad (3.8-14)$$

with

$$h_j(x) = \begin{cases} \{1 - (x - a_j)[l'_{j,n}(a_j) + l'_{j,r}(a_j)]\} l_{j,n}(x)l_{j,r}(x) & j = 1, \dots, r \\ l_{j,n}(x)[p_r(x)/p_r(a_j)] & j = r + 1, \dots, n \end{cases} \quad (3.8-15)$$

$$\bar{h}_j(x) = (x - a_j)l_{j,r}(x)l_{j,n}(x) \quad j = 1, \dots, r \quad (3.8-16)$$

and is called the *modified Hermite interpolation formula*. When $r = n$ the formula is

$$f(x) = \sum_{j=1}^n h_j(x)f(a_j) + \sum_{j=1}^n \bar{h}_j(x)f'(a_j) + \frac{p_n^2(x)}{(2n)!} f^{(2n)}(\xi) \quad (3.8-17)$$

with

$$\begin{aligned} h_j(x) &= [1 - 2(x - a_j)l'_{j,n}(a_j)]l_{j,n}^2(x) \\ \bar{h}_j(x) &= (x - a_j)l_{j,n}^2(x) \end{aligned} \quad j = 1, \dots, n \quad (3.8-18)$$

where we have replaced $l_{j,n}(x)$ by $l_j(x)$. Equation (3.8-17) is the *Hermite interpolation formula*, sometimes also called the formula for osculatory interpolation.

Both the Hermite and modified Hermite formulas can be useful interpolation formulas. They also serve as useful theoretical tools in other areas of numerical analysis, as we shall see in Chaps. 4, 5, and 8.

Example 3.8 Given the table below of the natural logarithm and its derivative

x	$\ln x$	$1/x$
40	-.916291	2.50
50	-.693147	2.00
70	-.356675	1.43
80	-.223144	1.25

estimate the value of $\ln .60$ using the Hermite interpolation formula.

From (3.8-18) we get

$$\begin{aligned} h_1(.60) &= .134 & \bar{h}_1(.60) &= .3180 \\ h_2(.60) &= .327 & \bar{h}_2(.60) &= .245 \\ h_3(.60) &= .327 & \bar{h}_3(.60) &= -.345 \\ h_4(.60) &= .134 & \bar{h}_4(.60) &= -.3180 \end{aligned}$$

and from (3.8-17)

$$\ln .60 = -.510824$$

whereas the true value is $-.510826$. Using (3.8-13) the error is bounded by

$$-.000031 \approx \frac{-1}{32768} < E(.60) < -\frac{1}{2^{15}} \approx -.0000001$$

so that the excellent agreement between the interpolated and true values is to be expected.

3.9 GENERAL POLYNOMIAL INTERPOLATION. THE DETERMINANT APPROACH

In the Lagrangian interpolation formula, $y(x)$ was required to be a polynomial of degree less than n with the property that $y(a_i) = f(a_i)$, $i = 1, \dots, n$. This can be expressed by writing

$$y(x) = c_0 + c_1x + \dots + c_{n-1}x^{n-1} \quad (3.9-1)$$

and

$$f(a_i) = c_0 + c_1a_i + \dots + c_{n-1}a_i^{n-1} \quad i = 1, \dots, n \quad (3.9-2)$$

In (3.9-2) we have a system of n equations for the n c_i 's. If we solve this system for each c_i using Cramer's rule and substitute the result into (3.9-1), we may write (3.9-1) in the determinantal form [33]

$$\begin{vmatrix} y(x) & 1 & x & \dots & x^{n-1} \\ f(a_1) & 1 & a_1 & \dots & a_1^{n-1} \\ \dots & \dots & \dots & \dots & \dots \\ f(a_n) & 1 & a_n & \dots & a_n^{n-1} \end{vmatrix} = 0 \quad (3.9-3)$$

The correctness of (3.9-3) is easily verified by noting that $y(x)$ is certainly a polynomial of degree $n - 1$ which vanishes at the points a_i , $i = 1, \dots, n$.

n . In a similar fashion [33], we may show that the equation for $y(x)$ in the modified Hermite interpolation formula may be written

$$\begin{vmatrix} y(x) & 1 & x & x^2 & \cdots & x^{n+r-1} \\ f(a_1) & 1 & a_1 & \cdots & \cdots & a_1^{n+r-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ f(a_n) & 1 & a_n & \cdots & \cdots & a_n^{n+r-1} \\ f'(a_1) & 0 & 1 & 2a_1 & \cdots & (n+r-1)a_1^{n+r-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ f'(a_r) & 0 & 1 & 2a_r & \cdots & (n+r-1)a_r^{n+r-2} \end{vmatrix} = 0 \quad (3.9-4)$$

Now let us consider the general problem of deriving an interpolation formula using the operator (3.1-1) with the property that at the tabular point a , the approximation $y(x)$ and its first r_i derivatives agree with $f(x)$ and its first r_i derivatives. The determinantal equation for $y(x)$ is then [33]

$$\begin{vmatrix} y(x) & 1 & x & \cdots & x^{r_1} & \cdots & x^{r_n} & \cdots & x^\beta \\ f(a_1) & 1 & a_1 & \cdots & \cdots & \cdots & \cdots & \cdots & a_1^\beta \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ f^{(r_1)}(a_1) & 0 & \cdots & \cdots & (r_1)! & \cdots & \cdots & \cdots & \frac{\beta!}{(\beta-r_1)!} a_1^{\beta-r_1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ f(a_n) & 1 & a_n & \cdots & \cdots & \cdots & \cdots & \cdots & a_n^\beta \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ f^{(r_n)}(a_n) & 0 & \cdots & \cdots & \cdots & \cdots & (r_n)! & \cdots & \frac{\beta!}{(\beta-r_n)!} a_n^{\beta-r_n} \end{vmatrix} = 0 \quad (3.9-5)$$

where $\beta = n - 1 + \sum_{i=1}^n r_i$. It is easy to see that (3.9-3) and (3.9-4) are special cases of (3.9-5). However, before we can assert that (3.9-5) defines a valid interpolation polynomial $y(x)$, we must show not only that $y(x)$ and its derivatives have the desired values at the tabular points but also that the minor of y in (3.9-5) does not vanish identically, for in this case there would be no formula. In the case of (3.9-3) and (3.9-4) our derivations of the Lagrangian and Hermite formulas are assurance of this. Thus it should be no surprise that, in the general case (3.9-5), it can be proved that, independent of the tabular points, the minor of y does not vanish identically; we leave the proof to a problem [35].

The interpolation polynomials [i.e., the coefficients of $f^{(r_i)}(a_i)$] for this general case may be derived by techniques similar to those used in the Lagrangian and Hermite cases, but of particular interest to us here is the form of the error term since we shall have use for this in Sec. 8.5-1. As we

would expect, (3.9-5) is exact for polynomials of degree β or less. Therefore, by analogy with (3.2-9) and (3.8-13) the error is given by [36]

$$E(x) = \frac{f^{(\beta+1)}(\xi)}{(\beta+1)!} \prod_{j=1}^n (x - a_j)^{r_j+1} \quad (3.9-6)$$

When all the r_j are equal, Eq. (3.9-5) is called the formula for hyperosculatory interpolation.

For the still more general case in which there may be gaps in the sequence of derivatives prescribed at a point (e.g., the function and its second derivative but not its first derivative may be specified at a point), a determinantal equation analogous to (3.9-5) may still be written down [37], but it may now be true that for some choices of tabular points the minor of y vanishes identically [37].

3.10 OTHER METHODS OF INTERPOLATION. EXTRAPOLATION

In interpolation, as in all branches of numerical analysis, there will be special cases in which methods superior to the general ones derived in this chapter can be derived and used without an unreasonable expenditure of effort. One example of this is the case of periodic functions in which methods based on Fourier-series approximations may be preferable to the polynomial approximations of this chapter; for more on this, see Sec. 6.8-1. In fact, whenever the function to be interpolated is known to have a special functional character, an approximation based on this known functional character may be desirable [38]. Of course, on a digital computer, if we use functions other than polynomials, these functions themselves must be evaluated using polynomial or rational approximations.

Although we are restricting ourselves in this book to functions of a single variable, interpolation of functions of two or more variables can often be effected by a sequence of interpolations using the formulas of this chapter [39].

This chapter is entitled Interpolation, but it has been equally about extrapolation. As their definition in Sec. 2.2-1 makes clear, interpolation and extrapolation are two aspects of the same type of procedure. Of the two, interpolation is much more common than extrapolation. The reason for this is straightforward and practical. We argued in Sec. 3.5 that $p_n(x)$ is minimized when x is as nearly as possible in the center of the interval spanned by the tabular points. Conversely, as x moves *outside* the interval spanned by the tabular points—as is the case in extrapolation—the factors $x - a_i$ in $p_n(x)$ grow, and therefore, the error tends to grow also. Thus extrapolation is inherently a more inaccurate process than interpolation and must therefore always be used with extreme caution.

When extrapolation must be used in some form—see, for example, Chap. 5—then the value of x should be restricted to be as near as possible to the interval spanned by the tabular points.

BIBLIOGRAPHIC NOTES

3.1-3.5 The topics covered in these sections will be found in virtually any textbook on numerical analysis. In particular, excellent discussions of interpolation may be found in Hildebrand (1956), Kopal (1955), and Kuntzmann (1959). The orientation in these books as well as in most other numerical analysis texts is much more toward difference and divided-difference techniques [25] than in this book. An excellent though somewhat older reference to classical interpolation techniques is Steffensen (1950). Hartree (1958) and Whittaker and Robinson (1948) contain a number of practical hints for special situations.

The coefficients of both the Lagrangian and finite-difference interpolation formulas have been extensively tabulated. A bibliography of these tables may be found in Fletcher, Miller, and Rosenhead (1962).

The error term in the Lagrangian formula is discussed in a more general context in Milne (1949). The derivation of the error term used here may also be found in Scarborough (1962). For another approach, see Hildebrand (1956) or Kopal (1955).

Our discussion of the lozenge diagram follows closely that of Hamming (1962); see also Kopal (1955). A thorough discussion of the use of differences in detecting errors in tables is given by Miller (1950). The use of difference methods in the construction of mathematical tables is considered by Fox (1957b); see also Fox (1957a).

The operational techniques introduced in Prob. 16 are further considered in the problems after the next chapter. A thorough discussion of these techniques may be found in Hildebrand (1956).

3.6 The basic references on iterated interpolation are the papers by Aitken (1932) and Neville (1934).

3.7 Inverse interpolation will be considered in much greater detail in Chap. 8; for tables of coefficients for particular cases of inverse interpolation using differences see Salzer (1943, 1944, 1945).

3.8-3.10 Hermite interpolation is discussed in many texts [e.g., Hildebrand (1956), Kopal (1955)]. The convergence of interpolation series [22] is considered by Erdos and Turan (1937). A detailed discussion of interpolation in several variables is given by Steffensen (1950); see also Pearson (1920).

BIBLIOGRAPHY

- Aitken, A. C. (1932): On Interpolation by Iteration of Proportional Parts, without the Use of Differences, *Proc. Edinburgh Math. Soc.*, vol. 3, series 2, pp. 56-76.
- Erdos, P., and P. Turan (1937): On Interpolation, I, *Ann. of Math.*, vol. 38, pp. 142-155.
- Fletcher, A., J. C. P. Miller, and L. Rosenhead (1962): *An Index of Mathematical Tables*, 2d ed., Addison-Wesley Publishing Company, Inc., Reading, Mass.
- Fox, L. (1957a): Minimax Methods in Table Construction in *On Numerical Approximation* (R. E. Langer, ed.), The University of Wisconsin Press, Madison, Wis.

- Fox, L. (1957b): *The Use and Construction of Mathematical Tables*, vol. 1, 2d ed., Physical Laboratory Mathematical Tables Series, London.
- Hamming, R. W. (1962): *Numerical Methods for Scientists and Engineers*, McGraw-Hill Book Company, New York.
- Hartree, D. R. (1958): *Numerical Analysis*, Oxford University Press, Fair Lawn, N.J.
- Hildebrand, F. B. (1956): *Introduction to Numerical Analysis*, McGraw-Hill Book Company, New York.
- Kopal, Z. (1955): *Numerical Analysis*, John Wiley & Sons, Inc., New York.
- Kuntzmann, J. (1959): *Methodes numériques, Interpolation—dérivées*, Dunod, Paris.
- Miller, J. C. P. (1950): Checking by Differences, *MTAC*, vol. 4, pp. 3-11.
- Milne, W. E. (1949): The Remainder in Linear Methods of Approximation, *J. Res. Nat. Bur. Standards*, vol. 43, pp. 501-511.
- Neville, E. H. (1934): Iterative Interpolation, *J. Indian Math. Soc.*, vol. 20, pp. 87-120.
- Pearson, K. (1920): *On the Construction of Tables and on Interpolation*, II, *Bivariate Interpolation*, University of London, Tracts for Computers III, Cambridge University Press, New York.
- Salzer, H. E. (1943): Tables of Coefficients for Inverse Interpolation with Central Differences, *J. Math. and Phys.*, vol. 22, pp. 210-224.
- Salzer, H. E. (1944): Tables of Coefficients for Inverse Interpolation with Advancing Differences, *J. Math. and Phys.*, vol. 23, pp. 75-102.
- Salzer, H. E. (1945): Inverse Interpolation for Eight, Nine, Ten and Eleven Point Direct Interpolation, *J. Math. and Phys.*, vol. 24, pp. 106-108.
- Scarborough, J. B. (1962): *Numerical Mathematical Analysis*, 5th ed., The Johns Hopkins Press, Baltimore.
- Steffensen, J. F. (1950): *Interpolation*, Chelsea Publishing Company, New York.
- Whittaker, E. T., and W. Robinson (1948): *The Calculus of Observations*, 4th ed., Blackie & Son, Ltd., Glasgow.

PROBLEMS

Section 3.2

1. (a) Assuming that the derivatives at x_1 can be calculated, discuss the relative accuracy of an interpolation formula based on the operator derived in Prob. 16, Chap. 2 with $N = n$ and an n -point Lagrangian interpolation formula.

(b) Use this interpolation formula with $n = 4$ and $x_1 = .50$ to approximate $\ln .60$. Compare this result with that of Example 3.1. Why is $\ln x$ one of the few functions for which interpolation using a truncated Taylor series is practical?

2. (a) If n is the order of a Lagrangian interpolation formula, show that

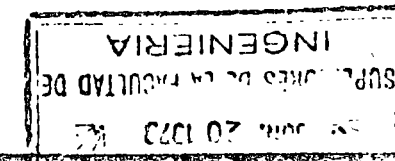
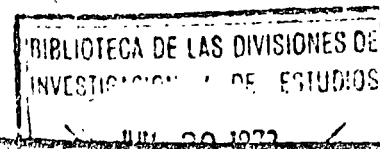
$$\sum_{j=1}^n a_j^k l_j(x) = x^k \quad k = 0, \dots, n-1$$

where the a_j are the tabular points.

(b) For $n = 3$ and equally spaced tabular points, compute

$$\max_{\alpha_1, \alpha_2} |l_j(x)|$$

for $j = 1, 2, 3$. Use Table 3.1 to estimate the bounds on $l_j(x)$ for $n = 5$. Use these



results to make an inference on the importance of roundoff error in interpolation using equally spaced data.

Section 3.3

3. (a) Using equally spaced data and a three-point Lagrangian formula, find a bound on $h^2 y'''(x)$ which, on the interval spanned by the three points, assures a truncation error of less than 10^{-4} where d is an integer.

(b) Similarly, find a bound on $h^2 y''(x)$ when using a five-point Lagrangian formula.

(c) Use these results to estimate the maximum value of h , for both the three- and five-point cases, that can be used to interpolate (i) $\sin x$ on $[-\pi, \pi]$, (ii) e^x on $[-4, 4]$, (iii) $\sin 100x$ on $[-\pi, \pi]$, with a truncation error of less than 10^{-10} .

4. (a) Show that $y(x)$ in the Lagrangian interpolation formula is the unique polynomial of degree $n - 1$ passing through the points $\{a_i, f(a_i)\}$.

(b) Use the Lagrangian interpolation formula to find the cubic passing through the points $(-3, -1), (0, 2), (3, -2), (6, 10)$.

5. (a) Do the computation of Examples 3.1 and 3.2 with the same tabular points when $f(x) = \sin x$.

(b) Repeat part a using $\tan^{-1} x$.

*6. Consider the following table for the Bessel functions $J_p(x)$, $p = 0, 1, 2, 3, 4, 5$ correctly rounded to four decimal places

x	$J_0(x)$	$J_1(x)$	$J_2(x)$	$J_3(x)$	$J_4(x)$	$J_5(x)$
2.0	.2239	.5767	.3528	.1289	.0340	.0070
2.1	.1566	.5683	.3746	.1453	.0405	.0088
2.2	.1104	.5560	.3951	.1623	.0476	.0109
2.3	.0855	.5399	.4139	.1800	.0556	.0134
2.4	.0625	.5202	.4310	.1981	.0649	.0162
2.5	-.0484	.4971	.4461	.2166	.0738	.0195
2.6	-.0968	.4708	.4590	.2353	.0870	.0232
2.7	-.1424	.4416	.4696	.2540	.0950	.0274
2.8	-.1850	.4097	.4777	.2727	.1067	.0321
2.9	-.2243	.3754	.4832	.2911	.1190	.0373
3.0	-.2601	.3391	.4861	.3091	.1320	.0430

(a) Suppose you wished to interpolate to find values of $J_0(x)$ at $x = 2.05 + .1j$, $j = 0, \dots, 9$. Use the relation

$$J'_j(x) = -J_{p+1}(x) + (p/x)J_p(x)$$

to find a bound on the truncation error in the worst case using (i) linear interpolation; (ii) a Lagrangian three-point formula. Which of these methods would you use if you wished to guarantee a total error in the result for every j of less than 5×10^{-4} in magnitude?

(b) Carry out the interpolation using this method.

(c) Repeat parts a and b to find values of $J_1(x)$ at $x = 2.05 + .1j$, $j = 1, \dots, 9$.

(d) How many correctly rounded decimal places for $J_0(x)$ would have to be given in order that the use of a five-point Lagrangian formula would give significantly higher accuracy than the three-point formula?

7. Use the data of Prob. 6 and a three-point Lagrangian formula to approximate (a) $J_2(2.07)$, (b) $J_2(2.405)$, (c) $J_2(2.64)$, (d) $J_2(2.91)$, with $p = 0, 1, 2$.

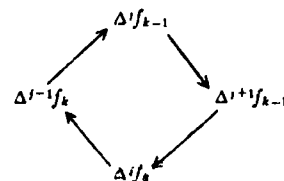
8. Derive (3.3.5).

9. Show that the first difference (forward, backward, or central) of a polynomial of degree n is a polynomial of degree $n - 1$. Thus deduce that the n th difference of a polynomial of degree n is a constant and the $(n + 1)$ st is zero.

10. Difference operators. Define the shifting operator E to be such that $Ef(x) = f(x + h)$. Using this and the definitions of Δ, ∇ , and δ establish the following identities:

(a) $\Delta = E - 1$; (b) $\nabla = 1 - E^{-1}$; (c) $\delta = E^{1/2} - E^{-1/2}$. Then use these relations to derive relations between Δ and ∇ and between Δ and δ .

*11. (a) Using the rules of Sec. 3.3-2-2, show that any closed path of the form



results in no contribution to any interpolation formula.

(b) Thus deduce that the path from $\Delta^{-1}f_k$ to Δf_{k-1} to $\Delta^{+1}f_{k-1}$ results in the same contribution as the path from $\Delta^{-1}f_k$ to Δf_k to $\Delta^{+1}f_{k-1}$. Similarly, show that the path from $\Delta^{-1}f_k$ to Δf_{k-1} to $\Delta^{+1}f_{k-1}$ to Δf_k and the path from $\Delta^{-1}f_k$ to Δf_k result in the same contribution. From these results, deduce that any closed path contributes nothing.

(c) Show also that the path from f_{i+1} to Δf_i to f_i contributes nothing.

(d) Use the results of parts a, b, and c to deduce that all formulas which terminate on a given difference and start anywhere in the functional-value column are algebraically equivalent.

12. (a) Suppose a table contains two errors in successive entries. Show how the difference patterns will overlap in the second difference and describe how you would determine the magnitude of each error. Do the same for the third difference. Generalize to show how you would unravel any two overlapping error patterns.

(b) The following tabulation may contain one or more misprints. By differencing the tabulation, correct these misprints. Describe the way you detect the misprints.

2001409	1634722	1267432
1949050	1582304	1215141
1896683	1529878	1162645
1844308	1477443	1110161
1791930	1424999	1057658
1739532	1372548	1005146
1687132	1320088	

[Ref.: Kopal (1955), p. 86.]

(c) Find and correct the misprints in the tabulation of $J_4(x)$ in Prob. 6.

*13. Suppose a single entry in a table is in error by an amount ϵ in the least significant digit and let all the other entries in the table be in error by the maximum roundoff error of $1/2$ in the least significant digit, alternately plus and minus. Therefore, in the position of the least significant digit, the error pattern would look like $\dots -1/2, 1/2, -1/2, 1/2, \epsilon, 1/2, -1/2, 1/2, \dots$

(a) In terms of ϵ , find the error in the difference of order $2k$, $k = 1, \dots, 6$ on the line with the ϵ entry.

(b) By considering the case where $\epsilon = -\frac{1}{2}$, find ϵ_{\max} , the largest error in a functional value which cannot be distinguished from roundoff in the difference of order $2k$, $k = 1, \dots, 6$

(c) Derive a formula for ϵ_{\max} in terms of k .

[Ref.: Miller (1950).]

*14. (a) Given a table of values at an interval h , discuss how you would generate a new table ("subtabulate") at an interval ρh ($0 < \rho < 1$) by using an appropriate interpolation formula.

(b) Show that as $n \rightarrow \infty$, the left-hand side of (3.3-11) approaches $f(a_0 + hm)$ if the series on the right-hand side converges (see Prob. 22).

(c) Let the forward-difference operator with respect to the interval ρh be represented by Δ_1 . Using (3.3-8) and the result of part b, show that

$$\Delta_1^j f_0 = \sum_{i=0}^{\infty} \sum_{k=0}^j (-1)^{i+k} \binom{j}{k} \binom{k\rho}{i} \Delta^i f_0 \quad j = 1, 2, \dots$$

(d) Use the results of Prob. 10 to show that in operational form

$$\Delta_1^j f_0 = [(1 + \Delta)^\rho - 1]^j f_0$$

Use this to calculate Δ_1^j , $j = 1, 2, 3, 4$ in terms of Δ^j and ρ , retaining terms through Δ^4 .

(e) Use the results of part c to subtabulate the data of Prob. 6 for $J_0(x)$ with (i) $\rho = \frac{1}{2}$; (ii) $\rho = \pm \frac{1}{3}$. Compare the results for $\rho = \frac{1}{2}$ with those of Prob. 6. How could you overcome the problems that arise near the end of the tabulation?

Section 3.4

15. (a) Derive the identities

$$(i) \sum_{k=0}^m (-1)^k \binom{m}{k} = 0 \quad (ii) \binom{r}{m} \binom{m}{k} = \binom{r-k}{m-k} \binom{r}{k}$$

(b) Use these results to show that $\sum_{j=r}^i (-1)^{i-j} \binom{i}{j} \binom{j}{r}$ vanishes and thus deduce

that the right-hand side of (3.4-1) is f_i .

16. (a) Use the results of Prob. 10 to express Newton's backward formula in terms of backward differences at a_0 .

(b) Similarly, express Gauss's forward and backward formulas in terms of central differences at a_0 .

(c) Using the notation $\mu\delta^{2m+1}f_0 = \frac{1}{2}(\delta^{2m+1}f_{\frac{1}{2}} + \delta^{2m+1}f_{-\frac{1}{2}})$ express Stirling's and Bessel's formulas in terms of central differences.

17. (a) Show that in any finite-difference interpolation formula a difference of any order can be eliminated by using the relation $\delta^m f_{k+1} - \delta^m f_k = \delta^{m+1} f_k$ or a similar relation for forward and backward differences.

(b) Use this result and the result of Prob. 16b to eliminate the odd differences from Gauss's forward formula and thus derive *Everett's interpolation formula*

$$y(m) = (1-m)f_0 + mf_1 - \frac{m(m-1)(m-2)}{3!} \delta^2 f_0 + \frac{(m+1)(m)(m-1)}{6} \delta^2 f_1 + \dots$$

(This formula is useful in interpolating in tables which provide auxiliary tables of even central differences.)

(c) Similarly, eliminate the even differences in Gauss's forward formula to get *Steffensen's interpolation formula*

$$y(m) = f_0 + \frac{(m+1)m}{2!} \delta f_{\frac{1}{2}} - \frac{m(m-1)}{2} \delta f_{-\frac{1}{2}} + \frac{(m+2)(m+1)m(m-1)}{4!} \delta^2 f_{\frac{1}{2}} - \dots$$

[Ref.: Hildebrand (1956), pp. 103-105, or Kopal (1955), pp. 50-54.]

*18. Throwback. (a) Use the result of Prob. 16c to show that the ratio of the coefficient B_4 of the fourth central difference in Bessel's formula to the coefficient B_2 of the second difference is $(m+1)(m-2)/12$ and that for $0 \leq m \leq 1$ this ratio varies between $-\frac{1}{6}$ and $-\frac{1}{3}$.

(b) Because this ratio varies very little on this interval, consider replacing B_4 by cB_2 . Show that $B_4 - cB_2$ as a function of m has a maximum independent of c and two minima dependent on c on $[0,1]$. Find the two values of c which equalize the minimum and maximum values of $B_4 - cB_2$ on this interval. Show that one of these values c_1 is very nearly equal to the average value of B_4/B_2 over $[0,1]$.

(c) Thus rewrite Bessel's formula as $y(m) = \frac{1}{2}(f_0 + f_1) + (m - \frac{1}{2})\delta f_{\frac{1}{2}} + B_2[\delta^2 f_0 + \delta^2 f_1] + c_1(\delta^2 f_0 + \delta^2 f_1)$. This procedure is called *throwback*; that is, we have "thrown back" the effect of the fourth difference onto the second difference. [Ref.: Kopal (1955), pp. 54-56.]

19. (a) Display the error terms for the Newtonian and Gaussian interpolation formulas terminated with the difference of order k in terms of h and m .

(b) Use these to derive the error terms for Stirling's and Bessel's formulas terminated with an odd or even difference.

Section 3.5

20. (a) What abscissas are involved in the calculation of each entry in the table in Example 3.5 for both the Gaussian and Newtonian formulas?

(b) Verify that the actual error using six differences is consistent with that calculated using the result of Prob. 19a

21. (a) How many terms in Gauss's forward formula can be used if x_0 is (i) the next to last entry in a table; (ii) the fourth entry?

(b) Use (3.4-7) and (3.4-8) to show that, when m is near zero, Stirling's formula is a desirable one to use and that, when m is near one-half, Bessel's formula is desirable.

*22. (a) If h is fixed, show that in the limit as $n \rightarrow \infty$ Newton's forward formula, if it converges, becomes with $a_0 = 0$

$$f(x) = f_0 + \sum_{j=1}^{\infty} \frac{\Delta^j f_0}{j! h^j} x(x-h) \cdots [x - (j-1)h]$$

(b) For $f(x) = e^{ax}$ and $a_0 = 0$, show that $\Delta^j f_0 = (e^{ah} - 1)^j$.

(c) By using the result of part b in part a, show that the ratio of the $k+1$ and k terms of the series is given by

$$\frac{e^{ah} - 1}{h(k+1)} (x - kh)$$

(d) By considering this ratio as $k \rightarrow \infty$, deduce that the series in part a converges if $e^{ah} < 2$ and diverges if $e^{ah} > 2$ unless x is a positive integral multiple of h , in which

does it converge? (A more difficult result is that for $e^x = 2$ the series converges if and only if $x > -h$.)

(c) Thus deduce that Newton's forward formula is an asymptotic series for e^{ax} when $e^{ah} > 2$. Contrast this with the convergence of the Taylor series for e^{ax} for all ax . In practice, why would we expect Newton's formula to be asymptotic even when $e^{ah} < 2$? [Ref.: Hildebrand (1956), pp. 114-116.]

23. Suppose you have a table of $\sin x$ at an interval $h = .1$. How many tabular points would have to be used in interpolating in this table to assure a truncation error of less than (a) 10^{-2} ; (b) 10^{-4} ; (c) 10^{-6} ; independent of a_0 and m ?

24. Use the data of Prob. 6 and a finite-difference interpolation formula to approximate (a) $J_p(2.07)$, (b) $J_p(2.405)$, (c) $J_p(2.64)$, (d) $J_p(2.91)$, with $p = 0, 1, 2$. In each case motivate your choice of a particular interpolation formula and compare the results with those of Prob. 7.

*25. Divided differences. The divided difference of order $k > 1$ of $f(x)$ is defined by

$$f[a_1, \dots, a_k] = \frac{f[a_2, \dots, a_k] - f[a_1, \dots, a_{k-1}]}{a_k - a_1}$$

with $f[a_i] = f(a_i)$.

(a) Prove that

$$f[a_1, \dots, a_k] = \sum_{i=1}^k \frac{f(a_i)}{(a_i - a_1) \cdots (a_i - a_{i-1})(a_i - a_{i+1}) \cdots (a_i - a_k)}$$

and thus deduce that the order of the arguments in a divided difference is immaterial.

(b) Use the data of Example 3.1 to generate a divided-difference table analogous to the difference table of Fig. 3.1.

(c) Show that

$$f[a_1, \dots, a_{k-1}, x] = f[a_1, \dots, a_k] + (x - a_k)f[a_1, \dots, a_k, x]$$

and use this result to derive the formula

$$f(x) = f[a_1] + (x - a_1)f[a_1, a_2] + (x - a_1)(x - a_2)f[a_1, a_2, a_3] + \cdots + (x - a_1)(x - a_2) \cdots (x - a_{n-1})f[a_1, \dots, a_n] + E(x)$$

where

$$E(x) = p_n(x)f[a_1, \dots, a_n, x]$$

This formula is called *Newton's divided-difference interpolation formula*.

(d) Deduce from part c that $(x - a_k)f[a_1, \dots, a_n, x] \rightarrow 0$ as $x \rightarrow a_k$, $k = 1, \dots, n$.

(e) Use this result to show that this formula must be algebraically equivalent to the Lagrangian interpolation formula which uses the tabular points a_1, \dots, a_n . Thus deduce that

$$f[a_1, \dots, a_n, x] = \frac{1}{n!} f^{(n)}(\xi)$$

where ξ is in the interval spanned by a_1, \dots, a_n and x .

(f) Use the results of parts c and e to show that when $a_i \rightarrow a$, $i = 1, \dots, n$ the Newton divided-difference formula and, therefore, the Lagrangian formula are both equivalent to a Taylor series with remainder.

(g) Use Newton's divided-difference formula and the table of part b to approximate $\ln .60$. Compare the result with Example 3.1.

(h) When the tabular points are equally spaced, show that

$$f[a_1, \dots, a_k] = \frac{1}{(k-1)!h^{k-1}} \Delta^{k-1}f_1$$

and thus use part c to derive Newton's forward formula.

Section 3.6

26. (a) Show that the table of (3.6-2) can be replaced by the symmetrical arrangement

a_1	$a_1 - x$	$y_1(x)$			
.	.	.	$y_{11}(x)$	$y_{12}(x)$	
.	.	.	$y_{21}(x)$.	$y_{22}(x)$
.	$y_{32}(x)$
.	$y_{42}(x)$
.	.	.	.	$y_{n-2,n-1,n}(x)$.
.	.	.	$y_{n-1,n}(x)$.	$y_n(x)$
a_n	$a_n - x$	$y_n(x)$			

What would the additional entries to the table be if the point a_{n+1} were used?

(b) Use the technique of part a to do the computation of Example 3.6. [Ref.: Neville (1934).]

27. Use iterated interpolation to do the calculations of parts a and b of Prob. 7. Compare the results with those of Probs. 7 and 24.

28. Interpolation near a singularity. Suppose you are given a tabulation of sine, cosine, and tangent as follows:

x	$\sin x$	$\cos x$	$\tan x$
1.566	.999885	.0047963	208.49128
1.567	.999928	.0037963	263.41125
1.568	.999961	.0027963	357.61106
1.569	.999984	.0017963	556.69098
1.570	.999997	.0007963	1255.76559

Using these data and any interpolation formula, calculate $\tan(1.5695)$ by (a) using the $\tan x$ tabulation directly; (b) calculating $\sin(1.5695)$ and $\cos(1.5695)$. Discuss the reasons for the varying errors in the two results. [Ref.: Kopal (1955), p. 84.]

Section 3.7

29. Use the data of Prob. 6 and inverse interpolation to approximate the zero of $J_0(x)$ between 2 and 3 (cf. Prob. 7b).

30. Inverse interpolation near a singularity. Suppose we wished to calculate that value of x for which $\sin x = .9999950$ using the data of Prob. 28. Why will the procedure of Sec. 3.7 not work here? To solve this problem, do the following:

(a) Obtain an initial approximation \bar{x} to x by linear inverse interpolation between $x_1 = 1.567$ and $x_2 = 1.568$.

(b) By direct interpolation, compute $\sin \bar{x}$.

If $|x - \bar{x}| < 0.0000050$, replace x_1 by \bar{x} and repeat this procedure. Otherwise, replace x_2 by \bar{x} . Continue until the process converges. What condition must $\sin x$ satisfy on $[x_1, x_2]$ in order for the process to converge?

Section 3.8

31. Derive Eqs. (3.8-10).

32. Use the data of Prob. 6 and Hermite interpolation to do the computation of parts *a* and *b* of Prob. 7 for $p = 0$. Compare with the previous results. Can the use of the Hermite interpolation formula be simplified for equally spaced data in a fashion analogous to that for the Lagrangian formula in Sec. 3.3-1?

Section 3.9

33. (a) Verify that, when Eqs. (3.9-3) and (3.9-4) are solved for $y(x)$, the results are, respectively, the Lagrangian and Hermite interpolation formulas.

(b) Similarly, show that $y(x)$ given by (3.9-5) is such that $y(x)$ and its first r_i derivatives agree with $f(x)$ at a_i , $i = 1, \dots, n$.

34. (a) Show that the value of the Vandermonde determinant

$$\Delta(a_1, \dots, a_n) = \begin{vmatrix} 1 & a_1 & \dots & a_1^{n-1} \\ 1 & a_2 & \dots & a_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & a_n & \dots & a_n^{n-1} \end{vmatrix}$$

contains all factors of the form $(a_i - a_j)$, $i \neq j$.

(b) By showing that there are $n(n-1)/2$ such factors and that the determinant is a polynomial of this degree in the variables a_1, \dots, a_n , deduce that

$$\Delta = \prod_{j>i=1}^n (a_j - a_i)$$

*35. (a) Use the result of the previous problem to derive the form of $l_i(x)$ directly from (3.9-3). Deduce that the minor of $y(x)$ in (3.9-3) does not vanish if the tabular points are distinct.

(b) Show directly that the minor of $y(x)$ in (3.9-4) does not vanish if the tabular points are distinct.

(c) Generalize this result to show that the minor of $y(x)$ in (3.9-5) does not vanish if the tabular points are distinct.

36. (a) Let $f(x)$ have zeros x_i , $i = 1, \dots, n$ of multiplicity ν_i , $i = 1, \dots, n$ in an interval (a, b) . Let $\nu = \sum_{i=1}^n \nu_i$. Prove that $f^{(k)}(x)$ has at least $\nu - k$ zeros in (a, b) .

(b) Use this result and a technique similar to that used to derive the Lagrangian and Hermite interpolation formula errors to derive (3.9-6).

37. (a) Write the determinantal form for an interpolation polynomial $y(x)$ which at the n tabular points is to have the same value and same second derivative as the function $f(x)$ (but not necessarily the same first derivative).

(b) Show that, for $n = 3$ and $a_1 = -1$, $a_2 = 0$, $a_3 = 1$, the minor of y in this determinant vanishes identically. Thus deduce that there is no quintic polynomial satisfying the desired conditions. [Ref.: Hamming (1962), p. 94.]

Section 3.10

38. Suppose it is desired to approximate $f(x)$ in the form

$$f(x) = \sum_{i=1}^n c_i e^{a_i x} = \sum_{i=1}^n c_i u_i^x \quad (u_i = e^{a_i})$$

(a) If the approximation is to be exact at the n equally spaced points, $0, 1, \dots, n-1$, write down the n equations for the c_i 's (assuming the u_i 's are known).

(b) Given the data

x :	0	1	2
$f(x)$:	2.4400	2.0851	2.1958

use the results of part *a* to find the coefficients of an exponential approximation with $a_1 = 0$, $a_2 = -1$, $a_3 = -2$. [Ref.: Hildebrand (1956), p. 380.]

39. Interpolation of functions of two variables. Suppose we are given a function of two variables $f(x, y)$ tabulated at points (a_i, b_j) , $i = 1, \dots, n$, $j = 1, \dots, m$.

(a) If we wish to approximate $f(x, y)$ at a nontabular point, show that we can do this by first interpolating to find $f(x, b_j)$ for a sequence of values of j and then using these values to interpolate to find $f(x, y)$ or vice versa.

(b) Given the table of values of the elliptic integral

$$E(x, y) = \int_0^y (1 - \sin^2 x \sin^2 t)^{1/2} dt$$

$x \backslash y$	50°	54°	58°	62°
50°	0.8134	0.8060	0.7988	0.7920
52°	0.8414	0.8332	0.8251	0.8174
54°	0.8690	0.8598	0.8508	0.8422
56°	0.8962	0.8859	0.8759	0.8663

find an approximation to $E(55.4^\circ, 53.1^\circ)$ by (i) interpolating horizontally to find $E(55.4^\circ, y)$ for $y = 50^\circ, 52^\circ, 54^\circ, 56^\circ$ and then interpolating vertically; (ii) interpolating vertically and then horizontally. If the desired point lies on a diagonal [e.g., $(52^\circ, 51^\circ)$], how could the interpolation procedure be simplified? [Ref.: Hildebrand (1956), p. 125.]

chapter 4

**NUMERICAL DIFFERENTIATION,
NUMERICAL QUADRATURE,
AND SUMMATION**

4.1 NUMERICAL DIFFERENTIATION FORMULAS

The form of the general numerical differentiation operator is

$$L[f(x)] = f^{(k)}(x) + \sum_{i=0}^m \sum_{j=1}^n A_{ij}(x) f^{(i)}(a_j) \tag{4.1-1}$$

where at least one $A_{0j}(x) \neq 0$. In this section we shall restrict ourselves to the case $m = 0$ since this is by far the most important case in practice. Implicit in this and the following sections on numerical differentiation is the existence and, where necessary, continuity of as many derivatives of $f(x)$ as we require.

Our basic approach in deriving numerical differentiation formulas will be to differentiate the interpolation formulas of the previous chapter. If we differentiate the Lagrangian interpolation formula (3.1-5) k times, we get

$$f^{(k)}(x) = \sum_{j=1}^n l_j^{(k)}(x) f(a_j) + \frac{d^k}{dx^k} \left[\frac{p_n(x)}{n!} f^{(n)}(\xi) \right] = y^{(k)}(x) + \frac{d^k}{dx^k} [E(x)] \tag{4.1-2}$$

In particular, for $k = 1$ we have

$$f'(x) = \sum_{j=1}^n l_j'(x) f(a_j) + \frac{d}{dx} \left[\frac{p_n(x)}{n!} f^{(n)}(\xi) \right] \tag{4.1-3}$$

where the derivative of $l_j(x)$ is easily calculated using (3.2-4). Determination of the error term in (4.1-2) or (4.1-3) presents a problem because ξ is an unknown function of x (see Sec. 3.2). Nevertheless, we can prove the following

Theorem 4.1 Let $p_n(x)f^{(n)}(\xi)/n!$ be the error term in the Lagrangian interpolation formula with ξ in the interval spanned by a_1, \dots, a_n and x . Then, if $f^{(n+1)}(x)$ is continuous,

$$\frac{1}{n!} \frac{d}{dx} f^{(n)}(\xi) = \frac{1}{(n+1)!} f^{(n+1)}(\eta) \tag{4.1-4}$$

where η is also in the interval spanned by a_1, \dots, a_n and x .

Proof If we take the Lagrangian interpolation formula (3.1-5) with $x \neq a_j, j = 1, \dots, n$, divide both sides by $p_n(x)$, and differentiate, we get, using (3.2-4),

$$\frac{d}{dx} \frac{f(x)}{p_n(x)} = \sum_{j=1}^n \frac{f(a_j)}{-(x-a_j)^2 p_n'(a_j)} + \frac{1}{n!} \frac{d}{dx} [f^{(n)}(\xi)] \tag{4.1-5}$$

Now consider (3.1-5) with n replaced by $n+1$. We have

$$\begin{aligned} p_{n+1}(x) &= p_n(x)(x - a_{n+1}) \\ p_{n+1}'(a_j) &= \begin{cases} (a_j - a_{n+1})p_n'(a_j) & j \neq n+1 \\ p_n(a_{n+1}) & j = n+1 \end{cases} \end{aligned} \tag{4.1-6}$$

Dividing (3.1-5) by $p_{n+1}(x)$, rearranging terms, and using (4.1-6), we may write

$$\begin{aligned} \frac{[f(x)/p_n(x)] - [f(a_{n+1})/p_n(a_{n+1})]}{x - a_{n+1}} &= \sum_{j=1}^n \frac{f(a_j)}{(x-a_j)(a_j - a_{n+1})p_n'(a_j)} + \frac{f^{(n+1)}(\tau)}{(n+1)!} \end{aligned} \tag{4.1-7}$$

where τ is in the interval spanned by a_1, \dots, a_{n+1} and x . Now take the limit of both sides of (4.1-7) as $a_{n+1} \rightarrow x$. We get

$$\frac{d}{dx} \frac{f(x)}{p_n(x)} = \sum_{j=1}^n \frac{f(a_j)}{-(x-a_j)^2 p_n'(a_j)} + \frac{f^{(n+1)}(\eta)}{(n+1)!} \tag{4.1-8}$$

where η is in the interval spanned by a_1, \dots, a_n and x . Comparing (4.1-5) and (4.1-8), the theorem is proved when $x \neq a_j$. But using the continuity of the derivatives, (4.1-4) must be true for any x .

By an extension of this argument, we may prove that [1]

$$\frac{1}{n!} \frac{d^n}{dx^n} f^{(n)}(\xi) = \frac{j!}{(n+j)!} f^{(n+j)}(\eta) \quad (4.1-9)$$

with η , in the interval spanned by a_1, \dots, a_n and x . Using the result of this theorem and Leibniz's rule, we may write the error term in (4.1-2) as

$$\frac{d^k}{dx^k} \left[\frac{p_n(x)}{n!} f^{(n)}(\xi) \right] = \sum_{i=0}^k \frac{k!}{i!} p_n^{(i)}(x) \frac{f^{(n+k-i)}(\eta_{k-i})}{(n+k-i)!} \quad (4.1-10)$$

and, in particular, for (4.1-3)

$$\frac{d}{dx} \left[\frac{p_n(x)}{n!} f^{(n)}(\xi) \right] = \frac{p_n(x)}{(n+1)!} f^{(n+1)}(\eta_1) + \frac{p_n'(x)}{n!} f^{(n)}(\eta_0) \quad (4.1-11)$$

From (4.1-11) we note that, if we are estimating the derivative at a tabular point, then the first term on the right-hand side is zero. This means in effect that *the derivative at a tabular point can be calculated by differentiating the Lagrangian formula while considering ξ to be a constant.* By using a technique similar to that used to derive the error in the Lagrangian formula in Sec. 3.2, we may also show that, if x is outside or at an end point of the interval spanned by a_1, \dots, a_n , then the error may again be written by dropping the first term on the right-hand side of (4.1-11) [1] (and, in general, changing η_0 to some other value in the interval spanned by a_1, \dots, a_n and x).

When the tabular points are equally spaced, $y^{(k)}(x)$ in (4.1-2) becomes

$$y^{(k)}(x) = \frac{1}{h^k} \sum_{j=1}^n l_j^{(k)}(m) f(a_j) \quad x = a_0 + hm \quad (4.1-12)$$

where the differentiation of $l_j(m)$ is with respect to m , and we have used the fact that

$$\frac{dy}{dx} = \frac{dy}{dm} \frac{dm}{dx} = \frac{1}{h} \frac{dy}{dm} \quad (4.1-13)$$

In a similar fashion, we may differentiate the interpolation formulas expressed in difference form. For example, if we differentiate Newton's forward formula (3.3-11), we get

$$\begin{aligned} \frac{d^k}{dx^k} y(a_0 + hm) &= \frac{1}{h^k} \frac{d^k}{dm^k} y(a_0 + hm) \\ &= \frac{1}{h^k} \sum_{j=0}^n \frac{d^k}{dm^k} \binom{m}{j} \Delta^j f_0 = \frac{1}{h^k} \sum_{j=k}^n \frac{d^k}{dm^k} \binom{m}{j} \Delta^j f_0 \quad (4.1-14) \end{aligned}$$

Similar equations can clearly be written down for the other finite-difference interpolation formulas [2]. For the particular case $m = 0$, we may write (4.1-14) as

$$\frac{d^k}{dx^k} y(a_0) = \frac{k!}{h^k} \sum_{j=k}^n \frac{S_j^{(k)}}{j!} \Delta^j f_0 \quad (4.1-15)$$

The $S_j^{(k)}$ are called *Stirling numbers of the first kind*. Some of their properties are considered in [4].

More general numerical differentiation formulas can be found by differentiating the general interpolation formula (3.9-5), but as we have mentioned, it is seldom useful to have numerical differentiation formulas which depend on derivatives of the function.

4.2 COMPUTING DERIVATIVES NUMERICALLY

Superficially, there would seem to be no more difficulty to computing derivatives using the formulas of the previous section than there was in using the interpolation formulas of the previous chapter. But a closer look indicates that roundoff error, while often not significant in using interpolation formulas, is not only significant but can be disastrous in numerical differentiation. In this section we shall consider numerical differentiation using the Lagrangian interpolation formula for equally spaced data, but this discussion may easily be extended to finite-difference differentiation formulas.

The roundoff error $R(x)$ incurred in using the Lagrangian interpolation formula (3.3-3) can be bounded by

$$|R(x)| \leq (5 \times 10^{-r-1}) \sum_{j=1}^n |l_j(m)| \quad (4.2-1)$$

if the functional values are all correctly rounded to r decimal places. Moreover, as Table 3.1 indicates, the values of $l_j(m)$ never get much larger than 1 for reasonably small n . Thus for such values of n the roundoff error will affect only the last significant digit (i.e., the r th decimal). Contrast this, however, with Eq. (4.1-12). Computing with this formula, the roundoff error $R_k(x)$ can only be bounded by

$$|R_k(x)| \leq \frac{5 \times 10^{-r-1}}{h^k} \sum_{j=1}^n |l_j^{(k)}(m)| \quad (4.2-2)$$

Clearly, if h is small, the roundoff can be very large. Now for equally spaced points, each term of the truncation error contains a power of h , as can be seen using Eqs. (4.1-10) and (4.1-13). Thus, whereas in interpola-

tion truncation error is proportional to a power of h and roundoff error does not depend on h , in numerical differentiation, truncation error is proportional and roundoff error is inversely proportional to a power of h . A small value of h then causes a large magnification of the roundoff error inherent in the functional values, † and a large value causes a large truncation error. This suggests the problem we now consider of finding the optimal value of h when numerically differentiating at equal intervals.

In particular, let us consider this question when the first derivative at a tabular point is desired. Then the first term on the right-hand-side of (4.1-11) is zero, and we have for the truncation error

$$E_1(x) = [p'_n(x)/n!]f^{(n)}(\eta_0) \tag{4.2-3}$$

Now, using (3.3-4) and (4.1-13), we may write

$$p'_n(x) = h^{n-1}p'_n(m) \tag{4.2-4}$$

so that

$$E_1(x) = h^{n-1}[p'_n(m)/n!]f^{(n)}(\eta_0) = h^{n-1}e_1(x) \tag{4.2-5}$$

where $e_1(x)$ does not depend on h . Similarly, we may write for the roundoff error

$$R_1(x) = (1/h)r_1(x) \tag{4.2-6}$$

where $r_1(x)$ also does not depend on h . A convenient way to define the optimal value of h (see [6] for another) is to choose that value which makes the bounds on the magnitudes of $E_1(x)$ and $R_1(x)$ equal. This will lead to an accurate optimum only when the two bounds are equally good. Nevertheless, this approach is defensible since we are really only looking for a reasonable value of h .

We have

$$e_1(x) \leq [p'_n(m)/n!]M_n \tag{4.2-7}$$

where M_n is such that

$$|f^{(n)}(\xi)| \leq M_n \tag{4.2-8}$$

for ξ in the interval spanned by the tabular points and x . Similarly

$$|r_1(x)| \leq \epsilon \sum_{j=1}^n |l'_j(m)| \tag{4.2-9}$$

where ϵ is the magnitude of the maximum roundoff error in each value of

† Because the factor $1/h^k$ causes a magnification of the roundoff error or, in engineering parlance, the "noise" in the functional values, numerical differentiation is often called a noise-magnification process.

$f(a_j)$ [5×10^{-7} in (4.2-3)]. Using (4.2-7) and (4.2-9) to equate the bounds on the roundoff and truncation errors, we get

$$h^{n-1} \frac{p'_n(m)}{n!} M_n = \frac{\epsilon}{h} \sum_{j=1}^n |l'_j(m)| \tag{4.2-10}$$

so that the optimal value of h is given by

$$h_{opt} = \left[\frac{en! \sum_{j=1}^n |l'_j(m)|}{M_n |p'_n(m)|} \right]^{1/n} \tag{4.2-11}$$

This, of course, determines a different h_{opt} for each m , which is clearly inconvenient. Since commonly we are most interested in the derivative at the central point of an odd number of tabular points, we set

$$m = (n + 1)/2$$

in (4.2-11). But then it is convenient to renumber the tabular points as $a_{(-n+1)/2}, \dots, a_0, \dots, a_{(n-1)/2}$. Doing this, (4.2-11) becomes

$$h_{opt} = \left[\frac{en! \sum_{j=-\frac{(n-1)/2}{(n-1)/2}}^{(n-1)/2} |l'_j(0)|}{M_n |p'_n(0)|} \right]^{1/n} \tag{4.2-12}$$

Using the definition of $l_j(m)$ and (3.3-4), we may show that [6]

$$\sum_{j=-\frac{(n-1)/2}{(n-1)/2}^{(n-1)/2} |l'_j(0)| = 2|p'_n(0)| \sum_{j=1}^{(n-1)/2} \frac{1}{|p'_n(j)j|} \tag{4.2-13}$$

so that

$$h_{opt} = \left[\frac{2en! \sum_{j=1}^{(n-1)/2} \frac{1}{|p'_n(j)j|}}{M_n} \right]^{1/n} \tag{4.2-14}$$

Example 4.1 With $n = 3$, find h_{opt} and expressions for the derivatives at the tabular points.

From (4.2-14)

$$h_{opt} = \left[\frac{12\epsilon}{M_3} \frac{1}{|p'_3(1)|} \right]^{3/5} = \left[\frac{6\epsilon}{M_3} \right]^{3/5} \tag{4.2-15}$$

Then, using (4.1-12) with the tabular points renumbered, $k = 1$, and the error term given by (4.1-11), we get

$$\begin{aligned} f'_{-1} &= (1/2h)(-3f_{-1} + 4f_0 - f_1) + (h^2/3)f'''(\eta_{-1}) \\ f'_0 &= (1/2h)(-f_{-1} + f_1) - (h^2/6)f'''(\eta_0) \\ f'_1 &= (1/2h)(f_{-1} - 4f_0 + 3f_1) + (h^2/3)f'''(\eta_1) \end{aligned} \tag{4.2-16}$$

If the h used is h_{opt} , then the truncation error in f'_0 is bounded by

$$\left(\frac{6\epsilon}{M_2}\right)^{3/4} \frac{M_2}{6} = \epsilon^{2/3} \left(\frac{M_2}{6}\right)^{1/4} \quad (4.2-17)$$

and the roundoff error is bounded by

$$\epsilon \left(\frac{M_2}{6\epsilon}\right)^{3/4} = \epsilon^{2/3} \left(\frac{M_2}{6}\right)^{1/4} \quad (4.2-18)$$

Thus the total error T_1 is such that

$$|T_1| \leq 2\epsilon^{2/3} \left(\frac{M_2}{6}\right)^{1/4}$$

Similar bounds could be derived for the errors in f'_{-1} and f'_1 . The case $n = 5$ is considered in [8].

In practice, we shall generally have empirical data of an unknown function and shall want to estimate the derivative at one of the tabular points. It will then be necessary to estimate M_n to get h_{opt} . Note, however, that the value of h we can use is restricted by the tabular data we are given. The following example indicates how one might calculate the derivative of a tabulated function.

Example 4.2 Given a three-place table of values of $\ln x$ at an interval of .01, find the derivative of $\ln x$ at $x = .5$ using (4.2-16).

Until we have chosen h , we do not know the range over which to estimate M_2 . But suppose we estimate M_2 to be 15 in the range of interest (is this reasonable?). Since $\epsilon = 5 \times 10^{-4}$ in a three-place table, we have from (4.2-15)

$$h_{opt} = \left(\frac{6\epsilon}{M_2}\right)^{3/4} \approx 5.85 \times 10^{-3}$$

A practical value of h to choose is therefore .05. Thus we use the data

x	$\ln x$
.45	-.799
.55	-.598

and (4.2-16) to calculate

$$f'_0 = \frac{1}{.1} (.799 - .598) = 2.01$$

whereas the true value is, of course, 2. The error bound (which is only approximate since we did not use h_{opt}) is

$$|T_1| \leq 2 \times (5 \times 10^{-4})^{2/3} (15/6)^{1/4} \approx 1.7 \times 10^{-2}$$

When empirical data are being differentiated even three-place accuracy may not be available. Thus the $1/2h$ factor in (4.2-16) may cause serious roundoff. If the magnitude of the derivative in the truncation error is such that h cannot be increased without making the truncation error unduly large, then determining the derivative with a reasonable

degree of accuracy may be impossible. When higher derivatives than the first are desired, the $1/h$ factor becomes a $1/h^k$ factor, where k is the order of the derivative, and the roundoff problem becomes just that much worse.

The calculation of derivatives numerically is then a hazardous operation, especially when dealing with low accuracy empirical data. And even for high accuracy data, derivatives higher than the first are likely to have sizable errors. A better approach to numerical differentiation than that considered here may be to first "smooth" the data and then to differentiate (cf. Chap. 6, especially Sec. 6.7).

4.3 APPROXIMATING DERIVATIVES WITH DIFFERENCES

The formulas of Sec. 4.1 enable us to express the derivatives of a function in terms of values of the function or differences of the function. One obvious application of this is in the numerical solution of differential equations. Consider the first-order ordinary differential equation

$$dz/dx = F(x, z) \quad z(x_0) = z_0 \quad (4.3-1)$$

Using (4.1-14) with $k = 1$, $n = 1$, $a_0 = x_0$, and $m = 0$, we may approximate dz/dx at x_0 as

$$\left.\frac{dz}{dx}\right|_{x_0} \approx \frac{1}{h} \Delta z_0 = \frac{z_1 - z_0}{h} \quad (4.3-2)$$

Inserting (4.3-2) in (4.3-1), we get

$$z(x_1) = z(x_0) + hF(x_0, z_0) \quad (4.3-3)$$

which gives us an approximation to the solution of (4.3-1) at x_1 . Having this approximation at x_1 , we could then approximate dz/dx at x_1 using (4.3-2) and then use (4.3-3) to get an approximation to the solution at x_2 , etc. By using more terms of (4.1-14), we could derive more sophisticated methods than (4.3-3). The use of (4.1-14) would enable us to derive the errors inherent in such methods. We shall not consider this matter further here because in Chap. 5 we shall approach the problem of the numerical solution of initial value problems of ordinary differential equations by a more general technique which subsumes all the methods derivable using difference techniques.

When dealing with partial differential equations,† however, replacing derivatives by differences lies at the heart of most methods for solving such equations. Consider, for example, Poisson's equation

$$\partial^2 u / \partial x^2 + \partial^2 u / \partial y^2 = g(x, y) \quad (4.3-4)$$

† And with boundary value problems of ordinary differential equations.

with appropriate boundary conditions (which we shall not state) on the boundary B in Fig. 4.1. Now using (4.1-14) with $k = n = 2$ we get an approximation to the second derivative of $f(x)$ as

$$f''(a_0 + hm) \approx \frac{1}{h^2} \frac{d^2}{dm^2} \binom{m}{2} \Delta^2 f_0 = \frac{f_0 - 2f_1 + f_2}{h^2} \quad (4.3-5)$$

This equation may also be used to approximate second partial derivatives of functions of several variables by holding all but one of the variables constant. To do this for Eq. (4.3-4), let us superimpose a square mesh on the region of Fig. 4.1. Then using (4.3-5) along the line $y = y_1$ with $a_0 = x_1$, we have

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_{\substack{x=x_1 \\ y=y_1}} \approx \frac{u(x_0, y_1) - 2u(x_1, y_1) + u(x_2, y_1)}{h^2} \quad (4.3-6)$$

Similarly, along $x = x_1$ with $a_0 = y_1$

$$\left. \frac{\partial^2 u}{\partial y^2} \right|_{\substack{x=x_1 \\ y=y_1}} \approx \frac{u(x_1, y_0) - 2u(x_1, y_1) + u(x_1, y_2)}{h^2} \quad (4.3-7)$$

Therefore, at the point (x_1, y_1) , we may approximate the partial differential

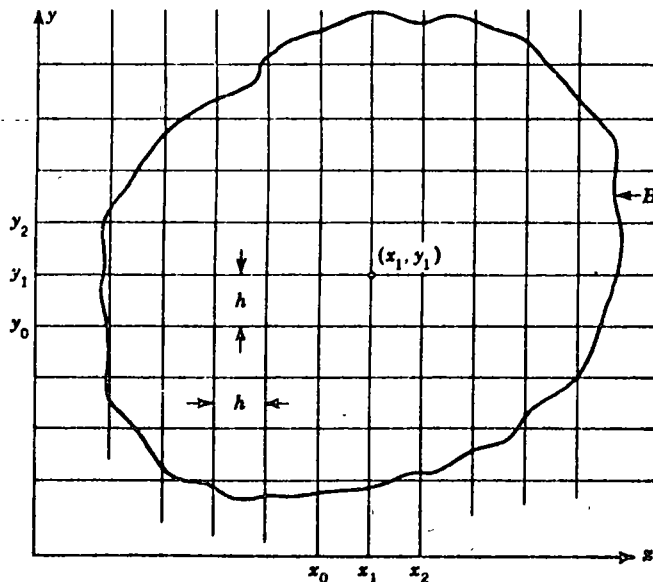


Fig. 4.1 Region in which solution of (4.3-4) is desired.

equation (4.3-4) by the equation

$$\frac{u(x_0, y_1) + u(x_2, y_1) + u(x_1, y_0) + u(x_1, y_2) - 4u(x_1, y_1)}{h^2} = g(x_1, y_1) \quad (4.3-8)$$

Using (4.1-10) the error in this approximation may be derived. An equation analogous to (4.3-8) may be written for every point interior to B by properly using the boundary conditions for points near B [for example, see Chap. 15 of Ralston and Wilf (1960)]. This leads to a system of $(linear)$ equations for the unknowns $u(x_i, y_j)$ which can be solved by one of the methods of Chap. 9 to get an approximate solution to (4.3-4). Techniques very similar to this are the basis of many of the methods for the numerical solution of partial differential equations. A detailed discussion of the numerical solution of partial differential equations is beyond the scope of this book, but an understanding of the basic idea involved in approximating derivatives by differences plus a study of Chap. 9 will give the reader the necessary background for the study of this subject.

One further application of approximating derivatives by differences is in the estimation of error terms. Consider using (4.1-14) with $n = k$ to approximate the k th derivative of a function $f(x)$ at a point $a_0 + hm$. We have

$$f^{(k)}(a_0 + hm) \approx y^{(k)}(a_0 + hm) = \frac{1}{h^k} \frac{d^k}{dm^k} \binom{m}{k} \Delta^k f_0 = \frac{\Delta^k f_0}{h^k} \quad (4.3-9)$$

Now this is certainly a very rough estimate in general, particularly if k is not small, but if the derivatives of order greater than k (and, therefore, also the differences of order greater than k) are well behaved, then (4.3-9) may give an acceptable approximation. Since the difficulty in estimating many of the error terms we have derived already and shall derive later lies in the difficulty in estimating some derivative of $f(x)$, Eq. (4.3-9) can, if used judiciously, enable estimates of error terms to be made when the derivatives of $f(x)$ are not reasonably calculable [9].

4.4 NUMERICAL QUADRATURE—THE GENERAL PROBLEM

The form of the general numerical quadrature operator is

$$L[f(x)] = f(b) - f(a) + \sum_{j=1}^n \sum_{i=1}^m A_{ij} f^{(i)}(a_{ij}) \quad (4.4-1)$$

If we substitute $\int_{-\infty}^{\infty} g(x) dx$ for $f(x)$ in (4.4-1), we get

$$L[f(x)] = L \left[\int_{-\infty}^{\infty} g(x) dx \right] = \int_a^b g(x) dx + \sum_{j=1}^n \sum_{i=1}^m A_{ij} g^{(i-1)}(a_{ij}) \quad (4.4-2)$$

and the quadrature equation then becomes†

$$\int_a^b g(x) dx + \sum_{j=1}^n \sum_{i=1}^m A_{ij} g^{(i)}(a_j) = E \quad (4.4-3)$$

Setting $E = 0$ in (4.4-3) and solving for $\int_a^b g(x) dx$ then gives us an approximation to the definite integral of $g(x)$ as a linear combination of values of $g(x)$ and its derivatives. The numerical quadrature problem is to specify the A_{ij} 's and a_j 's so that this approximation has desirable properties (i.e., achieves some desired accuracy).

Once again our approach will be that of exact polynomial approximation. That is, we shall attempt to choose the A_{ij} 's and a_j 's so that E in (4.4-3) is zero when $g(x)$ is a polynomial of sufficiently low degree. We shall again restrict ourselves mainly to the case $m = 1$; that is, we shall try to express the integral as a linear combination of functional values alone as is done, for example, in the trapezoidal rule. This is by far the most important case both theoretically and practically. With the restriction $m = 1$, we may rewrite (4.4-3) after some obvious changes in notation as

$$\int_a^b f(x) dx = \sum_{j=1}^n H_j f(a_j) + E \quad (4.4-4)$$

One equation of the form (4.4-4) can clearly be derived by integrating the Lagrangian interpolation formula (3.1-5). Without considering the details of this now, we can nevertheless see that, since the Lagrangian formula is exact for polynomials of degree $n - 1$ or less, then so will the formula resulting from its integration. This suggests the question: With no a priori restrictions on the "abscissas" a_j (such as that they be equally spaced) and the "weights" H_j , what is the highest-degree polynomial for which E in (4.4-4) can be made zero? We call the degree of this polynomial the *order of accuracy* of the formula. Since we have $2n$ constants at our disposal— n a_j 's and n H_j 's—we suspect that the answer is a polynomial of degree $2n - 1$. In the next section, we shall show that this is indeed the case.

We shall not explicitly consider the problem of evaluating the indefinite integral

$$y(x) = \int_{x_0}^x f(t) dt \quad (4.4-5)$$

in this chapter. This problem is equivalent to solving the differential

† Here and in the remainder of this chapter, we shall generally denote the error by E instead of $E(x)$ because the variable x will not appear explicitly in the error term as it has previously.

equation

$$\frac{dy}{dx} = f(x) \quad y(x_n) = 0 \quad (4.4-6)$$

and as such can be solved by the techniques of Chap. 5. For any specific value of x the methods of this chapter can, of course, be used to evaluate $y(x)$.

4.5 GAUSSIAN QUADRATURE

For now let us assume that a and b in (4.4-4) are finite. Then, if (4.4-4) is to be exact for polynomials of degree $2n - 1$ or less, we can get a set of $2n$ equations for the $2n$ unknown constants by substituting $f(x) = x^k$, $k = 0, 1, \dots, 2n - 1$ into (4.4-4) and setting $E = 0$. We get

$$\alpha_k = \sum_{j=1}^n H_j a_j^k \quad k = 0, \dots, 2n - 1 \quad (4.5-1)$$

where

$$\alpha_k = \int_a^b x^k dx = \frac{b^{k+1} - a^{k+1}}{k+1} \quad (4.5-2)$$

These nonlinear equations, if we can solve them and if the solution is real, will give us the abscissas and weights we desire. This algebraic approach to our problem is considered further in [10], but we abandon it here in favor of an analytic approach which (1) will tell us without actually calculating the weights and abscissas whether or not they are real; (2) will enable us to determine E when $f(x)$ is not a polynomial of degree $2n - 1$ or less; (3) will enable us to show that the abscissas are in many cases the zeros of well-known polynomials. As we shall see, once the abscissas are known, the weights are easily calculable.

The starting point of our analytical approach is the Hermite interpolation formula (3.8-17)

$$f(x) = \sum_{j=1}^n h_j(x) f(a_j) + \sum_{j=1}^n \bar{h}_j(x) f'(a_j) + \frac{p_n^2(x)}{(2n)!} f^{(2n)}(\xi) \quad (4.5-3)$$

which is exact for polynomials of degree $2n - 1$ or less. Integrating (4.5-3) between a and b , we get

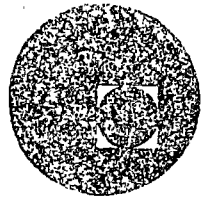
$$\int_a^b f(x) dx = \sum_{j=1}^n H_j f(a_j) + \sum_{j=1}^n \bar{H}_j f'(a_j) + E \quad (4.5-4)$$

where

$$H_j = \int_a^b h_j(x) dx \quad \bar{H}_j = \int_a^b \bar{h}_j(x) dx \quad (4.5-5)$$



centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



METODOS NUMERICOS CON LA COMPUTADORA DIGITAL

TEMA 6: INTREGRACION NUMERICA

Abril, 1978.

8. INTEGRACION NUMERICA

8.1 Introducción

El encontrar la integral de una curva $y=F(X)$ equivale a encontrar el área bajo dicha curva. Analíticamente la integral de una función unidimensional ($F(X)$) está dada por la expresión:

$$g(X) = \int_b^a F(X) dx \quad (8.1)$$

A menudo en problemas de tipo ingenieril las funciones están dadas en forma tabular o gráfica, por ejemplo: valores experimentales, en otras ocasiones es sumamente difícil el integrar una expresión analíticamente debido a su complejidad o puede no existir la integral exacta de dicha función. En todos los casos antes enunciados lo más apropiado o la única vía de solución es un método numérico.

La integración numérica consiste en encontrar la mejor aproximación posible del área que se encuentra debajo de la función a partir de sus valores discretos.

Para tal efecto se considerarán tres métodos:

- Trapezoidal,
- Simpson de 1/3,
- Simpson de 3/8.

La diferencia entre dichos métodos estriba en la cantidad de puntos que emplean para aproximar el área de la función y el error producido al evaluar el área. El programa que se discute a continuación emplea los tres métodos en forma tal que trata de minimizar el error en la evaluación del área mediante una adecuada elección de los métodos a emplear.

8.2 Métodos: Trapezoidal, Simpson de 1/3 y Simpson de 3/8

8.2.1 Objeto

Encontrar la integral de una curva $y = F(X)$ dada en forma discreta mediante los métodos de Simpson 1/3, Simpson 3/8 o Trapezoidal. Los métodos o método a emplear estarán en función de la cantidad de puntos en que esté discretizada la función.

8.2.2 Método

a) Método Trapezoidal

Dado que la integral de una función es el área debajo de la curva, este método lo que hace es dividir el intervalo de integración en "n+1" puntos equidistantes y aproxima la curva original por una serie de rectas en cada uno de los "n" subintervalos; finalmente, se encuentra el área de cada trapezoide y la suma de dichas áreas da la integral en la totalidad del intervalo.

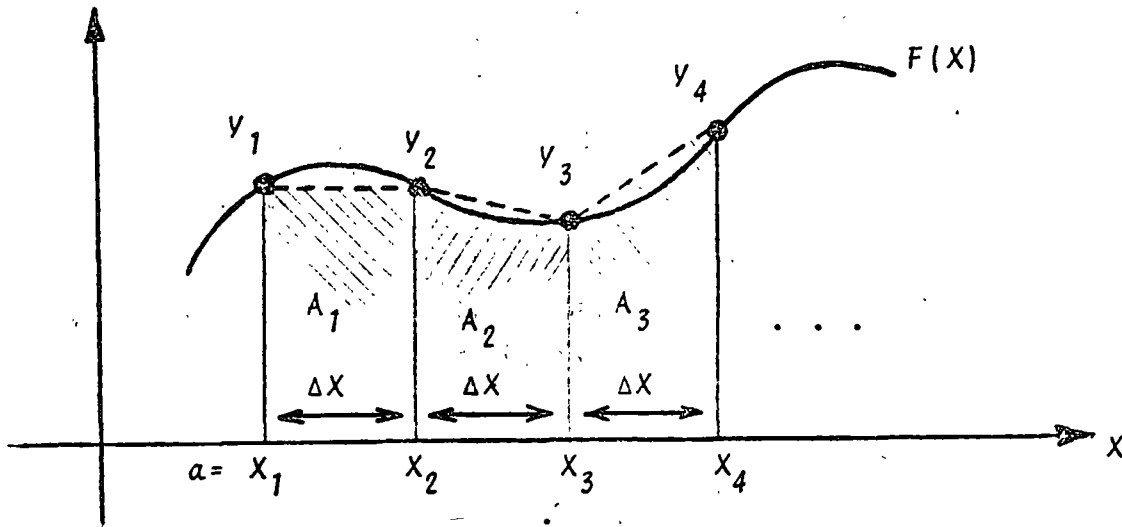


Fig. 8.1 Representación gráfica del método Trapezoidal

Numéricamente se tendrá:

$$\int_a^b F(x) dx \approx \sum_{i=1}^n A_i \quad * \quad (8.2)$$

$$A_1 = \frac{\Delta X}{2} (y_1 + y_2)$$

$$A_2 = \frac{\Delta X}{2} (y_2 + y_3)$$

⋮

$$A_n = \frac{\Delta X}{2} (y_n + y_{n+1}) \quad (8.3)$$

* \approx : aproximadamente

por lo tanto:

$$\int_a^b F(X) dX \approx \frac{\Delta X}{2} (Y_1 + 2Y_2 + \dots + 2Y_n + Y_{n+1}) \quad (8.4)$$

$$\int_a^b F(X) dX \approx \frac{\Delta X}{2} (Y_1 + Y_{n+1} + 2 \sum_{\text{resto orde}} \text{nadas}) \quad (8.5)$$

Para aplicar este método se requiere que el incremento ΔX sea lo más pequeño posible para reducir el error al mínimo. Se puede demostrar que el error producido es del orden de $(\Delta X)^2$.

b) Método de Simpson de 1/3

Este método aproxima tres puntos sucesivos de la función mediante una parábola de segundo grado y evalúa el área debajo de dicha curva. El procedimiento se repite para todos los puntos del intervalo (igualmente espaciados), de tres en tres, y al final se obtiene la suma de todas las áreas.

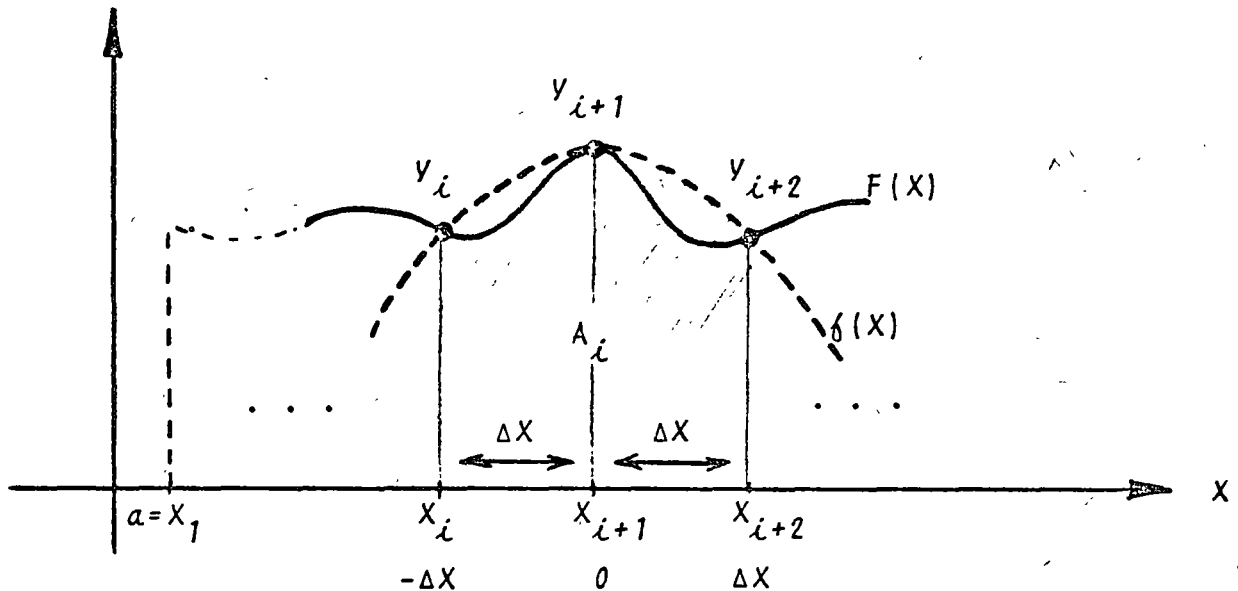


Fig. 8.2 Representación gráfica del método de Simpson de 1/3

Númericamente se tendrá:

$$F(X) \approx f(X) \quad (8.6)$$

$$f(X) = aX^2 + bX + c \quad (8.7)$$

$$\int_{-\Delta X}^{\Delta X} F(X) dX \cong \int_{-\Delta X}^{\Delta X} f(X) dX \quad (8.8)$$

$$A_i = \int_{-\Delta X}^{\Delta X} f(X) dX = \int_{-\Delta X}^{\Delta X} (aX^2 + bX + c) dX \quad (8.9)$$

$$A_i = \frac{2a \Delta X^3}{3} + 2c \Delta X \quad (8.10)$$

Para obtener $a, b,$ y $c,$ se obliga a que la curva (8.7) pase por los puntos muestrales, por lo tanto:

$$y_i = a \Delta X^2 - b \Delta X + c,$$

$$y_{i+1} = c \quad (8.11)$$

$$y_{i+2} = a \Delta X^2 + b \Delta X + c$$

resolviendo el sistema de ecuaciones se tiene:

$$a = \frac{y_i + 2y_{i+1} + y_{i+2}}{2 \Delta X^2}$$

$$b = \frac{y_{i+2} - y_i}{2 \Delta X} \quad (8.12)$$

$$c = y_{i+1}$$

substituyendo (8.12) en (8.10):

$$A_i = \frac{\Delta X (y_i + 4y_{i+1} + y_{i+2})}{3} \quad (8.13)$$

para todo el intervalo de integración se tendrá:

$$\begin{aligned} A_1 &= \frac{\Delta X}{3} (y_1 + 4y_2 + y_3) \\ &\vdots \\ &\vdots \\ A_{n/2} &= \frac{\Delta X}{3} (y_{n-1} + 4y_n + y_{n+1}) \end{aligned} \quad (8.14)$$

como:

$$\int_{x_1=a}^{x_{n+1}=b} F(x) dx \cong \sum_{i=1}^{n/2} A_i \quad (8.15)$$

se tiene:

$$\int_a^b F(x) dx \cong \frac{\Delta X}{3} (y_1 + y_{n+1}) + 2 \sum_{i=3}^n \text{ord. impares} + 4 \sum_{i=2}^n \text{ordenadas pares} \quad (8.16)$$

Para poder emplear el método se requiere que la cantidad de puntos muestrales $(n+1)$ sea non; en caso contrario se emplea una cantidad non de puntos muestrales y el resto del intervalo se integra por el método Trapezoidal. Se puede demostrar que el error producido es del orden de $(\Delta X)^4$.

c) Método de Simpson de 3/8

En este caso se interconectan cuatro puntos consecutivos del intervalo de integración mediante un polinomio de tercer grado y se evalúa el área bajo dicho polinomio en el subintervalo. La integral en todo el intervalo estará dada por la suma de todas las áreas encontradas.

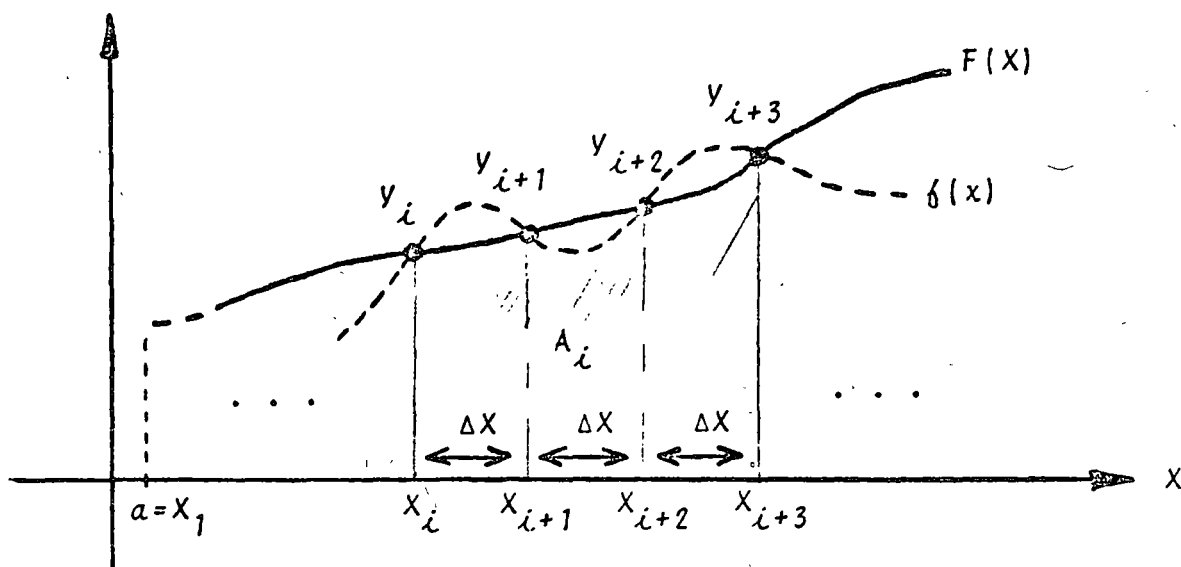


Fig. 8.3 Representación gráfica del método de Simpson de 3/8

De la figura 8.3 se tiene:

$$F(X) \cong f(X), \quad X_i \leq X \leq X_{i+3} \quad (8.17)$$

$$f(X) = aX^3 + bX^2 + cX + d \quad (8.18)$$

$$\int_{X_i}^{X_{i+3}} F(X) dX \cong \int_{X_i}^{X_{i+3}} f(X) dX = A_i \quad (8.19)$$

substituyendo (8.18) en (8.19):

$$A_i = \int_{X_i}^{X_{i+3}} (aX^3 + bX^2 + cX + d) dX \quad (8.20)$$

pero:

$$X_{i+3} - X_i = 3 \Delta X \quad (8.21)$$

empleando la expresión (8.21) al evaluar la expresión (8.20) se obtiene:

$$A_i = \frac{a}{4} (3 \Delta X)^4 + \frac{b}{3} (3 \Delta X)^3 + \frac{c}{2} (3 \Delta X)^2 + d(3 \Delta X) \quad (8.22)$$

De la ecuación (8.18) se obtienen a, b, c, y d al plantear un sistema de ecuaciones como en el método antes descrito y los valores obtenidos se substituyen en la ecuación (8.22) para obtener:

$$A_i = \int_{X_i}^{X_{i+3}} f(X) dX = \frac{3 \Delta X}{8} (y_i + 3y_{i+1} + 3y_{i+2} + y_{i+3}) \quad (8.23)$$

El área en todo el intervalo estará dada por:

$$\int_{X_1=a}^{X_{n+1}=b} F(X) dX \cong \sum_i A_i \quad (8.24)$$

substituyendo (8.23) en (8.24):

$$\int_a^b F(X) dX \cong \frac{3 \Delta X (Y_1 + Y_{n+1})}{8} + 2 \sum_{i=4}^n \text{ord. de orden } (\text{múltiplo de tres}) + 1 + 3 \sum_{i=2}^n \text{resto de ord.} \quad (8.25)$$

Para utilizar el método se requiere que "n" sea múltiplo de tres, existiendo "n+1" valores muestrales. En caso contrario se procede igual que en el método anterior. El error que produce este método es del orden de $(\Delta X)^4$.

En términos generales, cuando se desee integrar una función con la mayor exactitud posible se deberán utilizar los métodos antes descritos con la siguiente jerarquía:

1. Simpson de 3/8
2. Simpson de 1/3
3. Trapezoidal

8.2.3 Descripción del Programa

a) Subrutinas requeridas:

SUBROUTINE RIEMAN(N,H,Y,SINTEG), obtiene el área bajo la curva empleando los métodos Trapezoidal, Simpson de 3/8 y Simpson de 1/3. El programa principal se emplea para la lectura de datos e impresión de resultados.

b) Descripción de las variables:

Para la subrutina RIEMAN:

N	cantidad de puntos en que se discretiza la función
H	espaciamiento entre las abscisas de los puntos muestrales
Y(I)	valores discretizados de la función que se desea integrar
SINTEG	valor de la integral
L(I)	variable que indica cuáles puntos muestrales ya han sido considerados para evaluar la integral

AREA1	integral obtenida por el método Trapezoidal
AREA2	integral obtenida por el método de Simpson de 1/3
AREA3	integral obtenida por el método de Simpson de 3/8
M	variable empleada para determinar el tipo de integración a usar
SUMPAR	sumatoria de las ordenadas de índice par
SUMRES	sumatoria del resto de las ordenadas
SUMTRE	sumatoria de las ordenadas con índice múltiplo de tres + uno

Para el programa principal:

N	cantidad de puntos en que se discretiza la función
H	espaciamiento entre las abscisas de los puntos muestrales
V(I)	valores discretizados de la función que se desea integrar
SINTEG	valor de la integral

c) Dimensiones:

La proposición DIMENSION del programa principal y de la subrutina deberá ser modificada cuando:

$$N > 100$$

d) Formatos para los datos de entrada:

SEC. TARJETAS	FORMATO	INFORMACION
1	(I5, F10.0)	N, H
2	(8F10.0)	V(I), emplear tantas tarjetas como se requieran
:		
:		
:		

 otros paquetes de datos (opcional)

n

TARJETA EN BLANCO, al finalizar toda la información

e) Diagrama de bloques:

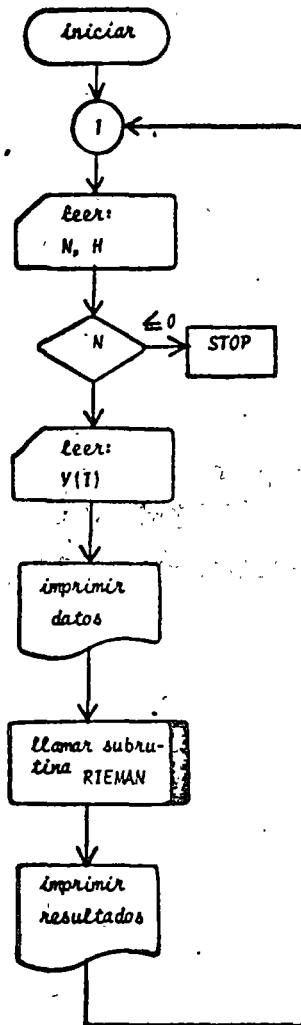


Fig. 8.4 Diagrama de bloques para el programa principal

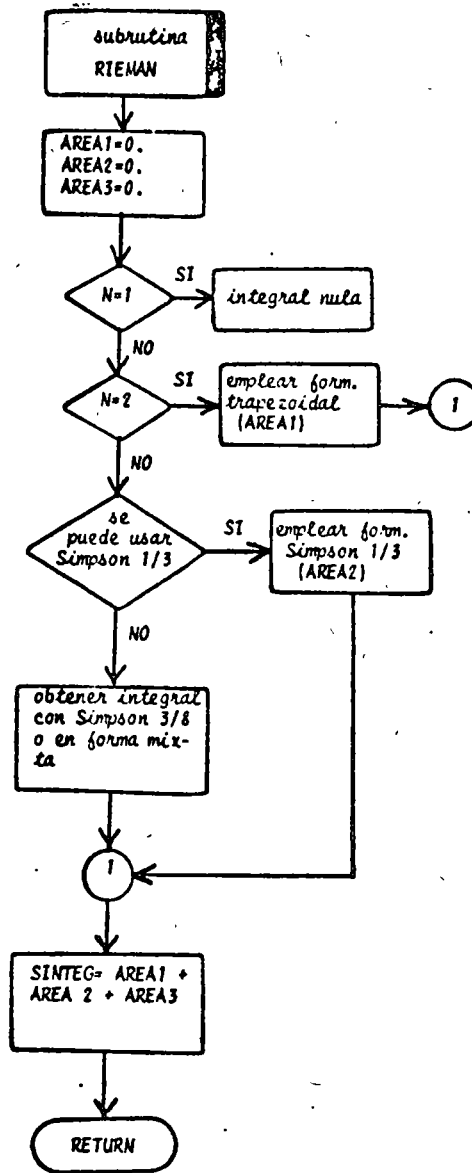


Fig. 8.5 Diagrama de bloques para la subrutina RIEMAN

6) Listado:

```

C   PROGRAMA PARA OBTENER LA INTEGRAL DE UNA FUNCION DISCRETIZADA EN=
C   PLEANDO LOS METODOS DE SIMPSON DE 1/3, SIMPSON 3/8 Y TRAPEZOIDAL
C   SIGNIFICADO DE LAS VARIABLES EMPLEADAS
C   N=CANTIDAD DE PUNTOS EN QUE SE DISCRETIZA LA FUNCION
C   Y=VALORES DISCRETIZADOS DE LA FUNCION
C   H=ESPACIAMIENTO ENTRE ABCISAS
C   SINTEG=INTEGRAL DE LA FUNCION EN EL INTERVALO DADO
-----
C   DIMENSION Y(101)
C   LECTURA DE DATOS
1   READ(5,100) N,H
   IF(N) 2,2,3
2   CALL EXIT
3   READ(5,150) (Y(I),I=1,N)
C   IMPRESION DE DATOS
   WRITE(6,200) N,H
   WRITE(6,250)
   DO 4 I=1,N
4   WRITE(6,300) I,Y(I)
C   LLAMADO DE SUBROUTINA PARA INTEGRAR
   CALL PIENAM(N,H,Y,SINTEG)
C   IMPRESION DE RESULTADOS
   WRITE(6,350) SINTEG
   GO TO 1
C   FORMAS DE LECTURA E IMPRESION
100 FORMAT(15,F10.0)
150 FORMAT(8F10.3)
200 FORMAT(14I,6(//),10X,'CANTIDAD DE PUNTOS MUESTRALES= ',15,3(//),10X,
1'ESPACIAMIENTO ENTRE ABCISAS= ',1PE15.8)
250 FORMAT(6(//),10X,'LOS VALORES DISCRETIZADOS DE LA FUNCION SON',//,
110X,'I',15X,'Y(I)',//)
300 FORMAT(7,8X,15,7X,1PE15.8)
350 FORMAT(6(//),10X,'EL VALOR DE LA INTEGRAL EN EL INTERVALO ES= ',1PE
115.8)
   END

```

Fig. 8.6 Listado del programa principal

```

SUBROUTINE RIEMAN(N,H,Y,SINTEG)
C-----
C----- SUBROUTINA PARA EFECTUAR INTEGRACION NUMERICA POR EL METODO TRAPE-
C----- ZOIDAL, SIMPSON DE 1/3, SIMPSON DE 3/8
C-----
C----- SIGNIFICADO DE LAS VARIABLES EMPLEADAS
C----- N=CANTIDAD DE PUNTOS EN QUE SE DISCRETIZA LA FUNCION POR INTEGRAR
C----- H=ESPACIAMIENTO ENTRE LAS ASCISAS DE LOS PUNTOS MUESTRALES
C----- Y=VALORES DISCRETIZADOS DE LA FUNCION
C----- SINTEG=INTEGRAL DE LA FUNCION
C----- AREA1=INTEGRAL POR EL METODO TRAPEZOIDAL
C----- AREA2=INTEGRAL POR EL METODO DE SIMPSON DE 1/3
C----- AREA3=INTEGRAL POR EL METODO DE SIMPSON DE 3/8
C----- L=CONTADOR DE PUNTOS MUESTRALES YA UTILIZADOS PARA EVALUAR LA IN-
C----- TEGRAL
C----- SUMP=SUMATORIA DE ORDENAS DE INDICE PAR
C----- SUMP3=SUMATORIA DE ORDENADAS CON INDICE MULTIPLO DE 3 + UNO
C----- SUMPES=SUMATORIA DEL RESTO DE ORDENADAS
C----- DIMENSION Y(101),L(101)
-----
AREA1=0.0
AREA2=0.0
AREA3=0.0
C----- IMPAGAR LA CANTIDAD DE PUNTOS MUESTRALES PARA SELECCIONAR EL METO-
C----- DO DE INTEGRACION
IF(N.EQ.1) GO TO 17
IF(N.EQ.2) GO TO 1
M=(N-1)/2
IF(2*M.EQ.(N-1)) GO TO 2
M=(N-1)/3
IF(3*M.EQ.(N-1)) GO TO 3
C----- INTEGRACION POR EL METODO TRAPEZOIDAL
1 AREA2=(H*(Y(1)+Y(2)))/2.0
GO TO 17
C----- INTEGRACION POR EL METODO DE SIMPSON DE 1/3
2 SUMP=0.0
IF(N.EQ.3) GO TO 4
DO 3 I=3,N-2,2
3 SUMP=SUMP + Y(I)
4 SUMPES=0.0
DO 5 I=2,N-1,2
5 SUMPES=SUMPES + Y(I)
ARF2=(H*(Y(1) + Y(N) + 2.0*SUMP + 4.0*SUMPES))/3.0
GO TO 17
C----- INTEGRACION POR EL METODO DE SIMPSON DE 3/8
6 SUMT=0.0
DO 7 I=1,N
7 L(I)=0
IF(N.EQ.4) GO TO 10
DO 8 I=4,N-3,3
SUMT=SUMT + Y(I)
8 L(I)=I
SUMPES=0.0
DO 9 I=2,N-1
IF(1.EQ.L(I)) GO TO 9
SUMPES=SUMPES + Y(I)
9 CONTINUE
GO TO 12
10 SUMPES=0.0
DO 11 I=2,N-1
11 SUMPES=SUMPES + Y(I)
12 AREA3=(3.0*H*(Y(1) + Y(N) + 2.0*SUMT + 3.0*SUMPES))/8.0
GO TO 17
C----- INTEGRACION COMBINANDO LOS METODOS DE SIMPSON DE 1/3 Y DE 3/8
13 NN=N-3
SUMP=0.0
IF(N.EQ.3) GO TO 15
DO 14 I=3,NN-2,2
14 SUMP=SUMP + Y(I)
15 SUMPES=0.0
DO 16 I=2,NN-1,2
16 SUMPES=SUMPES + Y(I)
AREA2=(H*(Y(1) + Y(NN) + 2.0*SUMP + 4.0*SUMPES))/3.0
AREA3=(3.0*H*(Y(NN-3) + Y(NN) + 3.0*(Y(NN-2) + Y(NN-1))))/8.0
C----- SUMAR TODAS LAS AREAS PARCIALES
17 SINTEG=AREA1 + AREA2 + AREA3
RETURN

```

Fig. 8.7 Listado de la subrutina RIEMAN

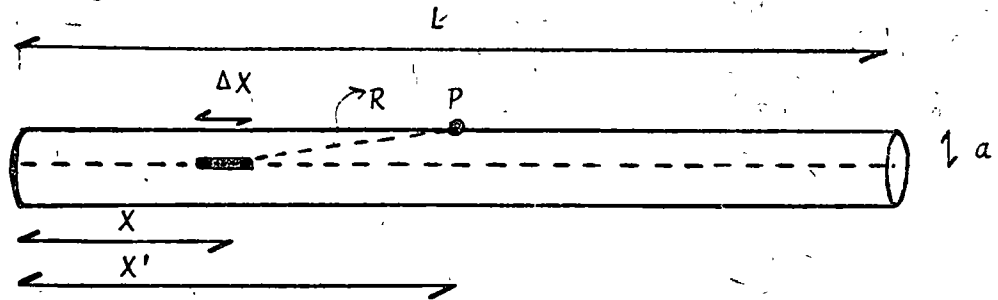
8.2.4 Ejemplo.

El potencial debido a una densidad de carga homogénea en una región está dado por:

$$v = \frac{1}{4\pi\epsilon} \sum_{i=1}^n \frac{\rho_i \Delta V_i}{R_i}$$

$\left\{ \begin{array}{l} \epsilon : \text{permitividad del medio} \\ \rho : \text{densidad volumétrica de carga} \\ \Delta V : \text{diferencial de volumen} \\ R_i : \text{distancia de la carga al punto considerado} \end{array} \right.$

Si se tiene una varilla larga con distribución superficial de carga ρ_L coulombs/m como se muestra:



la contribución de potencial en el punto X' debido a ΔX será:

$$v = \frac{\rho_L \Delta X}{4\pi\epsilon \sqrt{(X' - X)^2 + a^2}}$$

el potencial total debido a toda la carga es:

$$v_{X'} = \frac{\rho_L}{4\pi\epsilon} \int_0^L \frac{dx}{\sqrt{(X' - x)^2 + a^2}}$$

Determine numéricamente el valor del potencial para $X' = \frac{L}{2}$

si se sabe que:

$$L = 10 \text{ m}$$

$$a = 0.01 \text{ m}$$

$$\rho_L = 10^{-9} \text{ coul/m}$$

$$\epsilon = 8.854 \times 10^{-12} \text{ F/m}$$

Fraccionar el intervalo de integración en 20 partes.

*SOLUCION

TABLA 8.1 Datos para el problema del ejemplo 8.2.4

$N=21$

$H=0.5$

I	X(I)	Y(I)
1	0.0	1.798
2	0.5	1.997
3	1.0	2.247
4	1.5	2.568
5	2.0	2.996
6	2.5	3.595
7	3.0	4.493
8	3.5	5.991
9	4.0	8.987
10	4.5	17.972
11	5.0	898.774
12	5.5	17.972
13	6.0	8.987
14	6.5	5.991
15	7.0	4.493
16	7.5	3.595
17	8.0	2.996
18	8.5	2.568
19	9.0	2.247
20	9.5	1.997
21	10.0	1.798

TABLA 8.2 Resultados del problema del ejemplo 8.2.4

CANTIDAD DE PUNTOS MUESTRALES= 21

ESPACIAMIENTO ENTRE ABCISAS= 5.0000000E+01

LOS VALORES DISCRETIZADOS DE LA FUNCION SON

I	Y(I)
1	1.79754500E+00
2	1.99727100E+00
3	2.24692900E+00
4	2.56791500E+00
5	2.99589700E+00
6	3.59506800E+00
7	4.49381500E+00
8	5.99169500E+00
9	8.98729000E+00
10	1.79718900E+01
11	8.98774200E+02
12	1.79718900E+01
13	8.98729000E+00
14	5.99169500E+00
15	4.49381500E+00
16	3.59506800E+00
17	2.99589700E+00
18	2.56791500E+00
19	2.24692900E+00
20	1.99727100E+00
21	1.79754480E+00

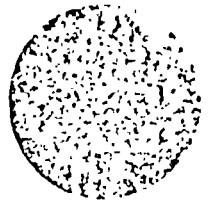
EL VALOR DE LA INTEGRAL EN EL INTERVALO ES= 3.55504987E+02

8.3 Bibliografía

1. JAMES M., SMITH G., WOLFORD J., "Applied Numerical Methods for Digital Computation with FORTRAN". Scranton Penn.: International Textbook Co., 1967. pp.272-290.
2. KUO S. Shan , "Computer Applications of Numerical Methods". Reading Mass.: Addison-Wesley Co., 1972. pp.274-312.
3. OLIVERA S. Antonio, "Apuntes de Métodos Numéricos". México: Fac. de Ingeniería UNAM, 1972. pp.5.31-5.41.



centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



METODOS NUMERICOS Y APLICACIONES CON LA COMPUTADORA
DIGITAL

TEMA 7: SOLUCION DE ECUACIONES DIFERENCIALES.

ABRIL, 1978,

9. SOLUCION DE ECUACIONES DIFERENCIALES ORDINARIAS

9.1 Introducción.

La formulación y el planteamiento matemático de una gran cantidad de problemas ingenieriles, especialmente de sistemas dinámicos, conduce a la obtención de ecuaciones diferenciales que pueden ser de tipo ordinario o parcial.

Ecuaciones diferenciales ordinarias son aquellas en las que la variable dependiente solo es función de una variable independiente, por ejemplo:

$$\left. \begin{aligned} y &= f(x) \\ z &= g(t) \end{aligned} \right\} \quad (9.1)$$

En muchas ocasiones la solución exacta de las ecuaciones diferenciales no existe, o es muy complicado el obtenerla, de ahí la necesidad de contar con métodos de tipo numérico que ofrezcan un camino alternativo de solución.

Para una ecuación diferencial ordinaria se definen los siguientes conceptos: orden y grado.

El orden está dado por la mayor derivada de la variable dependiente que se presente en la ecuación diferencial; el grado es la mayor potencia a la cual está elevada la variable dependiente o alguna de sus derivadas en la ecuación diferencial.

La forma más general de una ecuación diferencial ordinaria de orden "n" es:

$$y^{(n)} = f(x, y, y', y'', \dots, y^{(n-1)}) \quad * \quad (9.2)$$

Otra clasificación importante de las ecuaciones diferenciales ordinarias es en base a la linealidad de las mismas. Una ecuación diferencial ordinaria es lineal si la ecuación diferencial se puede expresar como una combinación lineal de la variable dependiente y de todas sus derivadas.

* $y^{(n)} = \frac{d^n y}{dx^n}$

Para que la solución de una ecuación diferencial ordinaria sea única se requiere especificar tantas condiciones iniciales o valores en la frontera como el orden de la ecuación diferencial. De acuerdo con lo anterior se clasifican los problemas ingenieriles que involucran ecuaciones diferenciales ordinarias en dos tipos: problemas con valores iniciales y problemas con valores en la frontera.

Un problema de valores iniciales se caracteriza porque toda la información concerniente al problema se especifica en un solo punto. Un problema con valores en la frontera es aquél para el cual toda la información se especifica en dos o más puntos diferentes.

Este capítulo se enfoca a obtener la solución de ecuaciones diferenciales ordinarias de primer orden mediante diversos métodos numéricos. No se tratan ecuaciones de mayor orden, dado que cualquier ecuación diferencial de orden "n" se puede descomponer en un sistema de "n" ecuaciones diferenciales de primer orden como se verá en el capítulo 10. Para problemas con valores en la frontera solo se menciona el método de diferencias finitas.

9.2 Solución General de una Ecuación Diferencial Ordinaria, Lineal y Homogénea

9.2.1 Objeto

Obtener la solución general de una ecuación diferencial lineal ordinaria del siguiente tipo:

$$y^{(n)} + a_{n-1}y^{(n-1)} + \dots + a_1y' + a_0y = 0 \quad (9.3)$$

$$y = y(x)$$

9.2.2 Método

La solución de una ecuación diferencial ordinaria, lineal y homogénea está en función de las raíces del polinomio característico de dicha ecuación, el cual es:

$$S^n + a_{n-1}S^{n-1} + \dots + a_0 = 0 \quad (9.4)$$

Las raíces de dicho polinomio pueden englobarse en los siguientes tipos:

- a) raíces reales diferentes,
- b) raíces reales repetidas,
- c) raíces complejas diferentes,
- d) raíces complejas repetidas.

La solución correspondiente a cada uno de estos cuatro tipos de raíces se menciona a continuación.

Raíces Reales Diferentes

En este caso si existen "m" raíces reales diferentes S_1, S_2, \dots, S_m ; la solución general correspondiente es:

$$y = B_1 e^{S_1 t} + B_2 e^{S_2 t} + \dots + B_m e^{S_m t} \quad (9.5)$$

Raíces Reales Repetidas

Si existe una raíz S_i repetida "m" veces, la solución general para dicha raíz es:

$$y = (B_1 + B_2 t + \dots + B_m t^{m-1}) e^{S_i t} \quad (9.6)$$

Raíces Complejas Diferentes

Las raíces complejas siempre aparecen por pares conjugados, es decir, si $a + jw$ es raíz de la ecuación característica también debe serlo $a - jw$ y la contribución de cada par conjugado a la solución general será:

$$y = e^{at} (B_1 \cos(wt) + B_2 \sin(wt)) \quad (9.7)$$

Raíces Complejas Repetidas

Si el par conjugado de raíces complejas $a \pm jw$ aparece repetido "m" veces, su contribución a la solución general de la ecuación diferencial es:

$$y = e^{at} ((B_1 + B_2 t + \dots + B_m t^{m-1}) \cos(wt) + (B_{m+1} + B_{m+2} t + \dots + B_{2m} t^{m-1}) \sin(wt)) \quad (9.8)$$

Para la obtención de la solución general mediante una computadora digital los pasos a seguir son:

- ① Leer coeficientes de la ecuación diferencial normalizados con respecto al coeficiente del término que da el orden de la ecuación diferencial
- ② Obtener las raíces del polinomio característico mediante el método de Lin-Bairstow
- ③ Contar las raíces repetidas de cada tipo
- ④ Dar la contribución de cada raíz a la solución general de acuerdo a su tipo y cantidad de veces que aparece repetida.

9.2.3 Descripción del Programa

a) Subrutinas requeridas:

SUBROUTINE RIR00, obtiene las raíces del polinomio característico mediante el método de Lin-Bairstow. Consultar el capítulo 4.

b) Descripción de las variables:

Para el programa principal:

CTE(I)	constantes arbitrarias de la solución general
X(I)	caracter alfanumérico igual a X^i , donde X es la variable independiente
NC	orden de la ecuación diferencial
A(I)	coeficientes de la ecuación diferencial
LMAX	máximo número de iteraciones a efectuar para encontrar las raíces de la ecuación característica auxiliar
RZERO y SZERO	valores de arranque para la búsqueda de las raíces
EPS	criterio de convergencia para el método de Lin-Bairstow
CEREQ	criterio para redondeamiento de raíces
NUM	variable que indica si se encontraron todas las raíces
NIM(I)	identificador de raíces complejas
NCON	contador de raíces repetidas
L	contador de raíces repetidas de un solo tipo

LEPER(I) contador que identifica a las raíces iguales de un solo tipo
 RARE(I) parte real de raíces reordenadas
 RAIM(I) parte imaginaria de raíces reordenadas
 RTREA(I) parte real de raíces sin reordenar
 RTIMA(I) parte imaginaria de raíces sin reordenar

c) Dimensiones:

Las proposiciones DIMENSION, COMMON y DATA deberán modificarse en el caso de que el orden de la ecuación diferencial sea mayor que 20.

d) Formatos para los datos de entrada:

SEC. TARJETAS	FORMATO	INFORMACION
1	(I2)	NC
2	(8F10.0)	A(I), el coeficiente del término que da el orden de la ecuación debe ser unitario y no se proporciona.
.		
.		
.		

 otros paquetes de datos (opcional)

n

TARJETA EN BLANCO, al finalizar toda la información.

e) Diagrama de bloques:

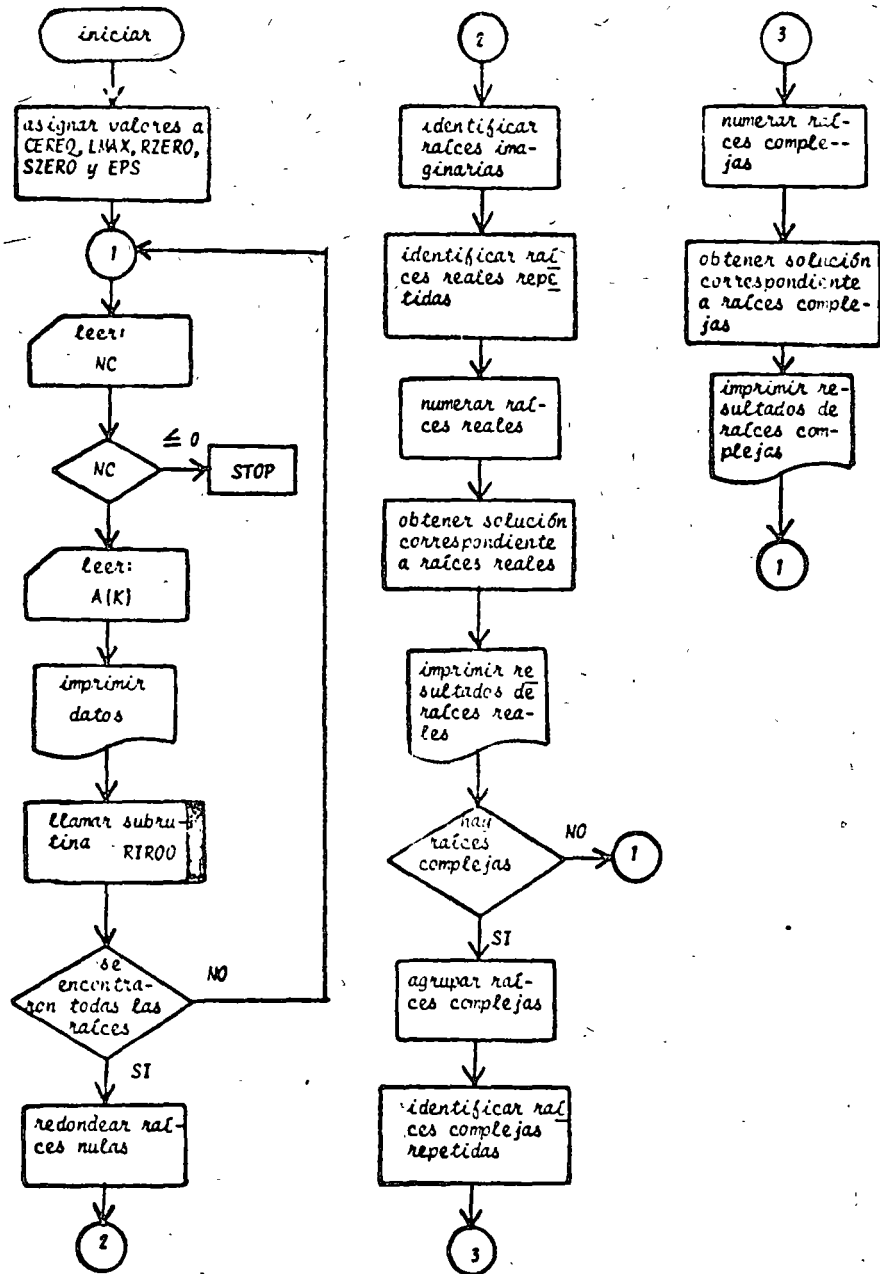


Fig. 9.1 Diagrama de bloques para el programa principal


```

C  NÚMERICACIÓN DE RAICES REALES
NPI=NCON+1
LPI=L+1
DO 7 I=NPI,LPI
8  RARE(I)=RTPEA(I)
IF(L=NCON) 101,10,101
C  IMPRESIÓN DE LOS RESULTADOS CORRESPONDIENTES A LAS RAICES REALES
101 DO 9 I=NPI,LPI
LI=L+NCON
9  WRITE(6,400) CTE(I),X(LI),PARE(I)
GO TO 11
10  WRITE(6,450) CTE(L+1),PARE(L+1)
C  INCREMENTA CONTADOR DE RAICES
11  NCON=L+1
IF(NCON=NC) 12,1,12
12  CONTINUE
C  ORDENACIÓN DE RAICES IMAGINARIAS REPETIDAS O NO
DO 13 I=1,NC
13  LEPEP(I)=0
DO 20 J=1,NC
IF(MIN(I)-J) 20,201,20
201 IF(LEPEP(I)-J) 202,20,202
202  LECON
DO 14 J=1,NC
IF(I-J) 141,14,141
141 IF(MIN(J)-I) 142,142,14
142 IF(ABS(PTPEA(I)-PTPEA(J))-CEREO) 143,143,14
143 PI=ABS(PTIMA(I))-ABS(PTIMA(J))
IF(ABS(PI)-CFREO) 144,144,14
144  LEPEP(J)=J
LEL=L+1
14  CONTINUE
C  ORDENAR RAICES IMAGINARIAS
NPI=NCON+1
LPI=L+1
DO 15 K=NPI,LPI
PARE(K)=RTPEA(I)
15  PAIM(K)=ABS(PTIMA(I))
C  IMPRESIÓN DE LOS RESULTADOS CORRESPONDIENTES A LAS RAICES IMAGINA-
C  RIAS
IF(L=NPI) 101,10,101
101 DO 17 K=NPI,LPI,2
IF(K=NPI) 101,10,101
161 LI=L+NCON
WRITE(6,500) CTE(K),X(LI),PARE(K),PAIM(K)
WRITE(6,550) CTE(K+1),X(LI),PARE(K),PAIM(K)
GO TO 17
16  LI=L+NCON+1
WRITE(6,500) CTE(K),X(LI),PARE(K),PAIM(K)
WRITE(6,550) CTE(K+1),X(LI),PARE(K),PAIM(K)
17  CONTINUE
GO TO 17
18  WRITE(6,600) CTE(L+1),PARE(L+1),PAIM(L+1)
WRITE(6,650) CTE(L+2),PARE(L+1),PAIM(L+1)
C  INCREMENTACIÓN DEL CONTADOR DE RAICES
19  NCON=L+1
IF(NCON=NC) 2,1,20
20  CONTINUE
GO TO 1
C  FORMATOS DE LECTURA E IMPRESIÓN
100  FORMAT(I2)
150  FORMAT (F10.3)
200  FORMAT (I4,///,10X,'EL ORDEN DE LA ECUACION DIFERENCIAL ES',I5)
250  FORMAT (///,10X,'LOS COEFICIENTES DE LA ECUACION SON')
300  FORMAT(///,3X,'1,000',9(2X,F10.3))
350  FORMAT (50I,15X,'LA SOLUCIÓN ESTA DADA DE LA SIG. FORMA',///,10X,'C
LADA POR CADA UNO ES UN TERMINO DE LA SOLUCIÓN Y LA SOLUCIÓN',///,10X,'EST
2A DADA POR LA SUMA DE CADA UNO DE LOS TERMINOS',///,15X,'LA SOLUCIO
3A ES',///)
400  FORMAT (/,15X,A2,A4,'EXP(',E12.5,')')
450  FORMAT (/,15X,A2,'EXP(',E12.5,')')
500  FORMAT (/,15X,A2,A4,'EXP(',E12.5,')COS(',E12.5,')')
550  FORMAT (/,15X,A2,A4,'EXP(',E12.5,')SEN(',E12.5,')')
600  FORMAT (/,15X,A2,'EXP(',E12.5,')COS(',E12.5,')')
650  FORMAT (/,15X,A2,'EXP(',E12.5,')SEN(',E12.5,')')
END

```

Fig. 9.2 Listado del programa principal

9.2.4 Ejemplo

La ecuación diferencial que caracteriza el comportamiento de un sistema dinámico sin excitaciones externas es:

$$\frac{d^8 y}{dt^8} + 11 \frac{d^7 y}{dt^7} + 68 \frac{d^6 y}{dt^6} + 320 \frac{d^5 y}{dt^5} + 979 \frac{d^4 y}{dt^4} + 2329 \frac{d^3 y}{dt^3} + 4032 \frac{d^2 y}{dt^2} + 3060 \frac{dy}{dt} = 0$$

Obtener la solución general para dicha ecuación diferencial.

*SOLUCION

TABLA 9.1 Datos para el problema del ejemplo 9.2.4

NC= 8

K	A(K)
1	11.
2	68.
3	320.
4	979.
5	2329.
6	4032.
7	3060.
8	0.

TABLA 9.2 Resultados del problema del ejemplo 9.2.4

EL GRADO DE LA ECUACION DIFERENCIAL ES 8

LOS COEFICIENTES DE LA ECUACION SON

1.000 .110E+02 .680E+02 .320E+03 .979E+03 .233E+04 .403E+04 .306E+04 0.

LA SOLUCION ESTA DADA DE LA SIG. FORMA
 CADA TERMINO ES UN TERMINO DE LA SOLUCION Y LA SOLUCION
 ESTA DADA POR LA SUMA DE CADA UNO DE LOS TERMINOS

LA SOLUCION ES

A EXP(0.)
 B EXP(-.20000E+01)
 C X EXP(-.20000E+01)
 D EXP(-.50000E+01)
 F EXP(0.) COS(.30000E+01)
 G EXP(0.) SEN(.30000E+01)
 H EXP(-.10000E+01) COS(.40000E+01)
 I EXP(-.10000E+01) SEN(.40000E+01)

9.3 Método de Euler y Euler Mejorado

9.3.1 Objeto

Obtener la solución de ecuaciones diferenciales ordinarias de primer orden del tipo:

$$y' = f(t, y) \quad (9.9)$$

$$y(t_0) = y_0$$

por el método de Euler y Euler mejorado. Se puede proporcionar la solución exacta de la ecuación diferencial para visualizar la exactitud del método.

9.3.2 Método

Euler

El encontrar la solución de la expresión (9.9) equivale a determinar el área bajo la curva $f(t, y)$; para tal propósito se muestrea la función a espacios equidistantes a fin de evaluar el área, como se muestra en la siguiente figura:

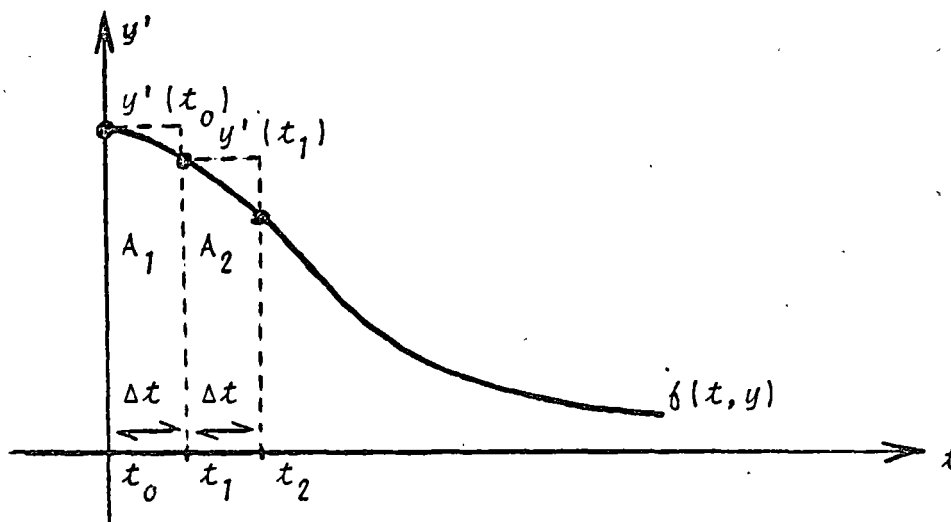


Fig. 9.3 Muestreo de la curva $f(t, y)$

Se sabe que:

$$y' = \frac{dy}{dt} \quad (9.10)$$

por lo tanto:

$$dy = y' dt \quad (9.11)$$

pasando a valores incrementales:

$$\Delta y = y' \Delta t \quad (9.12)$$

donde $y' \Delta t$ representa el área A_i , por lo que:

$$A_{i+1} = \Delta y = y_{i+1} - y_i = y' \Big|_i \Delta t \quad (9.13)$$

Reacomodando:

$$y_{i+1} = y_i + y' \Big|_i \Delta t \quad (9.14)$$

la expresión (9.14) representa la fórmula de Euler y para arrancar el método se requieren las condiciones iniciales en $t = t_0$, es decir, $y(t_0)$.

Este método es el más sencillo de todos pero tiene el inconveniente de ser el más inexacto dado que el error que produce es del orden de $(\Delta t)^2$. Para tener resultados aceptables se requiere que Δt sea pequeño.

Euler Modificado

El proceso es básicamente igual que el anterior solo que en este caso para cada y_i se efectúa una serie de iteraciones a fin de obtener su valor más exacto posible, con lo anterior se logra que el error acumulado disminuya.

El proceso se describe a continuación.

Dada la ecuación diferencial:

$$y' = f(t, y) \quad (9.15)$$

representada en la figura 9.4:

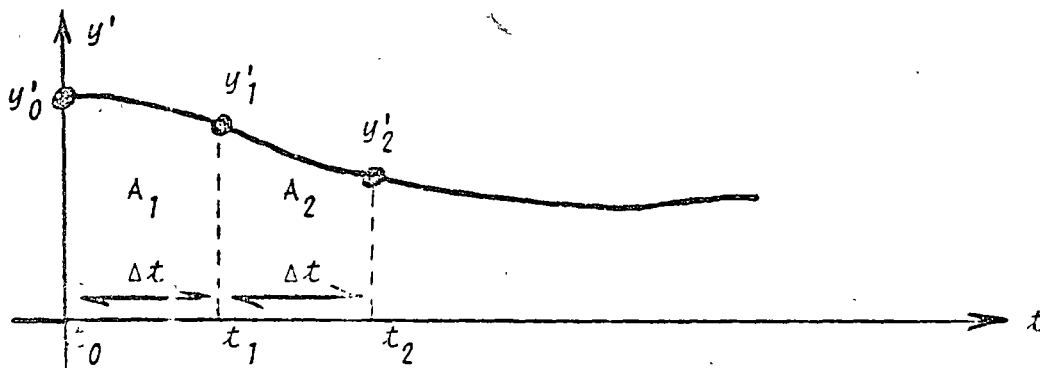


Fig. 9.4 Representación gráfica de $y' = f(t, y)$

Se sabe por el método de Euler que:

$$A_1 = y_1 - y_0 = y'_0 \Delta t \quad (9.16)$$

por lo tanto:

$$y_1 = y_0 + y'_0 \Delta t \quad (9.17)$$

y:

$$y'_1 = f(x_1, y_1) \quad (9.18)$$

dado que se conoce y_1 y y'_0 es posible aplicar el método trapezoidal de integración con lo que:

$$y_{1\text{corregida}} = y_0 + \frac{(y'_1 + y'_0)}{2} \Delta t \quad (9.19)$$

el valor de y' se corrige la cantidad de veces que sea necesaria empleando las ecuaciones (9.18) y (9.19) hasta que dos valores sucesivos corregidos difieran menos que un criterio de convergencia ϵ .

En términos generales será:

$$y_{i+1} = y_i + y'_i \Delta t \quad (9.20)$$

$$y'_{i+1} = f(x_{i+1}, y_{i+1}) \quad (9.21)$$

$$y_{i+1}^{(\text{corr.1})} = y_i + \frac{(y'_i + y'_{i+1})}{2} \Delta t \quad (9.22)$$

$$y'_{i+1}^{(\text{corr.1})} = f(x_{i+1}^{(\text{corr.1})}, y_{i+1}^{(\text{corr.1})}) \quad (9.23)$$

$$y_{i+1}^{(\text{corr.2})} = y_i + \frac{(y'_i + y'_{i+1}^{(\text{corr.2})})}{2} \Delta t \quad (9.24)$$

y así sucesivamente hasta que:

$$\left| y_{i+1}^{(\text{corr.}j)} - y_{i+1}^{(\text{corr.}j-1)} \right| < \epsilon \quad (9.25)$$

para cada y_{i+1} .

9.3.3 Descripción del Programa

a) Subrutinas requeridas:

SUBROUTINE FUNCT (C, D, F, G), en esta subrutina se proporciona la ecuación diferencial y su solución exacta en el caso de que se deseen comparar resultados.

SUBROUTINE GRAFI (A, N, M), gráfica los resultados de la ecuación diferencial dados por el método de Euler, Euler mejorado y la solución exacta en caso de haberse proporcionado. Consultar capítulo 1.

b) Descripción de las variables:

Para la subrutina FUNCT:

C valor de t_i
 D valor de y_i cuando se evalúa y_{i+1}
 G $f(t, y)$
 F expresión correspondiente a la solución exacta

Para el programa principal:

N cantidad de puntos en que se subdivide el intervalo de integración
 S espaciamiento entre puntos muestrales - (Δt)
 X(I) tiempo inicial (t_0)
 Y(I) condición inicial ($y(t_0)$)
 X(I) abscisas
 Y(I) solución obtenida por Euler
 Z(I) solución obtenida por Euler Mejorado
 W(I) solución exacta
 F valor de la solución exacta en el punto t_i
 G valor de la derivada en el punto (t_i, y_i)
 KEXAC variable que indica si se proporciona o no la solución exacta
 V variable de reemplazo
 D variable de reemplazo

EPS criterio de convergencia para el método
de Euler Mejorado

A(I, J) arreglo matricial para formar la gráfi-
ca

c) Dimensiones:

La proposición DIMENSION deberá modificarse si se
cumple que:

$$N > 100$$

d) Formatos para los datos de entrada:

SEC. TARJETAS	FORMATO	INFORMACION
1	(I5, 3F10.0)	N, S, X(1), Y(1)
2	(I1)	KEXAC, puede adquirir los siguientes valores: 1 si no se da solución -- exacta 0 cuando se da solución - exacta

otros paquetes de datos (opcional)

n

TARJETA EN BLANCO; al fi-
nalizar toda la informa-
ción.

e) Diagrama de bloques:

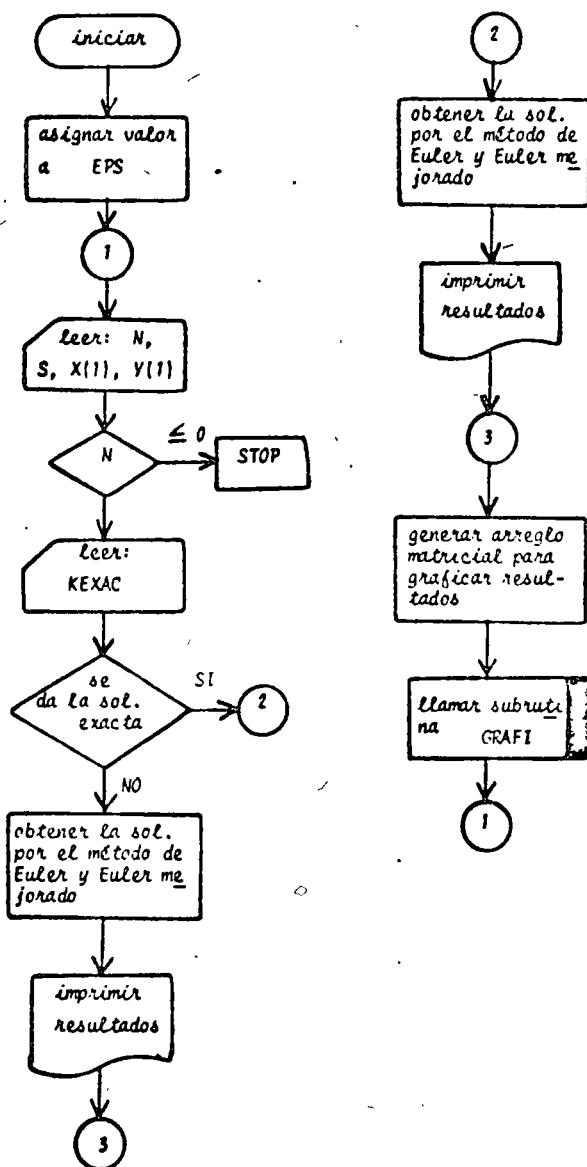


Fig. 9.5 Diagrama de bloques del programa principal

6) Listado:

```

C      PROGRAMA PARA RESOLVER ECUACIONES DIFERENCIALES POR EL METODO DE
C      EULER, EULER MEJORADO Y EXACTO(OPCIONAL)
C      SIGNIFICADO DE LAS VARIABLES EMPLEADAS
C      NECANTIDAD DE PUNTOS EN QUE SE SUBDIVIDE EL INTERVALO DE INTEGRA-
C      CION
C      S=ESPACIAMIENTO ENTRE ASCISAS
C      X(1)=VALOR INICIAL DE LA VARIABLE INDEPENDIENTE
C      Y(1)=VALOR INICIAL DE LA VARIABLE DEPENDIENTE
C      X=VALOR DE LAS ASCISAS
C      Y=SOLUCION POR EL METODO DE EULER
C      Z=SOLUCION POR EL METODO DE EULER MEJORADO
C      W=SOLUCION EXACTA
C      F=VALOR DE LA SOLUCION EXACTA EN EL PUNTO X(I)
C      G=VALOR DE LA DERIVADA EN EL PUNTO X(I)
C      EPS=CRITERIO DE CONVERGENCIA PARA EL METODO DE EULER MEJORADO
C      KEXAC=VARIABLE QUE INDICA SI SE PROPORCIONA O NO LA SOLUCION EXAC-
C      TA
-----
DIMENSION X(101),Y(101),Z(101),W(101),A(101,6)
WRITE(6,100)
-----
C      LECTURA DE DATOS
1  READ(5,200) N,S,X(1),Y(1)
-----
IF(N) 2,2,3
-----
2  CALL EXIT
-----
3  READ(5,250) KEXAC
-----
EPS=0.001
C      INDAGAR SI SE DA O NO LA SOLUCION EXACTA
IF(KEXAC) 51,5,51
-----
C      OBTENCION DE LA SOLUCION CUANDO NO SE DA SOLUCION EXACTA
51 Z(1)=Y(1)
DO 4 I=2,N
-----
X(I)=X(I-1) + S
C=Y(I-1)
D=Y(I-1)
CALL FUNCT(C,D,F,G)
Y(I)=Y(I-1) + S*D
U=Y(I)
D=Z(I-1)
CALL FUNCT(C,D,F,G)
V=Z(I-1) + S*D
G1=G
CALL FUNCT(U,V,F,G)
Z(I)=Z(I-1) + ((G1 + G)*S)/2.0
DO 20 L=1,50
-----
V=Z(I)
CALL FUNCT(U,V,F,G)
Z(I)=Z(I-1) + ((G1 + G)*S)/2.0
IF(ABS(ABS(Y)-ABS(Z(I)))-EPS) 4,4,20
-----
20 CONTINUE
4  CONTINUE
GO TO 7
-----
C      OBTENCION DE LA SOLUCION CUANDO SI SE DA SOLUCION EXACTA
5  Z(1)=Y(1)
W(1)=Y(1)
DO 6 I=2,N
-----
X(I)=X(I-1) + S
C=X(I-1)
D=Y(I-1)
CALL FUNCT(C,D,F,G)
Y(I)=D + S*D
U=Y(I)
CALL FUNCT(U,D,F,G)
W(I)=F
D=Z(I-1)
CALL FUNCT(C,D,F,G)
V=D + S*D
G1=G
CALL FUNCT(U,V,F,G)
Z(I)=D + ((G1 + G)*S)/2.0

```

```

DO 21 I=1,50
V=7(I)
CALL FUNCT(N,V,F,G)
Z(I)=D + ((G1 + G)/5)/2.0
IF (ABS(ABS(V)-ABS(Z(I)))-EPS) 6,6,21
21 CONTINUE
6 CONTINUE
C IMPRIMIR ESPACIAMIENTO USADO
7 WRITE(6,300) S
IF (NEXAC) 101,10,101
C IMPRESION DE RESULTADOS SIN SOLUCION EXACTA
101 WRITE(6,450)
DO 8 I=1,N
8 WRITE(6,550) X(I),Y(I),Z(I)
C GENERAR MATRIZ NECESARIA PARA GRAFICAR RESULTADOS SIN SOLUCION
C EXACTA
N=3
DO 9 I=1,N
A(I,1)=X(I)
A(I,2)=Y(I)
9 A(I,3)=Z(I)
GO TO 13
C IMPRESION DE RESULTADOS CON SOLUCION EXACTA
10 WRITE(6,400)
DO 11 I=1,N
11 WRITE(6,500) X(I),Y(I),Z(I),W(I)
C GENERAR MATRIZ PARA GRAFICAR RESULTADOS CON SOLUCION EXACTA
N=2
DO 12 I=1,N
A(I,1)=X(I)
A(I,2)=Y(I)
A(I,3)=Z(I)
12 A(I,4)=W(I)
C LLAMADO DE SUBROUTINA PARA GRAFICAR
13 CALL GRAF(A,N,N)
GO TO 1
C FORMATOS DE LECTURA E IMPRESION
100 FORMAT (1H1,30(/),12X,'SOLUCION DE UNA ECUACION DIFERENCIAL POR LO
15 METODOS DE',/,20X,'EULER,EULER MEJORADO Y EXACTO(OPCIONAL), CON
24R GRAFICAS CORRESPONDIENTES')
200 FORMAT (15,3F10.0)
250 FORMAT (11)
300 FORMAT (1H1,/,/,15X,'EL ESPACIAMIENTO USADO FUE ',F8.5)
400 FORMAT (///,17X,1X,20X,'EULER',12X,'EULER MEJORADO',12X,'EXACTO',
1/)
450 FORMAT (///,10X,1X,18X,'EULER',12X,'EULER MEJORADO',/)
500 FORMAT (/,12X,F10.5,10X,3(E12.5,10X))
550 FORMAT (/,13X,F10.5,10X,2(E12.5,10X))
END

```

Fig. 9.6 Listado del programa principal

```

SUBROUTINE FUNCT(C,D,F,G)
F = SQRT(4./(0.8+C + 1.))
G = 1 - 0.1*D**3 + 0.2*C
RETURN
END

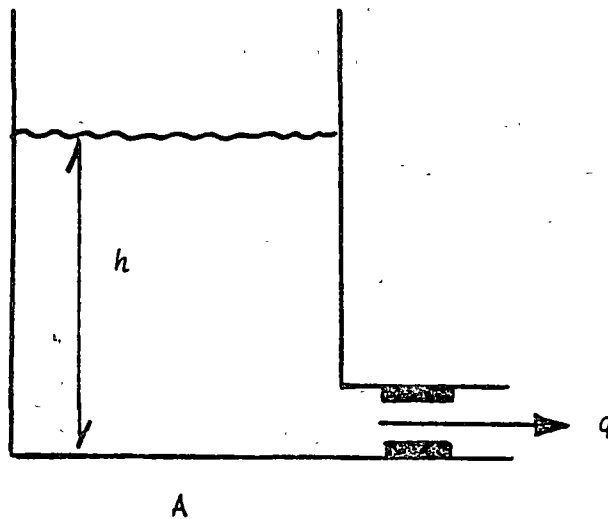
```

Fig. 9.7 Listado de la subrutina FUNCT

9.3.4 Ejemplo

La variación de la altura del sistema hidráulico de la Fig. (9.8) se encuentra caracterizada por la siguiente ecuación diferencial:

$$\frac{dh}{dt} = -\frac{K}{A} h^3$$



$$\begin{aligned} \gamma &: \text{peso específico} \\ K &= 0.1 \gamma \\ q &= Kh^2 \end{aligned}$$

Fig. 9.8 Sistema hidráulico del problema del ejemplo 9.3.4

La solución analítica de dicha ecuación diferencial es:

$$h(t) = \sqrt{\frac{A h_0^2}{2K h_0^2 t + A}}, \quad h_0 = h(t) \Big|_{t = t_0}$$

Determine la altura $h(t)$ por el método de Euler y Euler Mejorado, compare sus resultados con la solución exacta. Considérese lo siguiente:

$$\begin{aligned} h_0 &= 2 \text{ m} \\ A &= 1 \text{ m}^2 \\ K &= 0.1 \text{ 1/S} \\ t_0 &= 0 \text{ Seg} \\ \Delta t &= 0.5 \text{ Seg} \\ t_{\text{final}} &= 45 \text{ seg} \end{aligned}$$

* SOLUCION

TABLA 9.3 Datos del problema del ejemplo 9.3.4

$$N = 90$$

$$S = 0.5$$

$$X(1) = 0.$$

$$Y(1) = 2.$$

$$KEXAC = 0$$

$$F(C) = -\text{SQRT}(4./(0.8*C + 1.))$$

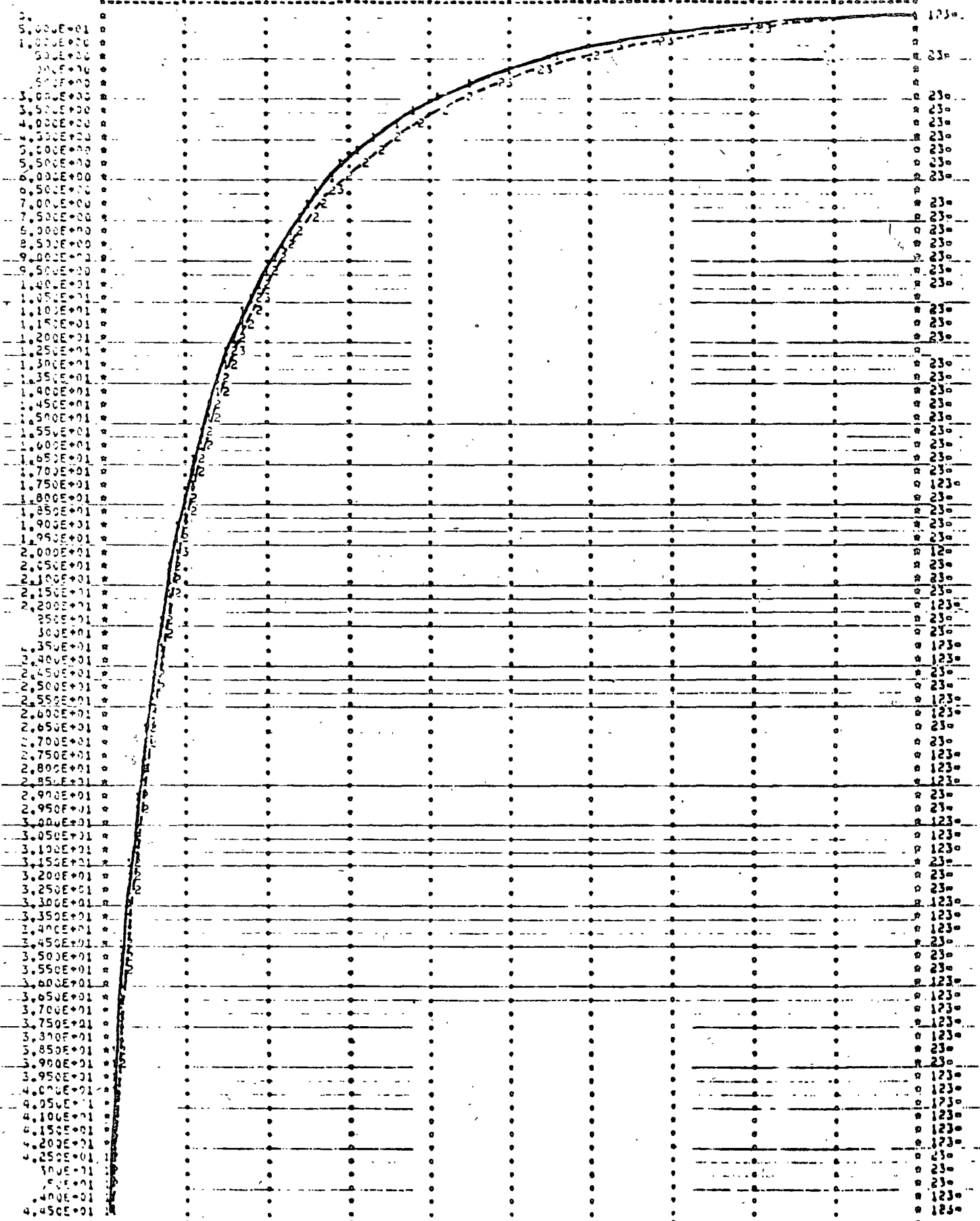
$$G(C, D) = -0.1*D**3 + 0.*C$$

TABLA 9.4 Resultados del problema del ejemplo 9.3.4

EL ESPACIAMIENTO USADO FUE 0,50000

X	EULER	FULER MEJORADO	EXACTO
0,00000	.20000E+01	.20000E+01	.20000E+01
0,50000	.16000E+01	.16810E+01	.16903E+01
1,00000	.13952E+01	.14810E+01	.14907E+01
1,50000	.12594E+01	.13398E+01	.13484E+01
2,00000	.11595E+01	.1232 ⁸ E+01	.12403E+01
2,50000	.10816E+01	.11481E+01	.11547E+01
3,00000	.10183E+01	.1078 ⁸ E+01	.10847E+01
3,50000	.96552E+00	.1020 ⁸ E+01	.10260E+01
4,00000	.92051E+00	.97126E+00	.97590E+00
4,50000	.88151E+00	.92834E+00	.93250E+00
5,00000	.84726E+00	.89066E+00	.89443E+00
5,50000	.81685E+00	.85725E+00	.86066E+00
6,00000	.78960E+00	.82733E+00	.83045E+00
6,50000	.76499E+00	.80034E+00	.80322E+00
7,00000	.74260E+00	.77586E+00	.77850E+00
7,50000	.72213E+00	.7534 ⁸ E+00	.75593E+00
8,00000	.70330E+00	.73295E+00	.73521E+00
8,50000	.68591E+00	.71400E+00	.71611E+00
9,00000	.66977E+00	.69646E+00	.69843E+00
9,50000	.65475E+00	.68015E+00	.68149E+00
10,00000	.64071E+00	.66493E+00	.66667E+00
10,50000	.62756E+00	.65069E+00	.65233E+00
11,00000	.61520E+00	.63733E+00	.63888E+00
11,50000	.60356E+00	.62476E+00	.62622E+00
12,00000	.59257E+00	.61291E+00	.61430E+00
12,50000	.58217E+00	.60171E+00	.60302E+00
•			•
•			•
43,00000	.33084E+00	.33590E+00	.33615E+00
43,50000	.32903E+00	.33402E+00	.33426E+00
44,00000	.32725E+00	.33218E+00	.33241E+00
44,50000	.32550E+00	.33036E+00	.33059E+00

1.355E+01 4.900E+01 1.600E+01 8.270E+01 9.950E+01 1.163E+02 1.330E+02 1.490E+02 1.600E+02 1.630E+02 1.650E+02 1.650E+02



9.4 Método de Runge-Kutta

9.4.1 Objeto

Obtener la solución de una ecuación diferencial de primer orden del tipo:

$$y' = f(t, y) \quad (9.26)$$

sujeta a las condiciones iniciales $y(t_0) = y_0$, por el método de Runge-Kutta con la opción de comparar la solución numérica con la solución exacta.

9.4.2 Método

El método de Runge-Kutta emplea la fórmula de recurrencia:

$$y_{i+1} = y_i + A_1 K_1 + A_2 K_2 + \dots + A_n K_n \quad (9.27)$$

para evaluar los valores sucesivos de la variable dependiente de la ecuación (9.26). Los parámetros K_j se determinan en la siguiente forma:

$$K_1 = (\Delta t) f(t_i, y_i)$$

$$K_2 = (\Delta t) f(t_i + p_1 \Delta t, y_i + q_{11} K_1)$$

$$K_3 = (\Delta t) f(t_i + p_2 \Delta t, y_i + q_{21} K_1 + q_{22} K_2)$$

⋮

$$K_n = (\Delta t) f(t_i + p_{n-1} \Delta t, y_i + q_{n-1,1} K_1 + q_{n-1,2} K_2 +$$

$$+ \dots + q_{n-1,n-1} K_{n-1}) \quad (9.28)$$

Los valores de A , p y q se obtienen al igualar la ecuación (9.27) con cierto número de términos del desarrollo por serie de Taylor de la variable y .

Dependiendo del valor de " n " en la ecuación (9.27) se habla del método de Runge-Kutta de orden " n ". El orden del error producido por el método es $(\Delta t)^{n+1}$ donde Δt es el espaciamiento entre los valores de la variable independiente.

El desarrollo mediante serie de Taylor para la variable

y_{i+1} dado que se conoce el valor y_i es:

$$y_{i+1} = y_i + (\Delta t)y'_i + \dots + \frac{(\Delta t)^n}{n!} y_i^{(n)} + \dots \quad (9.29)$$

Para el programa se emplearon fórmulas de Runge-Kutta de cuarto orden, las cuales son:

$$y_{i+1} = y_i + \frac{1}{6} (K_1 + 2K_2 + 2K_3 + K_4)$$

$$K_1 = (\Delta t) f(t_i, y_i)$$

$$K_2 = (\Delta t) f\left(t_i + \frac{\Delta t}{2}, y_i + \frac{K_1}{2}\right)$$

$$K_3 = (\Delta t) f\left(t_i + \frac{\Delta t}{2}, y_i + \frac{K_2}{2}\right)$$

$$K_4 = (\Delta t) f(t_i + \Delta t, y_i + K_3) \quad (9.30)$$

Geoméricamente los valores K_1 , K_2 , K_3 y K_4 representan las pendientes (derivadas de la curva) en diferentes puntos del intervalo (t_i, t_{i+1}) .

9.4.3 Descripción del Programa

a) Subrutinas requeridas:

SUBROUTINE FUNCT(C,D,F,G), en esta subrutina se proporciona la ecuación diferencial y su solución exacta en caso de conocerse.

SUBROUTINE GRAFI(A,N,M), grafica la solución obtenida de la ecuación diferencial por el método de Runge-Kutta y el exacto en caso de haberse proporcionado.

b) Descripción de las variables:

Para la subrutina FUNCT:

C valor de t_i

D valor de y_i

G $f(t, y)$

F solución exacta de la ecuación diferencial

Para el programa principal:

N cantidad de puntos en que se subdivide el intervalo
 S espaciamiento entre las abscisas
 X(1) valor inicial de la variable independiente
 Y(1) valor inicial de la variable dependiente
 KEXAC variable que informa si se da o no la solución exacta
 RUNGE(I) valores de la solución obtenidos por el método de Runge-Kutta
 Y(I) valores de la solución exacta
 F solución exacta de la ecuación diferencial
 G ecuación diferencial
 C variable de reemplazo
 D variable de reemplazo
 RUK(I) valor de los parámetros K_j que emplea el método
 A(I, J) arreglo matricial para trazar la gráfica

c) Dimensiones:

La proposición DIMENSION se deberá modificar cuando se presente el caso de que:

$$N > 100$$

d) Formatos para los datos de entrada:

SEC. TARJETAS	FORMATO	INFORMACION
1	(I5, 3F10.0)	N, S, X(1), Y(1)
2	(I1)	KEXAC, puede adquirir dos valores: 1 cuando no se da sol. exacta 0 cuando se da solución exacta

 otros paquetes de datos (opcional)

n

TARJETA EN BLANCO, al finalizar toda la información

e) Diagrama de bloques:

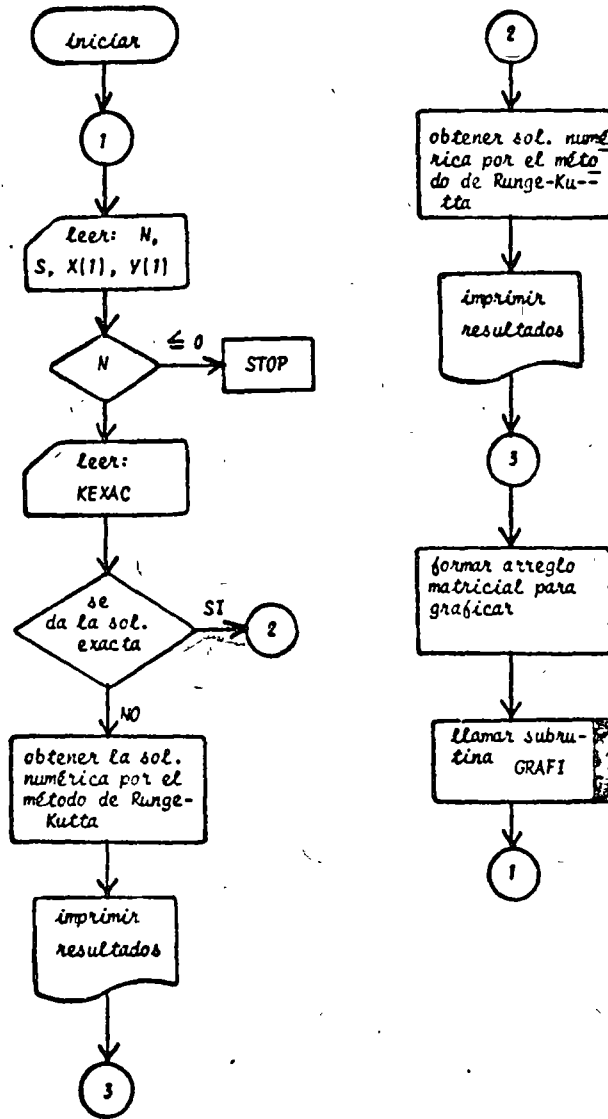


Fig. 9.9 Diagrama de bloques del programa principal

6) Listado:

```

C   PROGRAMA PARA RESOLVER ECUACIONES DIFERENCIALES POR EL METODO DE
C   RUNGE-KUTTA
C   SIGNIFICADO DE LAS VARIABLES EMPLEADAS
C   N=CANTIDAD DE PUNTOS EN QUE SE SUBDIVIDE EL INTERVALO
C   S=ESPACIAMIENTO ENTRE ABCISAS
C   X(1)=VALOR INICIAL DE LA VARIABLE INDEPENDIENTE
C   Y(1)=VALOR INICIAL DE LA VARIABLE DEPENDIENTE
C   KEXAC=PARAMETRO QUE INFORMA SI SE DA O NO LA SOLUCION EXACTA
C   RUNGE=VALORES DE LA SOLUCION POR EL METODO DE RUNGE-KUTTA
C   Y=VALORES DE LA SOLUCION EXACTA
C   F=SOLUCION EXACTA DE LA ECUACION DIFERENCIAL
C   G=ECUACION DIFERENCIAL
C   C Y D=VARIABLES DE REEMPLAZO
C   RUK=VALOR DE LAS CONSTANTES K DE LA FORMULA DE RUNGE-KUTTA

      DIMENSION X(101),Y(101),RUNGE(101),RUK(5),A(101,6)
      WRITE(4,100)
C   LECTURA DE DATOS
      1 READ(5,150) N,S,X(1),Y(1)
      IF(N)/2,2,3
      2 CALL EXIT
C   INFORMACION SOBRE SI SE DA SOLUCION EXACTA
      3 READ(5,200) KEXAC
      IF(KEXAC) 51,5,51
C   OBTENCION DE LA SOLUCION CUANDO NO SE DA SOLUCION EXACTA
      51 RUNGE(1)=Y(1)
      DO 4 I=2,N
      X(I)=X(I-1) + S
      C=X(I-1)
      D=RUNGE(I-1)
      CALL FUNCT(C,D,F,G)
      RUK(1)=G
      C=X(I-1) + 0.5*S
      D=RUNGE(I-1) + 0.5*S*RUK(1)
      CALL FUNCT(C,D,F,G)
      RUK(2)=G
      D=RUNGE(I-1) + 0.5*S*RUK(2)
      CALL FUNCT(C,D,F,G)
      RUK(3)=G
      C=X(I-1) + S
      D=RUNGE(I-1) + S*RUK(3)
      CALL FUNCT(C,D,F,G)
      RUK(4)=G
      RUNGE(I)=RUNGE(I-1) + (.S*(RUK(1) + 2.0*RUK(2) + 2.0*RUK(3) + RUK(4)
      1)))/5.0
      4 CONTINUE
      GO TO 7
C   OBTENCION DE LA SOLUCION CUANDO SI SE DA SOLUCION EXACTA
      5 RUNGE(1)=Y(1)
      DO 6 I=2,N
      X(I)=X(I-1) + S
      C=X(I-1)
      D=RUNGE(I-1)
      CALL FUNCT(C,D,F,G)
      RUK(1)=G
      C=X(I-1) + 0.5*S
      D=RUNGE(I-1) + 0.5*S*RUK(1)
      CALL FUNCT(C,D,F,G)
      RUK(2)=G
      D=RUNGE(I-1) + 0.5*S*RUK(2)
      CALL FUNCT(C,D,F,G)
      RUK(3)=G
      C=X(I-1) + S
      D=RUNGE(I-1) + S*RUK(3)

```

```

CALL FUNCT(C,D,F,G)
RUK(4)=G
RUNGE(I)=RUNGE(I-1) + (S*(PUK(I) + 2.0*RUK(2) + 2.0*RUK(3) + RUK(4
1)))/6.0
C=Y(I)
CALL FUNCT(C,D,F,G)
6 Y(I)=F
C IMPRESION DEL ESPACTAMIENTO USADO
7 WRITE(6,250) S
IF(EXAC) 121,12,121
C IMPRESION DE LOS RESULTADOS SIN SOLUCION EXACTA
111 WRITE(6,300)
DO 8 I=1,N
8 WRITE(6,350) X(I),RUNGE(I)
C GENERAR MATRIZ PARA GRAFICAR RESULTADOS SIN SOLUCION EXACTA
M=2
DO 11 I=1,N
A(I,1)=Y(I)
11 A(I,2)=RUNGE(I)
GO TO 17
C IMPRESION DE LOS RESULTADOS CON SOLUCION EXACTA
12 WRITE(6,400)
DO 13 I=1,N
13 WRITE(6,450) X(I),RUNGE(I),Y(I)
C GENERACION DE LA MATRIZ PARA GRAFICAR LOS RESULTADOS
M=3
DO 16 I=1,N
A(I,1)=Y(I)
A(I,2)=RUNGE(I)
16 A(I,3)=Y(I)
C LLAMADO DE SUBROUTINA PARA GRAFICAR
7 CALL GRAFI(N,M)
GO TO 1
C FORMATOS DE LECTURA E IMPRESION
110 FORMAT (1M1,3C(//),31X,'SOLUCION DE UNA ECUACION DIFERENCIAL POR LO
15 METODOS DE ',//,27X,'RUNGE-KUTTA Y EXACTO(OPCIONAL), CON LAS GRAFI
20AS CORRESPONDIENTES')
110 FORMAT (15, F10.0)
210 FORMAT (11)
210 FORMAT (1M1, //,15X,'EL ESPACTAMIENTO USADO FUE ',F8.5)
300 FORMAT (///,15X,'X',29X,'RUNGE-KUTTA',//)
310 FORMAT (/,10X,F10.5,20X,E15.8)
410 FORMAT (///,15X,'X',16X,'RUNGE-KUTTA',16X,'EXACTA',//)
410 FORMAT (/,10X,F10.5,10X,2(E15.8,10X))
END

```

Fig. 9.10 Listado del programa principal

```

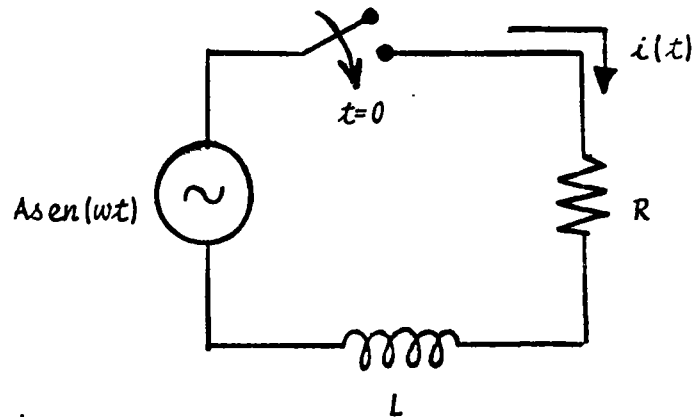
SUBROUTINE FUNCT(C,D,F,G)
F = 0.007935*(10.*SIN(380.*C) + 380.*COS(380.*C) + 380.*EXP(-10.
1*C))
G = 115.*SIN(380.*C) - 10.*D
RETURN
END

```

Fig. 9.11 Listado de la subrutina FUNCT.

9.4.4 Ejemplo

Para el circuito eléctrico que se muestra a continuación:



la ecuación diferencial que caracteriza el comportamiento de la corriente para $t \geq 0$ es:

$$\frac{di}{dt} = \frac{A \text{ sen}(wt)}{L} - \frac{R}{L} i$$

La solución analítica de la ecuación diferencial es:

$$i(t) = \frac{A}{R^2 + w^2 L^2} (R \text{ sen}(wt) - wL \text{ cos}(wt) + wL e^{-(R/L)t})$$

Obtenga la solución numérica de la ecuación diferencial y compare los resultados con la solución analítica para los siguientes valores:

$$A = 115 \text{ V}$$

$$L = 1 \text{ H}$$

$$R = 10 \ \Omega$$

$$w = 380 \text{ rad/s}$$

$$i_0 = 0 \text{ A}$$

$$t_0 = 0 \text{ s}$$

$$t_f = 1 \text{ s}$$

$$\Delta t = .01 \text{ s}$$

* SOLUCION

TABLA 9.5 Datos para el problema del ejemplo 9.4.4

$$N=100$$

$$S=0.01$$

$$X(1)=0.$$

$$Y(1)=0.$$

$$KEXAC=0$$

$$F(C)=0.0007935*(10.*\text{SIN}(380.*C) - 380.*\text{COS}(380.*C) + 380.*\text{EXP}(-10.*C))$$

$$G(C,D)=115.*\text{SIN}(380.*C) - 10.*D$$

TABLA 9.6 Resultados del problema del ejemplo 9.4.4

EL ESPACIAMIENTO USA00 FUE 0.01000

X	RUNGE-KUTTA	EXACTA
0.00000	0.	0.
0.01000	.57285601E+00	.50648103E+00
0.02000	.19614662E+00	.17878995E+00
0.03000	.11431645E+00	.97434745E-01
0.04000	.52555474E+00	.46943909E+00
0.05000	-.12905971E+00	-.11404764E+00
0.06000	.41754259E+00	.36789741E+00
0.07000	.13715049E+00	.12646394E+00
0.08000	-.29513185E-01	-.30114185E-01
0.09000	.45695747E+00	.40783343E+00
0.10000	-.19803788E+00	-.17470498E+00
0.11000	.30451534E+00	.26698100E+00
0.12000	.12185311E+00	.11288390E+00
0.13000	-.13033234E+00	-.11933897E+00
0.14000	.41635141E+00	.37107606E+00
0.15000	-.22787807E+00	-.20059544E+00
0.16000	.21597789E+00	.18792052E+00
0.17000	.13218685E+00	.12198715E+00
0.18000	-.20252440E+00	-.18303566E+00
0.19000	.38957842E+00	.34659517E+00
0.20000	-.23154303E+00	-.20326091E+00
0.21000	.14159762E+00	.12152925E+00
0.22000	.15675009E+00	.14363945E+00
0.23000	-.25456386E+00	-.22874903E+00
0.24000	.36809957E+00	.32681498E+00
0.25000	-.21722354E+00	-.18999646E+00
.	.	.
.	.	.
.	.	.
0.86000	-.33789921E+00	-.30005160E+00
0.87000	.24963596E+00	.21897656E+00
0.88000	-.56809302E-01	-.46016549E-01
0.89000	-.15958288E+00	-.14591809E+00
0.90000	.30942578E+00	.27699727E+00
0.91000	-.32975936E+00	-.29214004E+00
0.92000	.21236671E+00	.18527035E+00
0.93000	-.60689543E-02	-.83625394E-03
0.94000	-.20265463E+00	-.18384843E+00
0.95000	.32675628E+00	.29176219E+00
0.96000	-.31416151E+00	-.27761945E+00
0.97000	.17030745E+00	.14748720E+00
0.98000	.44817611E-01	.40370592E-01
0.99000	-.24114046E+00	-.21761855E+00

9.5 Método de Milne

9.5.1 Objeto

Obtener la solución de una ecuación diferencial de primer orden del tipo:

$$y' = f(t, y) \quad (9.31)$$

sujeta a las condiciones iniciales $y(t_0) = y_0$, mediante el método de Milne; con la opción de comparar los resultados numéricos con los resultados analíticos.

9.5.2 Método

El método de Milne subdivide cada subintervalo de integración en cinco puntos igualmente espaciados y aproxima la curva (ecuación diferencial) mediante una parábola de segundo grado que pasa por tres de esos puntos muestrales y el área en cada subintervalo se aproxima por el área debajo de la parábola. El área total es igual a la suma de las áreas de cada subintervalo. Gráficamente se tendrá:

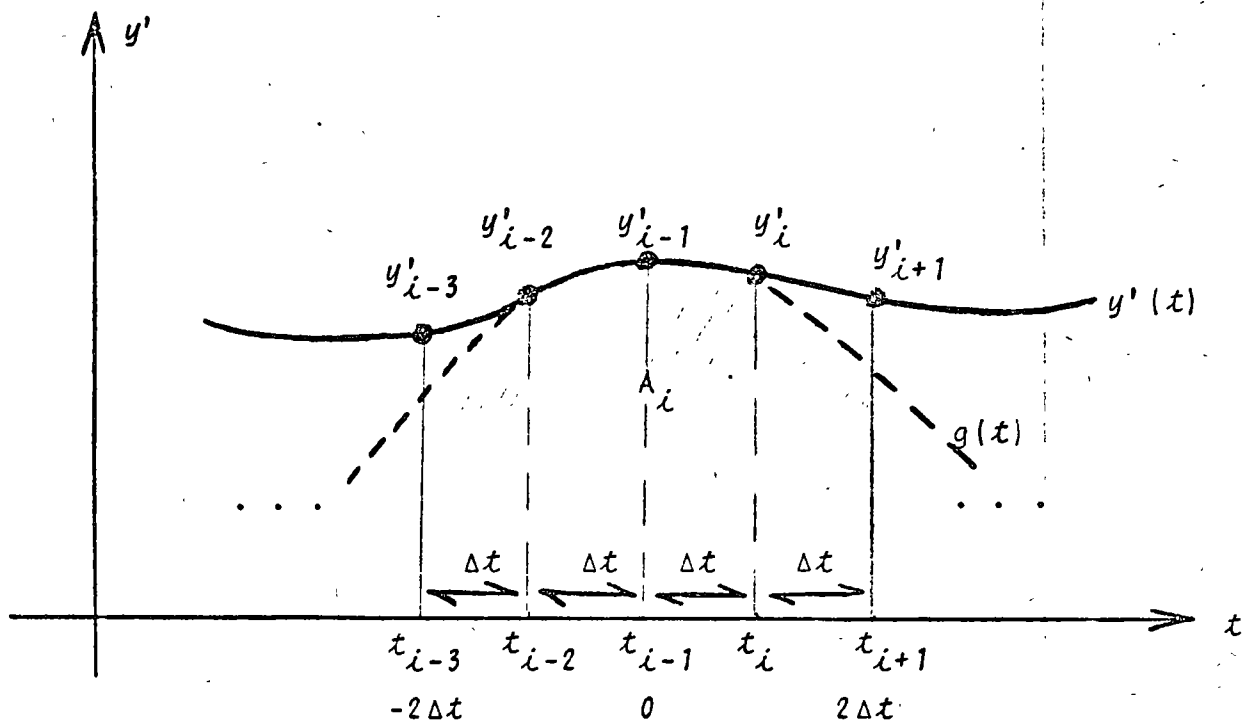


Fig. 9.12 Representación gráfica del muestreo para el método de Milne

Por lo tanto:

$$y'(t) \doteq g(t) \quad (9.32)$$

$$A_i = \int_{t_{i-3}}^{t_{i+1}} g(t) dt \quad (9.33)$$

$$g(t) = at^2 + bt + c \quad (9.34)$$

substituyendo (9.34) en (9.33):

$$A_i = \int_{t_{i-3}}^{t_{i+1}} (at^2 + bt + c) dt \quad (9.35)$$

haciendo el siguiente cambio en los límites de integración:

$$t_{i+1} = 2 \Delta t \quad (9.36)$$

$$t_{i-3} = -2 \Delta t$$

se obtiene:

$$A_i = \frac{16}{3} a (\Delta t)^3 + 4c (\Delta t) \quad (9.37)$$

para evaluar a , b y c se obliga a que la curva (9.34) pase por los puntos y'_{i-2} , y'_{i-1} , y'_i ; al efectuar lo anterior se obtiene:

$$a = \frac{y'_i - 2y'_{i-1} + y'_{i-2}}{2(\Delta t)^2} \quad (9.38)$$

substituyendo (9.38) en (9.37):

$$A_i = \frac{4}{3} (\Delta t) (2y'_i - y'_{i-1} + 2y'_{i-2}) \quad (9.39)$$

De la gráfica se puede observar que:

$$y_{i+1} = y_{i-3} + A_i \quad (9.40)$$

$$y_{i+1} = y_{i-3} + \frac{4}{3} (\Delta t) (2y'_i - y'_{i-1} + 2y'_{i-2}) \quad (9.41)$$

La ecuación (9.41) da la primera aproximación de la solución por el método de Milne. El valor obtenido se mejora empleando un procedimiento similar al del método de Euler mejorado; en este caso se emplea la fórmula de Simpson de 1/3 para efectuar la corrección utilizando los puntos correspondientes a t_{i-1} , t_i , t_{i+1} . El proceso se repite para cada y_i hasta que dos correcciones sucesivas de la variable dependiente sean aproximadamente iguales; en términos numéricos se tendrá:

$$y_{i+1}^{(pron.)} = y_{i-3} + \frac{4}{3} (\Delta t) (2y'_i - y'_{i-1} + 2y'_{i-2}) \quad (9.42)$$

$$y'_{i+1}^{(pron.)} = f(t_{i+1}, y_{i+1}^{(pron.)}) \quad (9.43)$$

$$y_{i+1}^{(corr.1)} = y_{i-1} + \frac{\Delta t}{3} (y'_{i-1} + 4y'_i + y'_{i+1}^{(pron.)}) \quad (9.44)$$

$$y'_{i+1}^{(corr.1)} = f(t_{i+1}, y_{i+1}^{(corr.1)}) \quad (9.45)$$

así sucesivamente hasta que:

$$\left| y_{i+1}^{(corr.j)} - y_{i+1}^{(corr.j-1)} \right| < \epsilon \quad (9.46)$$

El error producido por el método es del orden de $(\Delta t)^5$, igual que el producido por el método de Runge-Kutta pero con la ventaja de ser mucho más rápido. Su desventaja como se puede apreciar en la relación (9.42) es que para arrancar requiere que se conozcan los valores y_0 , y'_1 , y'_2 , y'_3 . Estos valores se pueden obtener mediante expansiones de la serie de Taylor o empleando el método de Runge-Kutta para el arranque, esto último es lo más usual y será el método empleado en el programa.

9.5.3 Descripción del Programa

a) Subrutinas requeridas:

SUBROUTINE FUNCT(C,D,F,G), en esta subrutina se propor-

cionan la ecuación diferencial y su solución exacta en caso de conocerse.

SUBROUTINE GRAFI(A,N,M), grafica la solución obtenida de la ecuación diferencial por el método de Milne y la solución exacta en caso de haberse proporcionado.

b) Descripción de las variables:

Para la subrutina FUNCT:

C valor de t_i
 D valor de y_i
 G es la ecuación diferencial $f(t, y)$
 F solución exacta de la ecuación diferencial

Para el programa principal:

N cantidad de puntos en que se subdivide el intervalo total de integración
 S espaciamiento entre los valores de la variable independiente
 X(1) valor inicial de la variable independiente
 Y(1) valor inicial de la variable dependiente
 X(I) valores de la variable independiente
 Y(I) valores de la solución exacta
 RILNE(I) solución obtenida por el método de Milne
 RUK(I) parámetros del método de Runge-Kutta
 PRE(I) pronóstico de y_i
 DIF(I) valor de y'_i
 CORR(I) corrección de y_i
 KEXAC variable que indica si se da o no la solución exacta
 C variable de reemplazo
 D variable de reemplazo
 G1 variable de reemplazo
 G2 variable de reemplazo
 A(I, J) arreglo matricial para imprimir la gráfica

c) Dimensiones:

La proposición DIMENSION deberá ser modificada cuando:
 $N > 100$

d) Formatos para los datos de entrada:

SEC. TARJETAS	FORMATO	INFORMACION
1.	(I5,3F10.0)	N,S,X(1),Y(1)
2	(I1)	KEXAC, puede adquirir alguno de los dos valores siguientes: 1 cuando no se da sol. exacta 0 cuando se da la sol. exacta

 otros paquetes de datos (opcional)

n

TARJETA EN BLANCO, al finalizar toda la información.

e) Diagrama de bloques:

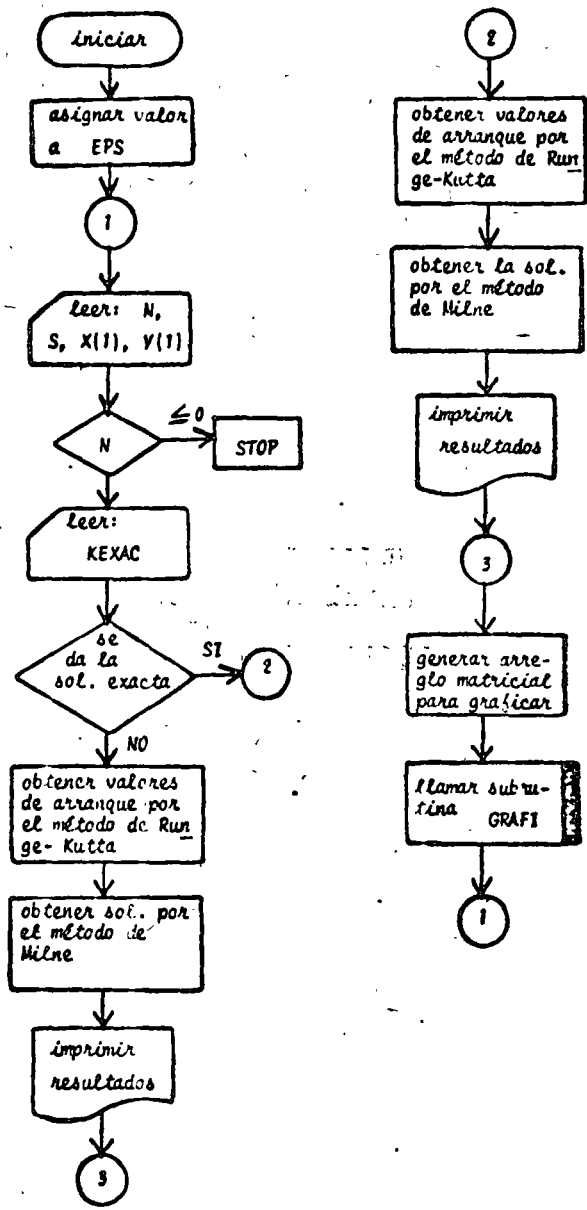


Fig. 9.13 Diagrama de bloques para el programa principal

6) Listado:

```

C PROGRAM PARA RESOLVER ECUACIONES DIFERENCIALES POR EL METODO DE
C MILNE
C SIGNIFICADO DE LAS VARIABLES EMPLEADAS
C N=CANTIDAD DE PUNTOS EN QUE SE SUBDIVIDE EL INTERVALO DE INTEGRACION
C S=ESPACIAMIENTO ENTRE LOS VALORES DE LA VARIABLE INDEPENDIENTE
C X(1)=VALOR INICIAL DE LA VARIABLE INDEPENDIENTE
C Y(1)=VALOR INICIAL DE LA VARIABLE DEPENDIENTE
C X=VALORES DE LA VARIABLE INDEPENDIENTE
C Y=VALORES DE LA SOLUCION POR EL METODO ANALITICO
C MILNE=SOLUCION OBTENIDA POR EL METODO DE MILNE
C PUK=PARAMETROS DEL METODO DE RUNGE-KUTTA
C PRF, DIF Y CORP=CORRECTORES DEL METODO DE MILNE
C KEYAC=VARIABLE QUE INDICA SI SE DA O NO LA SOL. EXACTA
C C, D, G1 Y G2=VARIABLES DE REEMPLAZO
C EPS=CRITERIO DE CONVERGENCIA

DIMENSION Y(100),Y(100),PRE(100),DIF(100),CORR(100),RUK(4),MILNE(1
100),A(101,6)
WRITE(6,100)
C LECTURA DE DATOS
EPS=0.001
1 READ(5,150) N,S,X(1),Y(1)
IF(N) 2,P,3
2 CALL EXIT
C INFORMACION SOBRE SI SE DA SOLUCION EXACTA
3 MILNE(1)=Y(1)
READ(5,200) KEYAC
IF(KEYAC) 71,7,71
C OBTENCION DE LA SOLUCION CUANDO NO SE DA SOL. EXACTA
71 DO 4 I=2,N
C OBTENER LA SOLUCION MEDIANTE EL METODO DE RUNGE-KUTTA
X(I)=X(I-1) + S
C=X(I-1)
D=MILNE(I-1)
CALL FUNCT(C,D,F,G)
RUK(1)=G
C=X(I-1) + 0.5*S
D=MILNE(I-1) + 0.5*S*RUK(1)
CALL FUNCT(C,D,F,G)
RUK(2)=G
D=MILNE(I-1) + 0.5*S*RUK(2)
CALL FUNCT(C,D,F,G)
RUK(3)=G
C=X(I-1) + S
D=MILNE(I-1) + S*RUK(3)
CALL FUNCT(C,D,F,G)
RUK(4)=G
MILNE(I)=MILNE(I-1) + (S*(RUK(1) + 2.0*RUK(2) + 2.0*PUK(3) + RUK(4
))) / 6.0
4 CONTINUE
C CONTINUAR LA SOLUCION CON EL METODO DE MILNE
DO 6 I=5,N
X(I)=X(I-1)+S
C=X(I-1)
D=MILNE(I-1)
CALL FUNCT(C,D,F,G)
G1=G
C=X(I-2)
D=MILNE(I-2)
CALL FUNCT(C,D,F,G)
G2=G
C=X(I-3)
D=MILNE(I-3)
CALL FUNCT(C,D,F,G)
PRE(I)=MILNE(I-4) + (4.0*S*(2.0*G1 + G2 + 2.0*G))/3.0
C=X(I)
D=PRE(I)

```

```

CALL FUNCT(C,D,F,G)
DIF(I)=C
CORR(I)=PILNE(I-2) + (S*(G2 + 4.0*G1 + DIF(I)))/3.0
DO 5 J=1,10
PRUE=CORR(I)
CALL FUNCT(C,PRUE,F,G)
DIF(I)=G
CORR(I)=PILNE(I-2) + (S*(G2 + 4.0*G1 + DIF(I)))/3.0
IF(ABS(ABS(PRUE)-ABS(CORR(I)))<EPS) 6,6,5
5 CONTINUE
6 PILNE(I)=CORR(I)
GO TO 30
C OBTENCION DE LA SOLUCION CUANDO SI SE DA SOLUCION EXACTA
7 DO 8 I=2,4
C ARRANCAR LA SOLUCION POR EL METODO DE RUNGE-KUTTA
X(I)=X(I-1) + S
C=X(I-1)
D=PILNE(I-1)
CALL FUNCT(C,D,F,G)
PILN(I)=C
C=X(I-1) + 0.5*S
D=PILNE(I-1) + 0.5*S*RUK(I)
CALL FUNCT(C,D,F,G)
RUK(2)=G
Q=PILNE(I-1) + 0.5*S*RUK(2)
CALL FUNCT(C,D,F,G)
RUK(3)=G
C=X(I-1) + S
D=PILNE(I-1) + S*RUK(3)
CALL FUNCT(C,D,F,G)
RUK(4)=G
PILNE(I)=PILNE(I-1) + (S*(RUK(1) + 2.0*RUK(2) + 2.0*RUK(3) + RUK(4)
1))/6.0
C=X(I)
CALL FUNCT(C,D,F,G)
8 Y(I)=F
C CONTINUAR LA SOLUCION POR EL METODO DE PILNE
DO 10 J=5,11
X(I)=X(I-1) + S
C=X(I-1)
D=PILNE(I-1)
CALL FUNCT(C,D,F,G)
G1=G
C=X(I-2)
D=PILNE(I-2)
CALL FUNCT(C,D,F,G)
G2=G
C=X(I-3)
D=PILNE(I-3)
CALL FUNCT(C,D,F,G)
PRE(I)=PILNE(I-4) + (4.0*S*(2.0*G1 + G2 + 2.0*G))/3.0
C=X(J)
D=PRE(I)
CALL FUNCT(C,D,F,G)
DI(I)=G
CORR(I)=PILNE(I-2) + (S*(G2 + 4.0*G1 + DIF(I)))/3.0
Y(I)=F
DO 9 J=1,10
PRUE=CORR(I)
CALL FUNCT(C,PRUE,F,G)
DIF(I)=C
CORR(I)=PILNE(I-2) + (S*(G2 + 4.0*G1 + DIF(I)))/3.0
IF(ABS(ABS(PRUE)-ABS(CORR(I)))<EPS) 10,10,9
9 CONTINUE
10 PILNE(I)=CORR(I)

```

```

C IMPRINTA ESPACIAMIENTO USADO
30 WRITE(6,250) 3
IF(MEXAC) 161,16,161
C GENERACION DE LA MATRIZ PARA GRAFICAR RESULTADOS SIN SOLUCION EXAC
C TA E IMPRESION DE LOS MISMOS
161 N=2
WRITE(6,300)
DO 11 I=1,N
11 WRITE(6,350) X(I),MILNE(I)
DO 15 I=1,N
A(I,1)=X(I)
15 A(I,2)=MILNE(I)
GO TO 22
C GENERACION DE LA MATRIZ PARA GRAFICAR RESULTADOS CON SOLUCION EXAC
C TA E IMPRESION DE LOS MISMOS
16 N=3
WRITE(6,400)
DO 17 I=1,N
17 WRITE(6,450) X(I),MILNE(I),Y(I)
DO 21 I=1,N
A(I,1)=X(I)
A(I,2)=MILNE(I)
21 A(I,3)=Y(I)
C LLAMADO DE SUBROUTINA PARA GRAFICAR
22 CALL GRAFI(A,N,N)
GO TO 1
C FORMATOS DE LECTURA E IMPRESION
100 FORMAT (I11,30(//),33X,'SOLUCION DE UNA ECUACION DIFERENCIAL POR LO
19 METODOS DE',/,30X,'MILNE Y EXACTO(OPCIONAL), CON LAS GRAFICAS CO
2PREFORCIEN'TPS')
150 FORMAT (15,3F10,0)
200 FORMAT (I1)
250 FORMAT (I11,///,15X,'EL ESPACIAMIENTO USADO FUE ',F8,5)
300 FORMAT (///,15X,'X',32X,'MILNE',/)
350 FORMAT (/,12X,F10,5,20X,E15,0)
400 FORMAT (///,15X,'X',19X,'MILNE',19X,'EXACTA',/)
450 FORMAT (/,10X,F10,5,10X,2(E15,0,10X))
END

```

Fig. 9.14 Listado del programa principal

```

SUBROUTINE FUNCT(C,D,F,G)
F = -1.05*EXP(-10.*C) + 1.25*EXP(-2.*C)
G = -10.*EXP(-2.*C) - 10.*D
RETURN
END

```

Fig. 9.15 Listado de la subrutina FUNCT

9.5.4 Ejemplo

La ecuación que caracteriza el voltaje del capacitor del sistema eléctrico mostrado en la figura 9.16 es:

$$\frac{dV_c}{dt} + 10V_c = 10V(t)$$

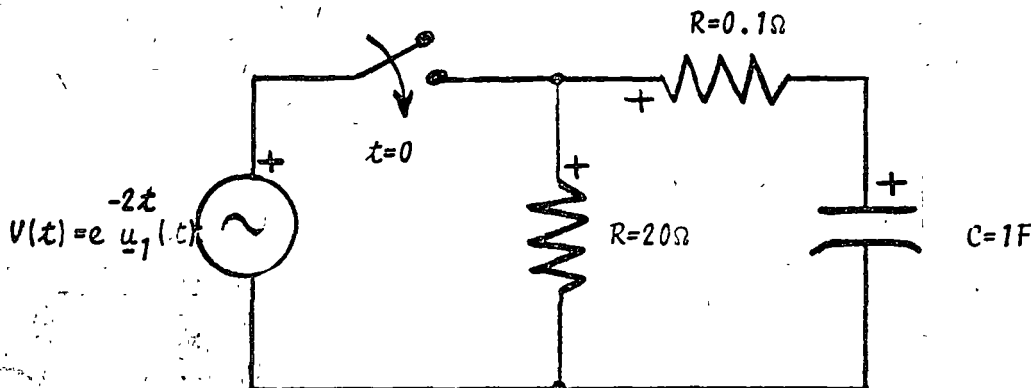


Fig. 9.16 Sistema eléctrico del problema del ejemplo 9.5.4

Obtenga la solución numérica de la ecuación diferencial que caracteriza a $V_c(t)$ para $t \geq 0$ y compare sus resultados con la solución exacta, emplear el método de Milne. Considere los siguientes valores:

$$V_c(t_0) = 0.2\text{V}$$

$$t_0 = 0.$$

$$t_f = 2 \quad \Delta$$

$$\Delta t = .02$$

* SOLUCION

La solución exacta de la ecuación diferencial es:

$$V_c(t) = (V_c(t_0) - \frac{10}{8})e^{-10t} + \frac{10}{8}e^{-2t}$$

TABLA 9.7 Datos del problema del ejemplo 9.5.4

$$N=100$$

$$S=0.02$$

$$X(1)=0.0$$

$$Y(1)=0.2$$

$$KEXAC=0$$

$$F(C)=-1.05*EXP(-10.*C) + 1.25*EXP(-2.*C)$$

$$G(C,D)= 10.*EXP(-2.*C) - 10.*D$$

TABLA 9.8 Resultados del problema del ejemplo 9.5.4

EL ESPACIAMIENTO USADO FUE 0.02000

X	MLLE	EXACTA
0.00000	.20000000E+00	.20000000E+00
0.02000	.34131695E+00	.34131695E+00
0.04000	.45005523E+00	.45005523E+00
0.06000	.53239325E+00	.53239833E+00
0.08000	.59338358E+00	.59338432E+00
0.10000	.63713710E+00	.63714003E+00
0.12000	.66703200E+00	.66703090E+00
0.14000	.68580087E+00	.68580286E+00
0.16000	.69509713E+00	.69509495E+00
0.18000	.69852785E+00	.69853157E+00
0.20000	.69580088E+00	.69579901E+00
0.22000	.68870037E+00	.68870221E+00
0.24000	.67822809E+00	.67822539E+00
0.26000	.66516124E+00	.66516343E+00
0.28000	.65016443E+00	.65016076E+00
0.30000	.63373525E+00	.63373812E+00
0.32000	.61631925E+00	.61631522E+00
0.34000	.59822607E+00	.59822931E+00
0.36000	.57975480E+00	.57975041E+00
0.38000	.56108984E+00	.56109372E+00
0.40000	.54243473E+00	.54242979E+00
0.42000	.52388819E+00	.52389280E+00
0.44000	.50559296E+00	.50558743E+00
0.46000	.48758897E+00	.48759437E+00
0.48000	.46998113E+00	.46997487E+00
0.50000	.45276816E+00	.45277446E+00
.	.	.
1.72000	.40511894E-01	.40080921E-01
1.74000	.33047450E-01	.38509235E-01
1.76000	.37493943E-01	.36999270E-01
1.78000	.35018602E-01	.35540511E-01
1.80000	.34722273E-01	.34154637E-01
1.82000	.32297325E-01	.32815417E-01
1.84000	.32180116E-01	.31528708E-01
1.86000	.29594642E-01	.30292451E-01
1.88000	.29852184E-01	.29104668E-01
1.90000	.27162695E-01	.27965450E-01
1.92000	.27724801E-01	.26840997E-01
1.94000	.24994620E-01	.25813524E-01
1.96000	.25735729E-01	.24861369E-01
1.98000	.22774402E-01	.23823470E-01

9.6 Método de Diferencias Finitas

9.6.1 Objeto

Obtener la solución de una ecuación diferencial ordinaria de segundo orden con valores en la frontera por el método de diferencias finitas.

Dado que el planteamiento para la solución de la ecuación diferencial depende de los puntos donde se especifican los valores en la frontera, se considerará una ecuación diferencial con valores en la frontera en el punto inicial y en el punto final del intervalo en el cual se desea la solución.

9.6.2 Método

Los problemas con valores en la frontera involucran dos o más condiciones del sistema especificadas en puntos diferentes, por lo que las ecuaciones diferenciales que caracterizan a dichos sistemas serán de orden mayor o igual a dos. Para la solución de tales problemas existen dos métodos: el de ensayo y error y el de diferencias finitas.

La solución de una ecuación diferencial por el método de diferencias finitas reduce la integración de la ecuación diferencial a la solución de un sistema de ecuaciones lineales. La solución del sistema de ecuaciones lineales representa la solución de la ecuación diferencial.

En ingeniería los problemas más frecuentes que involucran valores en la frontera son: pandeo y carga, conducción de calor, radiación de calor, deflexión de membranas. En términos generales se tendrá que diseñar un programa para cada problema dado que las condiciones de frontera no estarán especificadas para los mismos puntos.

El proceso que se sigue para aplicar el método de diferencias finitas es:

- ① Dividir el intervalo de integración en "n" subintervalos de igual longitud. A cada uno de los puntos que limita una partición se le denomina pivote.
- ② Substituir en la ecuación diferencial y en las condiciones de frontera las derivadas de la variable dependiente por sus expresiones correspondientes de tipo numérico, procurando que todas las fórmulas de derivación numérica den el mismo tipo de error. Para fórmulas de derivación numérica consultar las referencias 3 y 7 de la bibliografía citada.
- ③ Aplicar la aproximación discretizada de la ecuación diferencial a cada uno de los pivotes, solo se aplica en las fronteras cuando no se conoce su solución. Al encontrarse cerca de las fronteras puede suceder que las fórmulas de derivación numérica requieran puntos localizados fuera del intervalo de integración, este problema se elimina empleando las condiciones de frontera.
- ④ Al aplicar la ecuación diferencial a todos los pivotes se origina un sistema de ecuaciones lineales cuya solución será la solución discretizada de la ecuación diferencial.

Para el caso a tratar se considerará el siguiente sistema:

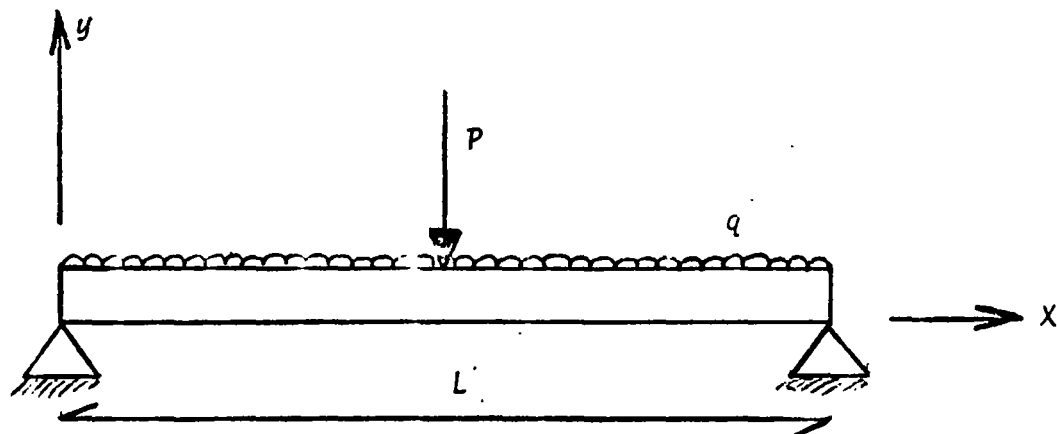


Fig. 9.17 Representación gráfica del problema a resolver por el método de diferencias finitas.

La ecuación que caracteriza el comportamiento de la elástica de una viga es:

$$\frac{d^2 y}{dX^2} = -\frac{Mx}{Ix E} \quad (9.47)$$

para nuestro caso \$E\$ e \$Ix\$ permanecen constantes en toda la extensión de la viga.

La fórmula de derivación numérica correspondiente a la segunda derivada con error \$(\Delta X)^2\$ es:

$$\left. \frac{d^2 y}{dX^2} \right|_i = \frac{1}{(\Delta X)^2} (y_{i-1} - 2y_i + y_{i+1}) \quad (9.48)$$

Al emplear la ecuación (9.48) para discretizar la ecuación (9.47) se obtiene:

$$\frac{1}{(\Delta X)^2} (y_{i-1} - 2y_i + y_{i+1}) = \frac{(Mx)_i}{E Ix} \quad (9.49)$$

Subdividiendo el intervalo de integración (claro de la viga) en "n" partes iguales se tiene:

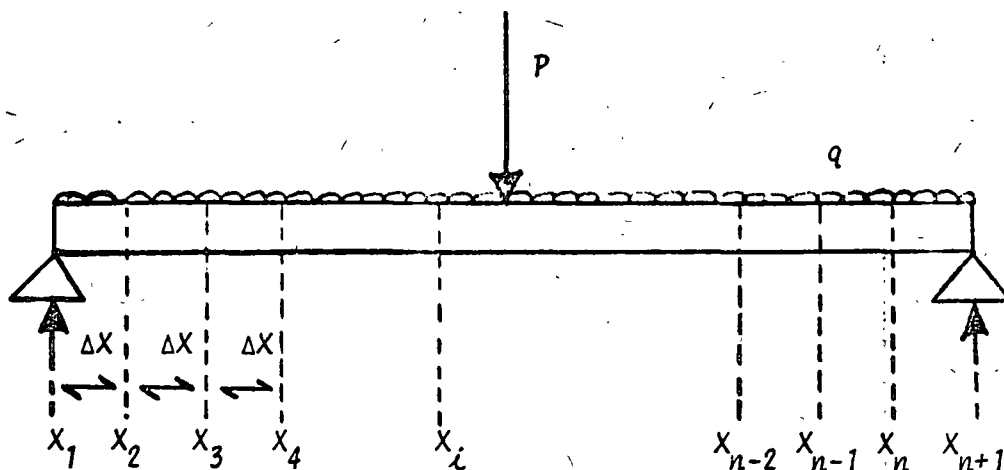


Fig. 9.18 Representación gráfica de la partición del claro.

Para toda viga libremente apoyada las condiciones de frontera son:

$$\left. \begin{aligned} y(X_1) &= y_1 = 0 \\ y(X_{n+1}) &= y_{n+1} = 0 \end{aligned} \right\} \quad (9.50)$$

El momento flexionante aplicado a la viga en estudio para cada valor de X_i , considerando el origen de las abscisas en el extremo izquierdo, es:

$$Mx_i = \begin{cases} \frac{PX_i + qLX_i - qX_i^2}{2}, & X_i \leq \frac{L}{2} \\ \frac{PL + qLX_i - PX_i - qX_i^2}{2}, & X_i > \frac{L}{2} \end{cases} \quad (9.51)$$

Aplicando las ecuaciones (9.49), (9.50) y (9.51) a los puntos X_2, X_3, \dots, X_{n-1} se obtiene el siguiente sistema de ecuaciones:

$$\left. \begin{aligned} -2y_2 + y_3 + 0 + 0 + \dots + 0 + 0 &= \frac{Mx_2}{EI} \\ y_2 - 2y_3 + y_4 + 0 + \dots + 0 + 0 &= \frac{Mx_3}{EI} \\ \vdots & \\ 0 + 0 + 0 + 0 + \dots + y_{n-2} - 2y_{n-1} &= \frac{Mx_{n-1}}{EI} \end{aligned} \right\} \quad (9.52)$$

Se resuelve el sistema de ecuaciones (9.52) mediante alguno de los métodos numéricos conocidos y los valores y_1, y_2, \dots, y_n son la solución de la ecuación diferencial (9.47) y por lo tanto las ordenadas de la curva de la elástica.

Al programa solo se le alimentan los valores de P, q, E e I_x , internamente se plantea el sistema de ecuaciones y se obtiene la solución del mismo.

9.6.3 Descripción del Programa

a) Subrutinas requeridas:

REAL FUNCTION FMOM (P, Q, AL, X), obtiene el momento flexionante para cada uno de los puntos X_i del claro.

SUBROUTINE GAUTOR (A, B, N, EPS, DET), obtiene la solución del sistema de ecuaciones lineales. Consultar capítulo 3.

SUBROUTINE GRAFI (A, N, M), obtiene la gráfica de la curva de la elástica. Consultar capítulo 1.

b) Descripción de las variables.

Para la función FMOM:

P	valor de la carga concentrada
Q	valor de la carga uniformemente distribuida
AL	longitud del claro
X	punto del claro en el cual se desea evaluar el momento flexionante
AL/2	magnitud de la mitad del claro (AL/2)
FMOM	valor del momento flexionante para el punto X_i

Para el programa principal:

N	cantidad de partes en que se subdivide el intervalo de integración
AL	longitud del claro
P	carga concentrada a la mitad del claro
Q	carga uniformemente distribuida en todo el claro

XI	momento de inercia con respecto al eje X
E	módulo de elasticidad
A(I, J)	matriz de coeficientes del sistema de -- ecuaciones
B(I, 1)	vector de términos independientes del -- sistema de ecuaciones, se transforma en la solución
C(I, J)	arreglo matricial con las abscisas y or- denadas para graficar la curva de la --- elástica
X	valor de las abscisas (variable indepen- diente)
DELTA	espaciamiento entre abscisas
XCUA	espaciamiento elevado al cuadrado
DET	variable que indica si el sistema de --- ecuaciones tiene solución
EPS	criterio para determinar si existe solu- ción del sistema de ecuaciones

c) Dimensiones:

La proposición DIMENSION y el valor de N se deberán modificar cuando se desee partir el claro de la viga en más de 15 partes.

d) Formatos para los datos de entrada:

SEC. TARJETAS	FORMATO	INFORMACION
1	(5F15.0)	AL, P, Q, XI, E

 otros paquetes de datos (opcional)

n

TARJETA EN BLANCO, al fi-
nalizar toda la informa-
ción.

e) Diagrama de bloques:

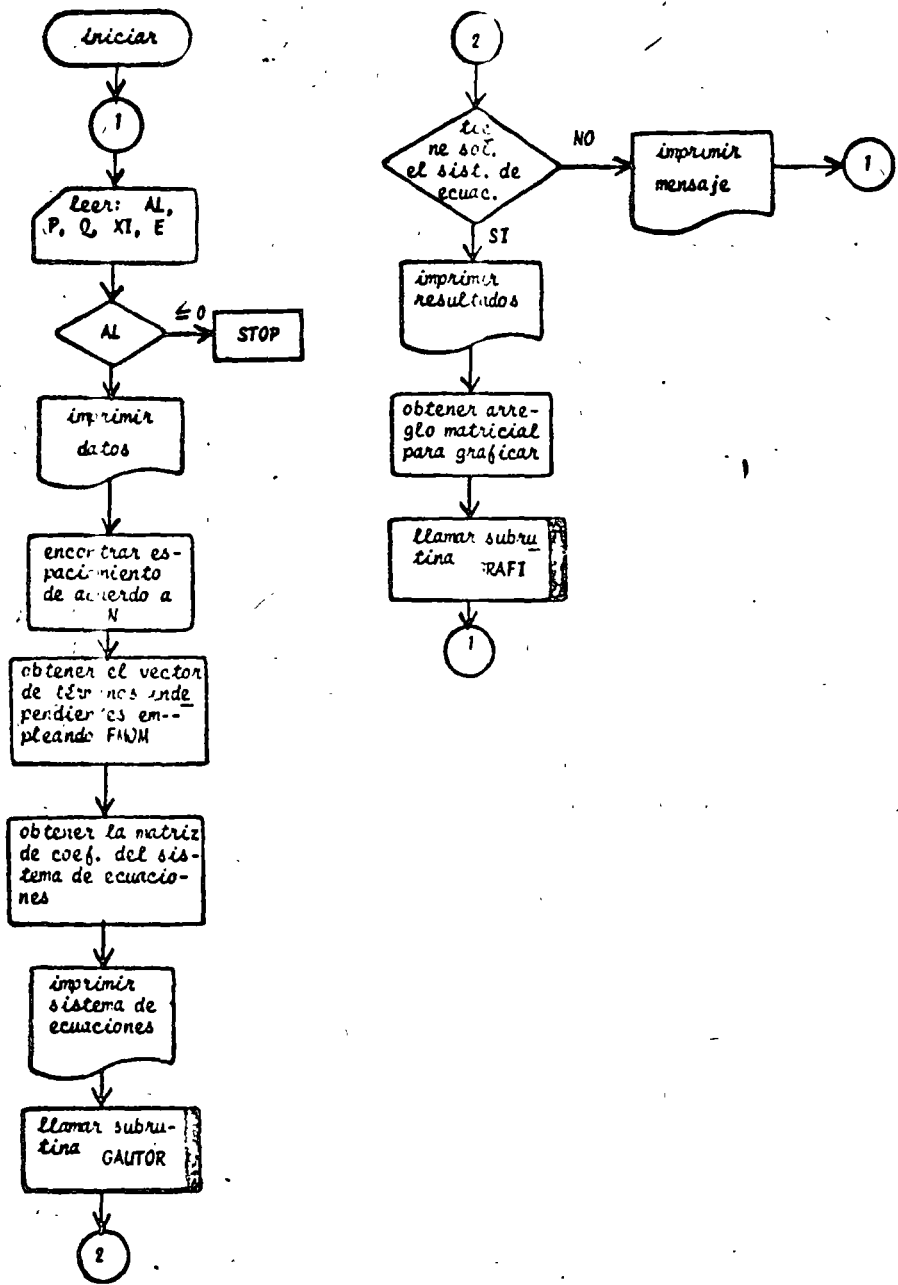


Fig. 9.19 Diagrama de bloques para el programa principal

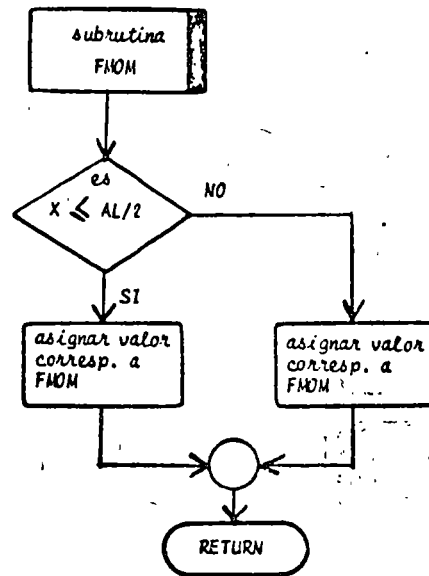


Fig. 9.20 Diagrama de bloques para la función FMOM

6) Listado:

```

C   PROGRAMA PARA ENCONTRAR LA CURVA DE LA ELASTICA DE UNA VIGA LIBRE-
C   MENTE APYADA, CON CARGA CONCENTRADA A LA MITAD DEL CLARO Y CON
C   CARGA UNIFORMEMENTE REPARTIDA, EMPLEANDO EL METODO DE DIFERENCIAS
C   FINITAS.
C   SIGNIFICADO DE LAS VARIABLES EMPLEADAS
C   N=CANTIDAD DE SUBINTERVALOS EN QUE SE DIVIDE EL CLARO
C   AL=LONGITUD DEL CLARO
C   P=VALOR DE LA CARGA CONCENTRADA
C   Q=VALOR DE LA CARGA UNIFORMEMENTE DISTRIBUIDA
C   XI=MOMENTO DE INERCIA DE LA VIGA
C   E=MODULO DE ELASTICIDAD
C   DELT=MAGNITUD DE LOS SUBINTERVALOS
C   X=VALOR DE LAS ASCISAS
C   A=MATRIZ DE COEFICIENTES DEL SISTEMA DE ECUACIONES
C   B=VECTON DE TERMINOS INDEPENDIENTES DEL SISTEMA DE ECUACIONES
C   C=ARRREGLO MATRICIAL PARA GRAFICAR LA CURVA DE LA ELASTICA
C   EPS=CRITERIO PARA DETERMINAR SI EL SISTEMA DE ECUACIONES TIENE SO-
C   LUCION
C   DET=VARIABLE QUE INDICA SI EL SISTEMA DE ECUACIONES TIENE O NO SO-
C   LUCION

```

```

C   DIMENSION B(16),A(16,16),AINV(16,16),C(101,16)
C   LECTURA DE DATOS
1   HEAD(5,100) AL,P,Q,XI,E
   IF(AL) 2,2,3
2   CALL EXIT
3   N=15
   NP1=N + 1
   NM1=N - 1
   DELT=AL/FLOAT(N)
C   IMPRESION DE DATOS
   WRITE(6,150) P
   WRITE(6,200) Q
   WRITE(6,210) E
   WRITE(6,220) XI
C   FORMAR EL SISTEMA DE ECUACIONES POR EL METODO DE DIFERENCIAS FINI-
C   TAS
   DO 15 I=1,NM1
   B(I)=0.0
   DO 15 J=1,NM1
15  A(I,J)=0.0
   X=DELT
   XCUA=DELT**2
   DO 4 I=1,NM1
   B(I)=(P*Q*(AL-X)*XCUA)/(E*XI)
4   X=X + DELT
   DO 7 I=1,NM1
   A(I,I)=-2.0
   IF(I=1) 41,6,41
41  IF(I=NM1) 42,5,42
42  A(I,I-1)=1.0
   A(I,I+1)=1.0
   GO TO 7
5   A(I,I-1)=1.0

```

```

GO TO 7
5 A(I,I+1)=1.0
7 CONTINUE
C IMPRIMIR EL SISTEMA DE ECUACIONES
WRITE(6,250)
WRITE(6,550)
DO 9 I=1,NM1
8 WRITE(6,300) (A(I,J),J=1,NM1)
WRITE(6,500)
DO 8 I=1,NM1
88 WRITE(6,400) B(I)
EPS=0.000001
C LLAMADO DE SUBROUTINA PARA RESOLVER EL SISTEMA DE ECUACIONES
CALL GALTOR(A,B,NM1,EPS,DET)
IF(DET.LE.EPS) GO TO 12
C FORMAR ARREGLO MATRICIAL PARA GRAFICAR LA CURVA DE LA ELASTICA
C(1,1)=C.0
C(1,2)=C.0
X=DELT
DO 9 I=2,N
C(I,1)=X
C(I,2)=P(I-1)
9 X=X + DELT
C(NP1,1)=X
C(NP1,2)=0.3
C IMPRIMIR COORDENADAS DE LA CURVA DE LA ELASTICA
WRITE(6,350)
DO 10 I=1,NP1
10 WRITE(6,400) C(I,1),C(I,2)
C LLAMADO DE SUBROUTINA PARA GRAFICAR RESULTADOS
CALL GRAFIC(C,NP1,2)
GO TO 1
12 WRITE(6,450)
GO TO 1
C FORMATOS DE LECTURA E IMPRESION
100 FORMAT (5F15.0)
150 FORMAT (1H1,2X/),10X,'EL VALOR DE LA CARGA CONCENTRADA APLICADA A
1HITAD DEL CLARO ES',2X,E11.4)
200 FORMAT (4//),10X,'LA CARGA UNIFORMEMENTE DISTRIBUIDA EN TODA LA VI
1GA ES',2X,E11.4)
210 FORMAT (4//),10X,'EL VALOR DEL MODULO DE ELASTICIDAD ES',2X,E11.4)
220 FORMAT (4//),10X,'EL VALOR DEL MOMENTO DE INERCIA ES',2X,E11.4)
250 FORMAT (4//),10X,'EL SISTEMA DE ECUACIONES OBTENIDO AL APLICAR EL
1METODO DE DIFERENCIAS FINITAS ES',//)
300 FORMAT (//,23(F4.1,1X),F4.1)
350 FORMAT (4//),10X,'LA SOLUCION PARA LA CURVA DE LA ELASTICA ES',//,
110X,'DISTANCIA AL ORIGEN (Z)',12X,'ORDENADA DE LA CURVA (Y)',//)
400 FORMAT (//,15X,2(E15.9,20X))
450 FORMAT (3//),20X,'NO EXISTE LA INVERSA DE LA MATRIZ DE COEFICIENTE
1S DEL SISTEMA')
500 FORMAT (//,10X,'VECTOR DE CONSTANTES INDEPENDIENTES',//)
550 FORMAT (//,10X,'MATRIZ DE COEFICIENTES',//)
END

```

Fig. 9.21 Listado del programa principal

```

REAL FUNCTION FMOM(P,0,AL,X)
ALI2=AL/2.0
IF(X=ALI2) 1,1,2
2 FMOM=(P*AL + 0*AL*X - P*X - 0*X*X)/2.0
RETURN
1 FMOM=(P*X + 0*AL*X - 0*X*X)/2.0
RETURN
END

```

Fig. 9.22 Listado de la función FMOM

9.6.4 Ejemplo

Determinar la curva de la elástica para una viga de perfil H180 con las siguientes características:

$$P = 2\,000 \text{ Kg}$$

$$L = 4. \text{ m}$$

$$q = 4\,000 \text{ Kg/m}$$

$$E = 2.1 \times 10^6 \text{ Kg/cm}^2$$

$$I = 3830 \text{ cm}^4$$

$$\text{peso propio} = 52 \text{ Kg/m}$$

* SOLUCION

TABLA 9.9 Datos para el problema del ejemplo 9.6.4

$$AL = 4.0$$

$$P = 2000.$$

$$Q = 4056.$$

$$XI = 0.00003830$$

$$E = 21000000000.$$

TABLA 9.10 Resultados para el problema del ejemplo 9.6.4.

EL VALOR DE LA CARGA CONCENTRADA APLICADA A MITAD DEL CLARO ES .2000E+04

LA CARGA UNIFORMEMENTE DISTRIBUIDA EN TODA LA VIGA ES .4056E+04

EL VALOR DEL MODULO DE ELASTICIDAD ES .2100E+11/

EL VALOR DEL MOMENTO DE INERCIA ES .3830E-04

EL SISTEMA DE ECUACIONES OBTENIDO AL APLICAR EL METODO DE DIFERENCIAS FINITAS

MATRIZ DE COEFICIENTES

2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	-2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	1.0	-2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	-2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	-2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	-2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	-2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.0	-2.0	1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	-2.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	-2.0	1.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	-2.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	-2.0	1.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	-2.0	1.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	-2.0

VECTLR DE CONSTANTES INDEPENDIENTES

.20208299E-03

.37866512E-03

.52974639E-03

.65532680E-03

.75540636E-03

.82998505E-03

.87906289E-03

.87906289E-03

.82998505E-03

.75540636E-03

.65532680E-03

.52974639E-03

.37866512E-03

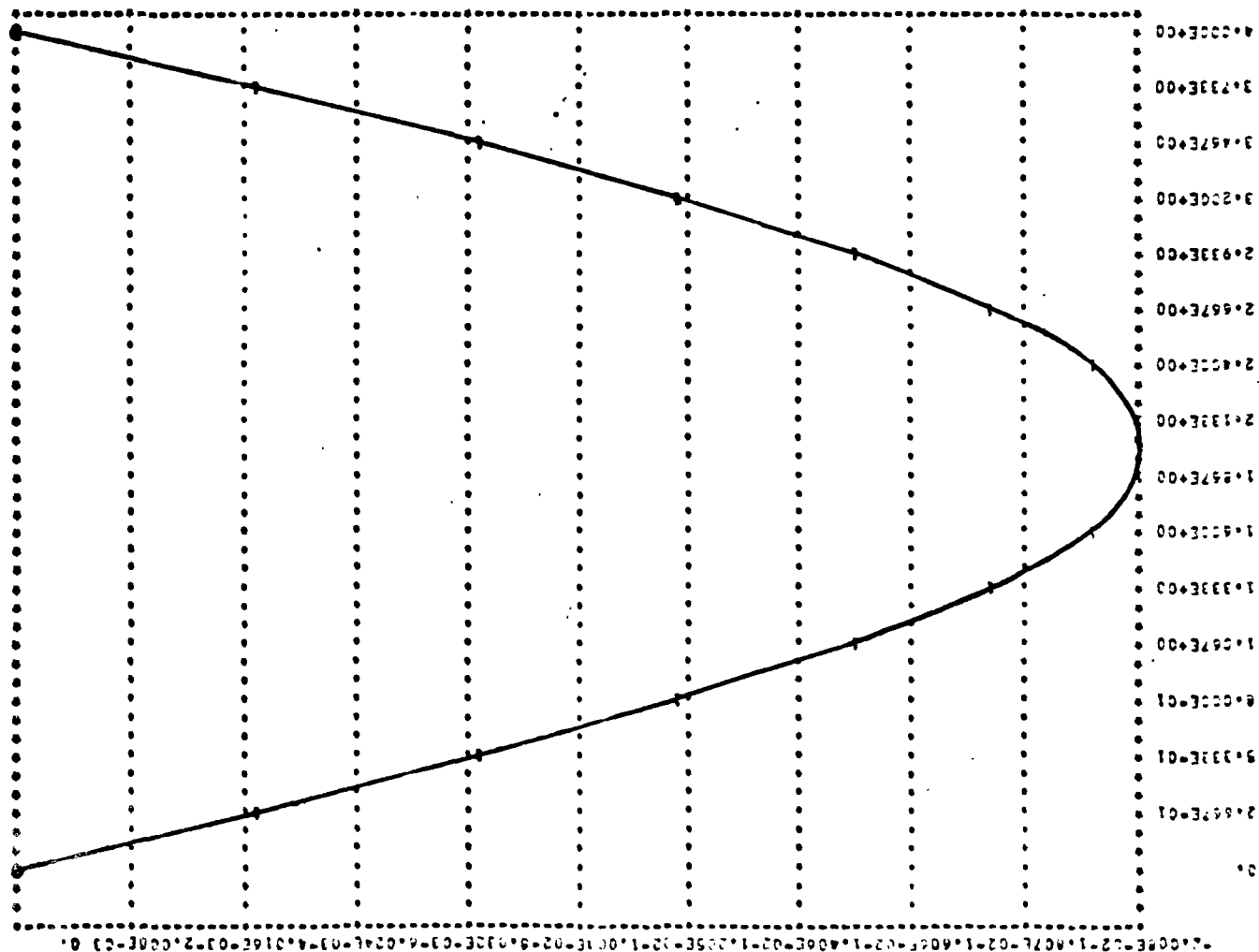
.20208299E-03

LA SOLUCION PARA LA CURVA DE LA ELASTICA ES

DISTANCIA AL ORIGEN (Z)

ORDENADA DE LA CURVA (Y)

Z	Y
0.0000000E+00	0.0000000E+00
0.2666667E+00	0.42302756E-02
0.5333333E+00	0.82584682E-02
0.8000000E+00	0.11907996E-01
1.0666667E+01	0.15027777E-01
1.3333333E+01	0.17492231E-01
1.6000000E+01	0.19201279E-01
1.8666667E+01	0.20080342E-01
2.1333333E+01	0.20080342E-01
2.4000000E+01	0.19201279E-01
2.6666667E+01	0.17492231E-01
2.9333333E+01	0.15027777E-01
3.2000000E+01	0.11907996E-01
3.4666667E+01	0.82584682E-02
3.7333333E+01	0.42302756E-02
4.0000000E+01	0.0000000E+00



9.7 Bibliografía

1. CARNAHAN B., LUTHER H., WILKES J., "Applied Numerical Methods". New York: John Wiley & Sons Inc., 1969.
pp. 341-428.
2. HAMMING Richard, "Numerical Methods for Scientists and Engineers". New York: Mc Graw Hill Book Co., 1962.
pp. 211-222.
3. JAMES M., SMITH G., WOLFORD J., "Applied Numerical -- Methods for Digital Computation with FORTRAN". ---- Scranton Penn.: International Textbook Co., 1967.
pp. 313-459.
4. KAPLAN Wilfred, "Elements of Ordinary Differential -- Equations". Reading Mass.: Addison-Wesley Co., 1964.
pp. 80-104, 138-161, 250-263.
5. KUO S. Shan, "Computer Applications of Numerical --- Methods". Reading Mass: Addison-Wesley Co., 1972.
pp. 128-145.
6. NASH William, "Resistencia de Materiales". México: -- Mc Graw Hill Book Co., 1969.
pp. 139-165.
7. OLIVERA S. Antonio, "Apuntes de Métodos Numéricos". - México: Facultad de Ingeniería, UNAM., 1972.
pp. 6.1-6.19
8. SHANLEY F.R., "Mecánica de Materiales". México" Mc -- Graw Hill Book Co., 1971.
pp. 214-235.

10. SOLUCION DE SISTEMAS DE ECUACIONES DIFERENCIALES ORDINARIAS DE PRIMER ORDEN

10.1 Introducción

Un sistema de "n" ecuaciones diferenciales ordinarias de primer orden tiene la siguiente configuración:

$$\left. \begin{aligned} \frac{dx_1}{dt} &= f_1(x_1, x_2, \dots, x_n, t) \\ \frac{dx_2}{dt} &= f_2(x_1, x_2, \dots, x_n, t) \\ &\vdots \\ \frac{dx_n}{dt} &= f_n(x_1, x_2, \dots, x_n, t) \end{aligned} \right\} \quad (10.1)$$

Dependiendo de las características de las funciones f_j , el sistema puede ser lineal o no lineal.

Cualquier ecuación diferencial de orden "n" se puede expresar como un sistema de "n" ecuaciones diferenciales ordinarias de primer orden mediante un cambio de variables como se indica a continuación.

Sea una ecuación diferencial ordinaria de orden "n":

$$x^{(n)} = g(x, x', x'', \dots, x^{(n-1)}, t) \quad (10.2)$$

efectuando el siguiente cambio de variables:

$$\left. \begin{aligned} x_1 &= x \\ x_2 &= \frac{dx_1}{dt} = x' \\ &\vdots \\ x_n &= \frac{dx_{n-1}}{dt} = x^{(n-1)} \end{aligned} \right\} \quad (10.3)$$

$$* x^{(n)} = \frac{d^n x}{dt^n}$$

se obtiene:

$$\left. \begin{aligned} \frac{dx_1}{dt} &= x_2 \\ \frac{dx_2}{dt} &= x_3 \\ &\vdots \\ \frac{dx_n}{dt} &= g(x_1, x_2, \dots, x_n, t) \end{aligned} \right\} \quad (10.4)$$

El arreglo (10.4) representa un sistema de "n" ecuaciones diferenciales ordinarias de primer orden.

En el presente capítulo se tratarán únicamente dos métodos de solución: el de Runge-Kutta para sistemas no lineales y el de Variación de Parámetros para sistemas lineales.

10.2 Método de Runge-Kutta

10.2.1 Objeto

Obtener la solución de un sistema de dos ecuaciones diferenciales ordinarias de primer orden, lineales o no, mediante el método de Runge-Kutta; es decir, resolver el sistema de ecuaciones diferenciales:

$$\left. \begin{aligned} \frac{dX}{dt} &= f(t, X, Y) \\ \frac{dY}{dt} &= g(t, X, Y) \end{aligned} \right\} \quad (10.5)$$

con condiciones iniciales t_0 , X_0 , y Y_0 .

10.2.2 Método

El método de Runge-Kutta a emplear es el mismo que para una ecuación diferencial ordinaria de primer orden; solo que en este caso se establecen relaciones recursivas para las dos ecua

ciones diferenciales, dichas relaciones se deben de ir resolviendo simultáneamente para obtener la solución. El método se puede extender para resolver un sistema de "n" ecuaciones diferenciales ordinarias de primer orden, en cuyo caso se plantearán 5n relaciones recursivas si se emplean fórmulas de Runge-Kutta de cuarto orden.

Las fórmulas iterativas para un sistema de dos ecuaciones diferenciales ordinarias de primer orden como el mostrado en la ecuación (10.5), empleando fórmulas de Runge-Kutta de cuarto orden, son:

$$\left. \begin{aligned} X_{i+1} &= X_i + \frac{1}{6} (K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 &= (\Delta t) f(t_i, X_i, Y_i) \\ K_2 &= (\Delta t) f\left(t_i + \frac{\Delta t}{2}, X_i + \frac{K_1}{2}, Y_i + \frac{Q_1}{2}\right) \\ K_3 &= (\Delta t) f\left(t_i + \frac{\Delta t}{2}, X_i + \frac{K_2}{2}, Y_i + \frac{Q_2}{2}\right) \\ K_4 &= (\Delta t) f(t_i + \Delta t, X_i + K_3, Y_i + Q_3) \end{aligned} \right\} (10.6)$$

$$\left. \begin{aligned} Y_{i+1} &= Y_i + \frac{1}{6} (Q_1 + 2Q_2 + 2Q_3 + Q_4) \\ Q_1 &= (\Delta t) g(t_i, X_i, Y_i) \\ Q_2 &= (\Delta t) g\left(t_i + \frac{\Delta t}{2}, X_i + \frac{K_1}{2}, Y_i + \frac{Q_1}{2}\right) \\ Q_3 &= (\Delta t) g\left(t_i + \frac{\Delta t}{2}, X_i + \frac{K_2}{2}, Y_i + \frac{Q_2}{2}\right) \\ Q_4 &= (\Delta t) g(t_i + \Delta t, X_i + K_3, Y_i + Q_3) \end{aligned} \right\} (10.7)$$

10.2.3 Descripción del Programa

a) Subrutinas requeridas:

SUBROUTINE FUNCO(T,X,Y,F,G), en esta subrutina se proporcionan las dos ecuaciones diferenciales ordinarias de primer orden.

SUBROUTINE GRAFI(A,N,M), obtiene la gráfica de la solución para las dos ecuaciones diferenciales. Consultar el capítulo 1.

b) Descripción de las variables:

Para la subrutina FUNCO:

T	valor de la variable independiente t_i
X	valor de la primera variable dependiente x_i
Y	valor de la segunda variable dependiente y_i
F	primera ecuación diferencial correspondiente a la derivada de la variable dependiente x_i
G	segunda ecuación diferencial correspondiente a la derivada de la variable dependiente y_i

Para el programa principal:

N	cantidad de puntos en que se divide el intervalo de integración
S	espaciamiento entre los valores de la variable independiente (Δt)
X(1)	valor inicial de la variable dependiente X
Y(1)	valor inicial de la variable dependiente Y
T(1)	valor inicial de la variable independiente t
T(I)	valores de la variable independiente t
X(I)	valores de la variable dependiente X
Y(I)	valores de la variable dependiente Y
RUK(I)	parámetros de las fórmulas iterativas de Runge-Kutta

QUK(I)	parámetros de las fórmulas iterativas de Runge-Kutta
C	variable de reemplazo
D	variable de reemplazo
E	variable de reemplazo
A(I, J)	arreglo matricial para graficar los resultados
M	columnas de la matriz A
F	valor de la derivada de X
G	valor de la derivada de Y

c) Dimensiones:

La proposición DIMENSION deberá modificarse en el caso de que:

$$N > 100$$

d) Formatos para los datos de entrada:

SEC. TARJETAS	FORMATO	INFORMACION
1	(I5,4F10.0)	N, S, X(1), Y(1), T(1)

 otros paquetes de datos (opcional)

n	TARJETA EN BLANCO, al finalizar toda la información.
---	--

e) Diagrama de bloques:

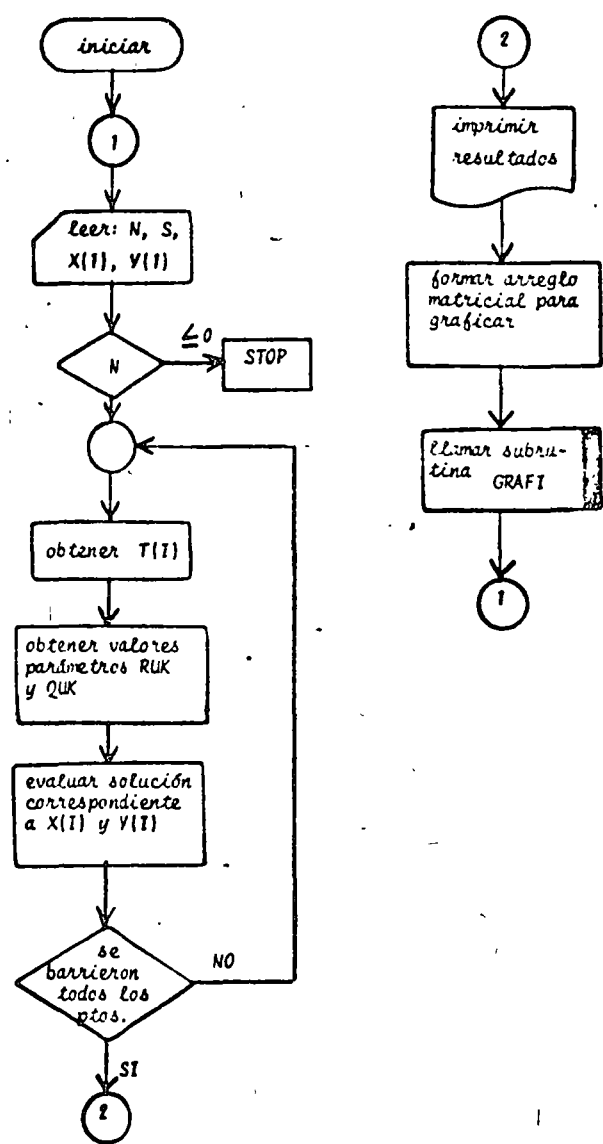


Fig. 10.1 Diagrama de bloques del programa principal


```

SUBROUTINE FUNC(X,F)
  DIMENSION Y(10)
  Y(1) = 1.0
  DO 10 I=2,10
    Y(I) = 1.0 + 0.02 * Y(I-1)
  10 CONTINUE
  F = Y(10)
  RETURN
END

```

Fig. 10.3 Listado de la subrutina FUNC

10.2.4 Ejemplo

El sistema mecánico de la figura 10.4 tiene como ecuación diferencial que caracteriza su movimiento a:

$$\frac{d^2\theta}{dt^2} = \frac{T(t)}{ML^2} - \frac{g}{L} \text{sen}(\theta) - \frac{Ba}{ML} \dot{\theta} \quad (10.8)$$

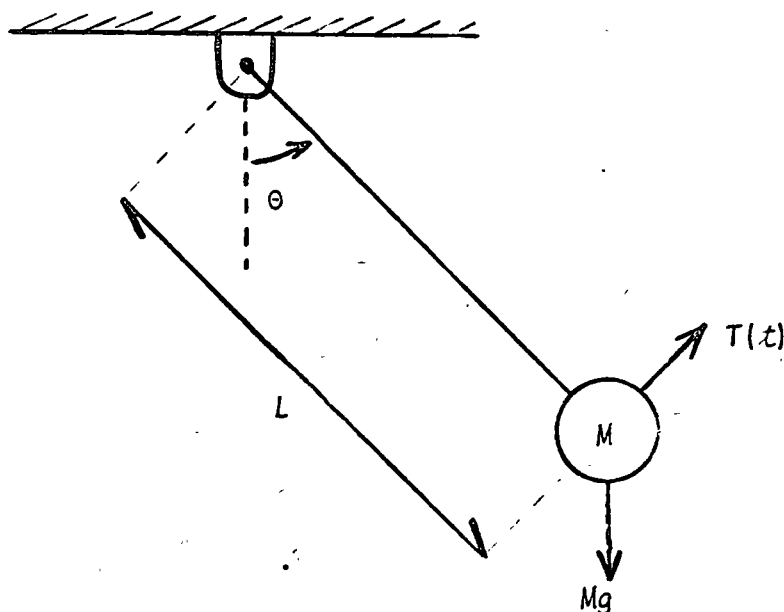


Fig. 10.4 Sistema mecánico del problema del ejemplo 10.2.4

La representación de la ecuación (10.8) mediante un sistema de ecuaciones es:

$$\dot{\theta} = w$$

$$\dot{w} = \frac{T(t)}{ML^2} - \frac{g}{L} \text{sen}(\theta) - \frac{Ba}{ML} w$$

Obtenga la solución del sistema de ecuaciones para las siguientes condiciones y valores de los parámetros:

$$M = 1 \text{ Kgm}$$

$$L = 0.5 \text{ m}$$

$$g = 9.8 \text{ m/s}^2$$

$$Ba = 0.01 \text{ Nt-m-s}$$

$$\begin{aligned}
 t_0 &= 0. \\
 \theta_0 &= 0 \text{ rad} \\
 \omega_0 &= 0.3 \text{ rad/s} \\
 t_f &= 2 \text{ s} \\
 T(t) &= 1 \text{ Nt-m}
 \end{aligned}$$

* SOLUCION

TABLA 10.1 Datos para el problema del ejemplo 10.2.4

$$N = 101$$

$$S = .02$$

$$X(1) = 0.$$

$$Y(1) = 0.3$$

$$T(1) = 0.$$

$$F = Y$$

$$G = 4.0 - 19.6 * \text{SIN}(X) - 0.02 * Y$$

TABLA 10.2 Resultados del problema del ejemplo 10.2.4

EL ESPACIO INTERNO DE TIEMPO EN LOS AÑOS 1990 Y 2000

TIEMPO	X	Y
0.	0.	.30000000E+00
.20000000E+01	.67733324E+02	.37353460E+00
.40000000E+01	.15123393E+01	.45417230E+00
.60000000E+01	.24932293E+01	.52617294E+00
.80000000E+01	.36142616E+01	.59472369E+00
.10000000E+02	.43633040E+01	.65719766E+00
.12000000E+02	.52390359E+01	.71520458E+00
.14000000E+02	.77233709E+01	.76759098E+00
.16000000E+02	.73059203E+01	.81397375E+00
.18000000E+02	.10775025E+02	.85398371E+00
.20000000E+02	.12717499E+02	.88732847E+00
.22000000E+02	.14519731E+02	.91376594E+00
.24000000E+02	.15367601E+02	.93311052E+00
.26000000E+02	.15247357E+02	.94523590E+00
.28000000E+02	.20143004E+02	.95007523E+00
.30000000E+02	.22742793E+02	.94762102E+00
.32000000E+02	.23727537E+02	.93792444E+00
.34000000E+02	.25739731E+02	.92109401E+00
.36000000E+02	.27059263E+02	.89729357E+00
.38000000E+02	.29374401E+02	.86673913E+00
.40000000E+02	.31071095E+02	.82959934E+00
.42000000E+02	.32634000E+02	.78683515E+00
.44000000E+02	.34213955E+02	.73745316E+00
.46000000E+02	.35635270E+02	.69299331E+00
.	.	.
.	.	.
.17400000E+01	.22110946E+02	.7337703E+00
.17600000E+01	.23270350E+02	.72340977E+00
.17800000E+01	.2511204E+02	.70619165E+00
.18000000E+01	.27601345E+02	.68307302E+00
.18200000E+01	.29338113E+02	.65260812E+00
.18400000E+01	.31707921E+02	.61576148E+00
.18600000E+01	.32977107E+02	.57794417E+00
.18800000E+01	.34070033E+02	.54220317E+00
.19000000E+01	.35941306E+02	.50924443E+00
.19200000E+01	.36770791E+02	.4813790E+00
.19400000E+01	.3793103E+02	.45890673E+00
.19600000E+01	.3938417E+02	.44079276E+00
.19800000E+01	.4113001E+02	.42705907E+00
.20000000E+01	.4317000E+02	.4179914E+00

CONCEPTOS BASICOS

Por Ing. Rubén Chávez S.

B. - AGUA SUBTERRANEA: EL RECURSO DEL FUTURO.

Estimaciones comparativas han revelado que, a nivel mundial, el recurso hidráulico disponible en el subsuelo es mucho mayor que el disponible en la superficie. Según una de las estimaciones, más del 90% del agua dulce existente en la Tierra se encuentra bajo la superficie del terreno; otra de ellas indica que el volumen de agua almacenado en el subsuelo de nuestro planeta es unas 20 veces mayor que el de agua dulce superficial.

Independientemente de la dudosa precisión de las cifras anteriores, el hecho es que las fuentes de agua superficial ya están siendo aprovechadas en su mayoría, mientras las demandas de agua continúan aumentando progresivamente a causa de la explosión demográfica. Esto significa que en el futuro las demandas tendrán que ser satisfechas cada vez en mayor proporción con agua procedente de las fuentes subterráneas. Por esto se agrega que gran parte del planeta está ocupado por zonas desérticas, donde el único recurso hidráulico disponible se encuentra en el subsuelo; queda fuera de toda duda la gran importancia de este recurso.

Aguas Subterráneas vs. Aguas Superficiales.-

Por lo demás de su mayor abundancia, el agua subterránea presenta, por naturaleza, varias ventajas con respecto al agua superficial, haciendo su aprovechamiento más atractivo en las zonas:

a) Menores pérdidas por evaporación. Todos los días en la superficie del agua superficial pierden cantidades significativas de agua por evaporación. Por ejemplo, en una zona donde la tasa de evaporación anual es de unos 2 m/año, una masa de agua superficial perdería por este concepto un volumen del orden de 2 millones de m³ por km² de extensión superficial. Este volumen sería equivalente al extraído por un pozo que operara con-

10.3 Solución de Sistemas Homogéneos de Ecuaciones Diferenciales Ordinarias Lineales de Primer Orden

10.3.1 Objeto

Obtener la solución numérica de sistemas de ecuaciones diferenciales ordinarias de primer orden, homogéneos y lineales por el método de Variación de Parámetros. Los sistemas del tipo antes mencionado tienen la siguiente configuración:

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \cdot \\ \cdot \\ \frac{dx_n}{dt} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdot & \cdot & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \quad (10.9)$$

sujeto a las condiciones iniciales:

$$t_0, x_1(t_0), x_2(t_0), \dots, x_n(t_0)$$

10.3.2 Método

El sistema de ecuaciones diferenciales de la expresión (10.9) se puede representar en la forma compacta:

$$\left. \begin{aligned} \dot{\underline{x}}(t) &= \underline{A} \underline{x}(t) \\ \underline{x}(t_0) &= \underline{x}_0 \end{aligned} \right\} \quad (10.10)$$

El método de variación de parámetros establece que la solución del sistema de ecuaciones (10.10) es:

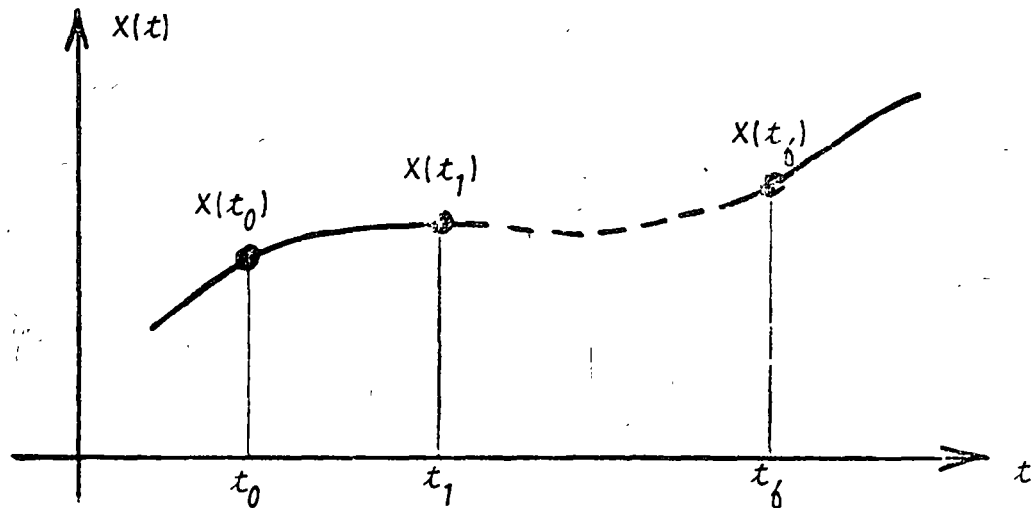
$$\underline{x}(t) = e^{\underline{A}(t-t_0)} \underline{x}_0 \quad (10.11)$$

donde a la matriz $e^{\underline{A}(t-t_0)}$ se le conoce como matriz de transición y se define mediante la siguiente serie infinita:

$$e^{\underline{A}(t-t_0)} = \underline{I} + \frac{\underline{A}(t-t_0)}{1!} + \dots + \frac{\underline{A}^n(t-t_0)^n}{n!} + \dots \quad (10.12)$$

Para un estudio detallado del origen de las expresiones (10.11) y (10.12) se recomienda consultar las referencias 1, 3 y 6 de la bibliografía anexa.

La ley de evolución de estados para un sistema lineal se puede representar gráficamente como:



es decir, se puede pasar del estado $\underline{X}(t_0)$ al estado $\underline{X}(t_f)$ directamente o a través de $\underline{X}(t_1)$. La solución del sistema (10.10) mediante computadora digital se basa en esta característica; para evaluar la respuesta $\underline{X}(t)$ desde t_0 hasta t_f se discretiza el intervalo de integración en "n+1" puntos igualmente espaciados y se obtiene la solución haciendo evolucionar el estado desde t_0 hasta t_1 , de t_1 a t_2 y así sucesivamente. Analíticamente se tendrá:

$$\underline{X}(t_1) = e^{\underline{A}(t_1-t_0)} \underline{X}(t_0)$$

$$\underline{X}(t_2) = e^{\underline{A}(t_2-t_1)} \underline{X}(t_1)$$

$$\vdots$$

$$\underline{x}(t_n) = e^{\underline{A}(t_n - t_{n-1})} \underline{x}(t_{n-1}) \quad (10.13)$$

pero:

$$e^{\underline{A}(t_1 - t_0)} = e^{\underline{A}(t_n - t_{n-1})} = e^{\underline{A} \Delta t} \quad (10.14)$$

por lo que solo se evaluará una vez la matriz $e^{\underline{A} \Delta t}$ ya que es la misma matriz para todos los intervalos puesto que el espaciamiento es constante.

Para evaluar $e^{\underline{A} \Delta t}$ se emplea la relación (10.12), donde la cantidad de términos empleados dependerá del criterio de convergencia empleado; o sea, dado un criterio ϵ la sumatoria de términos se detendrá cuando:

$$\left| e^{\frac{\underline{A} \Delta t}{x_j(n)}} - e^{\frac{\underline{A} \Delta t}{x_j(n-1)}} \right| < \epsilon, \text{ para toda } ij \quad (10.15)$$

Una vez evaluada $e^{\underline{A} \Delta t}$ la solución se obtiene mediante las relaciones dadas en (10.13).

10.3.3 Descripción del Programa

a) Subrutinas requeridas:

SUBROUTINE EXPMA(DELTA, M, A, EXPO), obtiene la matriz de transición. Criterio de convergencia empleado:
= 0.0001.

SUBROUTINE GRAFI(A, N, M), grafica las soluciones de todas las variables dependientes. Consultar el capítulo 1.

SUBROUTINE MULTMA(A, B, N, M, L, X), efectúa el producto entre dos matrices conformables. Consultar el capítulo 2.

b) Descripción de las variables:

Para la subrutina EXPMA:

DELTA espaciamiento entre los valores de la variable independiente (t)
M cantidad de ecuaciones diferenciales
A(I, J) matriz de coeficientes constantes del sistema de ecuaciones

EXPO(I, J) matriz de transición
 CMA(I, J) matriz identidad
 B(I, J) matriz A elevada a la potencia "n"
 X(I, J) producto matricial AB
 EPS criterio de convergencia para la serie
 DIV factorial de "n"
 CN contador que incrementa a DIV
 TNEW espaciamiento de la variable independiente elevado a la potencia "n"

Para el programa principal:

M cantidad de ecuaciones diferenciales
 PERIO magnitud del intervalo de integración
 DELT espaciamiento entre los valores de la variable independiente
 N cantidad de puntos muestrales en que se dará discretizada la solución
 A(I, J) matriz de coeficientes constantes del sistema de ecuaciones diferenciales
 X(1, 1) condición inicial de la variable independiente
 X(1, J) condiciones iniciales de las variables X_1, X_2, \dots, X_n para $J > 1$
 EXPO(I, J) matriz de transición
 X(I, J) valores de la variable independiente y de las variables dependientes

c) Dimensiones:

La proposición DIMENSION del programa principal y de las subrutinas deberá ser modificada cuando:

$M > 5$ y/o $N > 101$

d) Formatos para los datos de entrada:

SEC. TARJETAS	FORMATO	INFORMACION
1	(I5, 2F10.0)	M, DELT, PERIO
2	(8F10.0)	A(I, J), los elementos de la matriz se dan renglón por renglón. Emplear las tarjetas que sean necesarias
.		
.		
.		

3

(8F10.0)

X(1,J), el primer valor corresponde a la condición inicial de la variable independiente

 otros paquetes de datos (opcional)

n

TARJETA EN BLANCO, al finalizar toda la información

e) Diagrama de bloques:

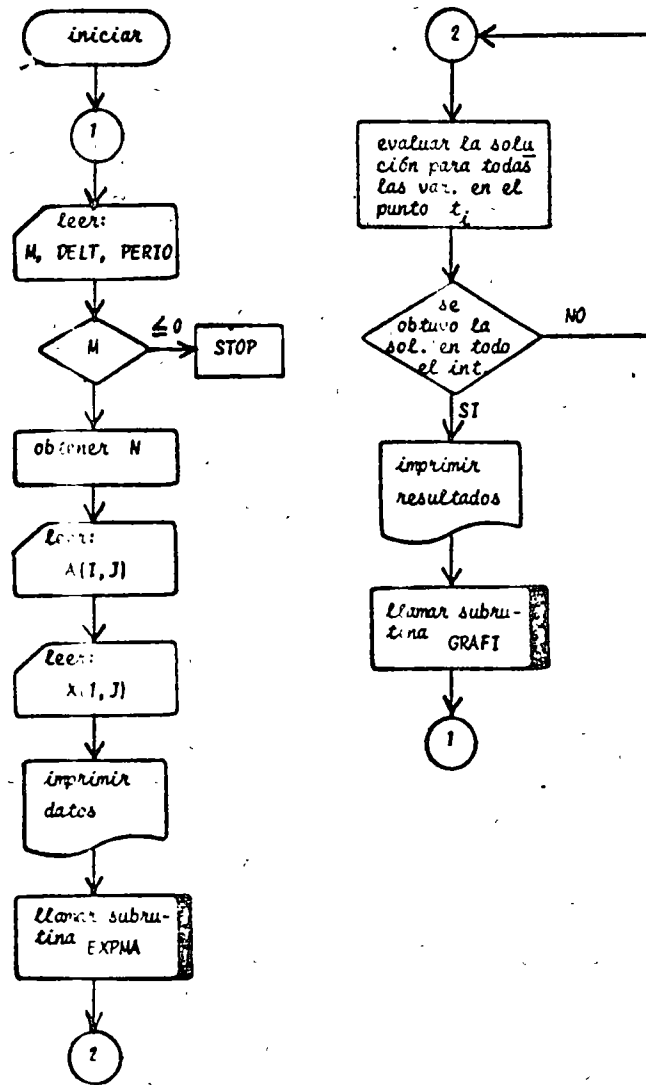


Fig. 10.5 Diagrama de bloques del programa principal

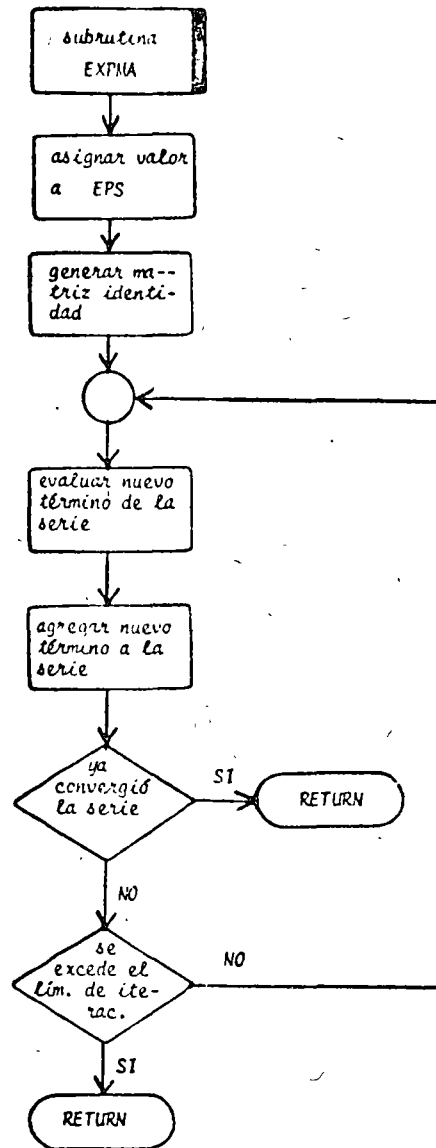


Fig. 10.6 Diagrama de bloques de la subrutina EXPMA

6) Listado:

```

C      PROGRAMA PARA RESOLVER SISTEMAS DE ECUACIONES DIFERENCIALES HOMOG
C      NEAS ORDINARIAS DE PRIMER ORDEN
C      SIGNIFICADO DE LAS VARIABLES EMPLEADAS
C      M=CANTIDAD DE ECUACIONES DIFERENCIALES
C      PERIO=MAGNITUD DEL INTERVALO DE INTEGRACION
C      DELT=ESPACIAMIENTO ENTRE VALORES DE LA VARIABLE INDEPENDIENTE
C      A=MATRIZ DE COEFICIENTES CONSTANTES DEL SISTEMA DE ECUACIONES
C      X(1,J)=CONDICIONES INICIALES
C      X=SOLUCIONES DEL SISTEMA

      DIMENSION A(6,6),EXPO(6,6),X(101,6)
C      LECTURA DE DATOS
1      READ(5,100) M,DELT,PERIO
      IF(M) 2,2,3
2      CALL EXIT
3      RN=PERIO/DELT
      N=RN + 1
      MP1=M+1
      DO 4 I=1,M
4      READ(5,200) (A(I,J),J=1,M)
      READ(5,200) (X(1,J),J=1,MP1)
C      IMPRESION DE DATOS
      WRITE(6,300)
      DO 5 I=1,M
5      WRITE(6,400) (A(I,J),J=1,M)
      WRITE(6,500)
      WRITE(6,400) (X(1,J),J=1,MP1)
C      LLAMADO DE SUBROUTINA PARA OBTENER LA MATRIZ DE TRANSICION
C      CALL EXPOA(DELT,M,A,EXPO)
C      OBTENER LA SOLUCION
      DO 9 K=2,N
      DO 7 J=1,M
      X(K,J+1)=0.0
      DO 7 I=2,MP1
7      X(K,J+1)=X(K,J+1) + EXPO(J,I-1)*X(K-1,I)
      X(K,1)=X(K-1,1) + DELT
9      CONTINUE
C      IMPRIMIR RESULTADOS
      WRITE(6,600)
      DO 10 I=1,N
10      WRITE(6,400) (X(I,J),J=1,MP1)
C      LLAMADO DE SUBROUTINA PARA GRAFICAR
      CALL GRAFI(X,N,MP1)
      GO TO 1
C      FORMATOS DE LECTURA E IMPRESION
100  FORMAT (15,2F10.0)
200  FORMAT (8F10.0)
300  FORMAT (1H1,5(/),15X,4HLA MATRIZ DE LOS COEFICIENTES DEL SISTEMA
      1ES,/)
400  FORMAT (/,152 6(E12.5,32))
500  FORMAT (4(/),15X,29HLAS CONDICIONES INICIALES SON,/,18X,6HTIEMPO,1
      10X,4HX(1),11X,4HX(2),11X,4HX(3),11X,4HX(4),11X,4HX(5),/)
600  FORMAT (4(/),15X,14HLA SOLUCION ES,/,18X,6HTIEMPO,10X,4HX(1),11X,4
      1HX(2),11X,4HX(3),11X,4HX(4),11X,4HX(5),/)
      END

```

Fig. 10.7 Listado del programa principal

```

SUBROUTINE EXPMA(DELTA,M,A,EXPO)
SUBROUTINA PARA OBTENER LA MATRIZ DE TRANSICION ASOCIADA A UNA MA-
TRIZ CUADRADA
C SIGNIFICADO DE LAS VARIABLES EMPLEADAS
C DELTA=MAGNITUD DE LOS INCREMENTOS DE TIEMPO
C M=ORDEN DE LA MATRIZ
C A=MATRIZ DE LA QUE SE DESEA LA MATRIZ DE TRANSICION
C EPS=CRITERIO DE CONVERGENCIA
C CMA=MATRIZ IDENTIDAD
C EXPC=MATRIZ DE TRANSICION
C CN=CONTADOR DE ITERACIONES
C DIV=FACTORIAL DIVISOR
C U=MATRIZ A ELEVADA A LA POTENCIA N
C X=MATRIZ PRODUCTO AB
C TNEW=ESPACIAMIENTO DE TIEMPO ELEVADO A LA POTENCIA N
DIMENSION A(6,6),EXP(6,6),CMA(6,6),B(6,6),X(6,6)
EPS=0.00001
C GENERAR MATRIZ IDENTIDAD
DO 3 I=1,M
DO 2 J=1,M
IF(I.EQ.J) GO TO 1
CMA(I,J)=0.0
GO TO 2
1 CMA(I,J)=1.0
2 CONTINUE
3 CONTINUE
C OBTENER LOS PRIMEROS TERMINOS DE LA SERIE
DO 4 I=1,M
DO 4 J=1,M
EXP(I,J)=CMA(I,J) + A(I,J)*DELTA
4 B(I,J)=A(I,J)
TNEW=DELTA
CN=2.0
DIV=1.0
C OBTENER LOS TERMINOS RESTANTES
5 DIV=DIV*CN
TNEW=TNEW*DELTA
CALL MULTA(6,6,M,M,4,X)
DO 6 I=1,M
DO 6 J=1,M
B(I,J)=X(I,J)
6 CMA(I,J)=(X(I,J)+TNEW)/DIV
IF(CN.LE.4.0) GO TO 9
AMAX=CMA(1,1)
DO 8 I=1,M
DO 7 J=1,M
IF(ABS(CMA(I,J)).LE.ABS(AMAX)) GO TO 7
AMAX=CMA(I,J)
7 CONTINUE
8 CONTINUE
C REVISAR CONVERGENCIA
IF(ABS(AMAX).LE.EPS) GO TO 11
9 DO 10 I=1,M
DO 10 J=1,M
10 EXP(I,J)=EXP(I,J) + CMA(I,J)
IF(CN.GT.20.) GO TO 11
CN=CN + 1.0
GO TO 5
11 RETURN
END

```

Fig. 10.8 Listado de la subrutina EXPMA

10.3.4 Ejemplo

La representación mediante variables de estado para el sistema mecánico de la figura 10.9 es:

$$\dot{x}_1 = s_1$$

$$\dot{x}_2 = s_2$$

$$\dot{s}_1 = \frac{K}{M_1} (x_2 - x_1)$$

$$\dot{s}_2 = \frac{f(t)}{M_2} - \frac{BS_2}{M_2} - \frac{K}{M_2} (x_2 - x_1)$$

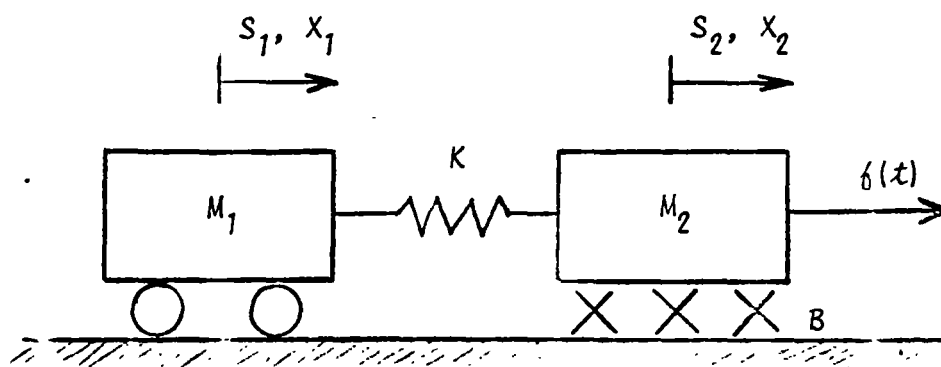


Fig. 10.9 Sistema mecánico del ejemplo 10.3.4

Determine la respuesta libre del sistema si:

$$M_1 = 1$$

$$M_2 = 0.1$$

$$K = 0.9$$

$$B = 1.5$$

$$x_0 = 0.0$$

$$t_0 = 10.0$$

$$x_1(t_0) = 0.$$

$$s_1(t_0) = 1.$$

$$x_2(t_0) = 2.$$

$$s_2(t_0) = 1.5$$

Todas las unidades están dadas en unidades del sistema MKS. Considere como salidas del sistema las velocidades y los desplazamientos.

* SOLUCION

TABLA 10.3 Datos para el problema del ejemplo 10.3.4

$$M = 4$$

$$\text{PERIO} = 10.$$

$$\text{DELT} = 0.1$$

$$\underline{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 9 & 0 & -9 & -15 \end{bmatrix}$$

$$\underline{X}(1, J) = \begin{bmatrix} 0 & 0 & 1 & 2 & 1.5 \end{bmatrix}$$

TABLA 10.4 Resultados del problema del ejemplo 10.3.4

LA MATRIZ DE LOS COEFICIENTES DEL SISTEMA ES

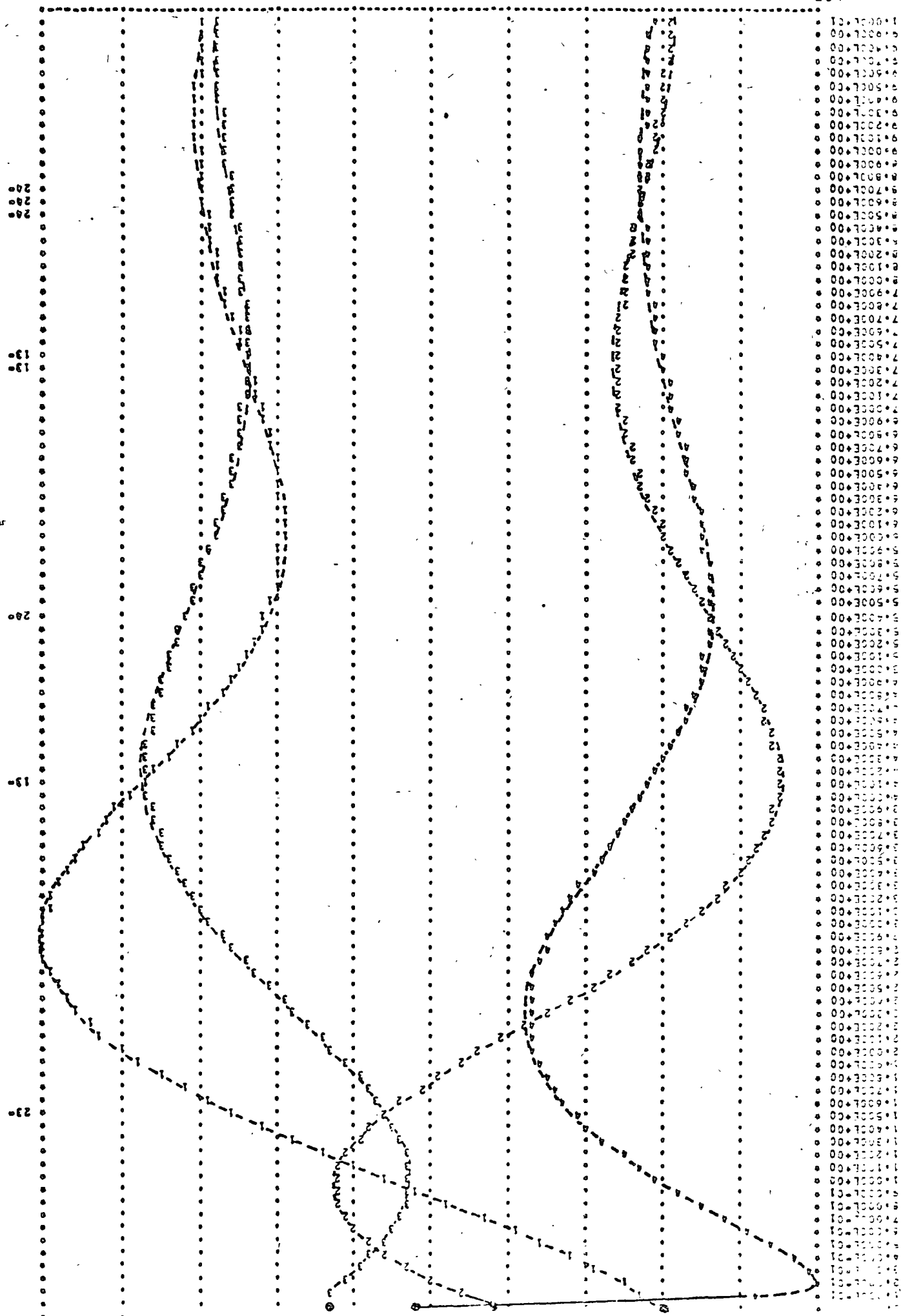
0.	.10000E+01	0.	0.
-.10000E+01	0.	.10000E+01	0.
0.	0.	0.	.10000E+01
.90000E+01	0.	-.90000E+01	-.15000E+02

LAS CONDICIONES INICIALES SON

TIEMPO	X(1)	X(2)	X(3)	X(4)
0.	0.	.10000E+01	.20000E+01	.15000E+01

LA SOLUCION ES

TIEMPO	X(1)	X(2)	X(3)	X(4)
0.	0.	.10000E+01	.20000E+01	.15000E+01
.10000E+00	.10995E+00	.11974E+01	.20202E+01	-.57957E+00
.20000E+00	.23890E+00	.13782E+01	.19371E+01	-.96037E+00
.30000E+00	.38482E+00	.15360E+01	.18408E+01	-.93625E+00
.40000E+00	.54529E+00	.16692E+01	.17528E+01	-.81575E+00
.50000E+00	.71784E+00	.17776E+01	.16783E+01	-.67329E+00
.60000E+00	.89999E+00	.18615E+01	.16183E+01	-.52764E+00
.70000E+00	.10893E+01	.19215E+01	.15727E+01	-.38428E+00
.80000E+00	.12835E+01	.19585E+01	.15413E+01	-.24541E+00
.90000E+00	.14803E+01	.19734E+01	.15234E+01	-.11238E+00
.10000E+01	.16775E+01	.19675E+01	.15186E+01	.13766E-01
.11000E+01	.18732E+01	.19421E+01	.15259E+01	.13216E+00
.
.
.
.90000E+01	.28338E+01	.50181E-01	.26629E+01	.10430E+00
.91000E+01	.28380E+01	.33384E-01	.26732E+01	.10122E+00
.92000E+01	.28405E+01	.17264E-01	.26932E+01	.97306E-01
.93000E+01	.28415E+01	.19436E-02	.26927E+01	.92644E-01
.94000E+01	.28410E+01	-.12472E-01	.27017E+01	.87326E-01
.95000E+01	.28390E+01	-.25890E-01	.27101E+01	.81447E-01
.96000E+01	.28358E+01	-.38236E-01	.27179E+01	.75101E-01
.97000E+01	.28314E+01	-.49451E-01	.27251E+01	.68378E-01
.98000E+01	.28260E+01	-.59489E-01	.27316E+01	.61371E-01
.99000E+01	.28196E+01	-.68317E-01	.27378E+01	.54166E-01
.10000E+02	.28124E+01	-.75923E-01	.27426E+01	.46850E-01



91

10.4 Solución de Sistemas de Ecuaciones Diferenciales Lineales No Homogéneas de Primer Orden

10.4.1 Objeto

Obtener la solución de sistemas de ecuaciones diferenciales no homogéneas, lineales, de primer orden mediante el método de Variación de Parámetros.

La representación en forma matricial para este tipo de sistemas de ecuaciones diferenciales es:

$$\begin{aligned}\dot{\underline{X}}(t) &= \underline{A} \underline{X}(t) + \underline{B} \underline{U}(t) \\ \underline{X}(t_0) &= \underline{X}_0\end{aligned}\quad (10.16)$$

donde $\underline{U}(t)$ representa el vector de entradas externas si se habla de sistemas físicos.

Debido a que cuando se modelan sistemas dinámicos lineales las salidas no siempre corresponden a las variables empleadas en las ecuaciones diferenciales, en este programa se considera la representación completa mediante variables de estado de un sistema lineal, la cual es:

$$\left. \begin{aligned}\dot{\underline{X}}(t) &= \underline{A} \underline{X}(t) + \underline{B} \underline{U}(t) \\ \underline{Y}(t) &= \underline{C} \underline{X}(t) + \underline{D} \underline{U}(t) \\ \underline{X}(t_0) &= \underline{X}_0\end{aligned}\right\} \quad (10.17)$$

donde $\underline{Y}(t)$ representa el vector de salidas del sistema.

10.4.2 Método

El método de variación de parámetros establece que la solución del sistema de ecuaciones diferenciales lineales (10.17) tiene por solución:

$$\underline{X}(t) = e^{\underline{A}(t-t_0)} \underline{X}_0 + \int_{t_0}^t e^{\underline{A}(t-\sigma)} \underline{B} \underline{U}(\sigma) d\sigma \quad (10.18)$$

donde la matriz $e^{\underline{A}(t - t_0)}$ es la matriz de transición definida en la sección 10.3.2.

Por lo tanto la solución total será la suma de la respuesta debida a las condiciones iniciales más la respuesta debida a las excitaciones externas. Para la primera parte de la solución se discutió su obtención en la sección 10.3.2.

Dado que el primer término de la solución se evalúa mediante una evolución de estados a incrementos iguales de tiempo, la segunda parte de la solución:

$$\int_{t_0}^t e^{\underline{A}(t - \sigma)} \underline{B} \underline{u}(\sigma) d\sigma \quad (10.19)$$

también se evaluará a incrementos iguales de tiempo.

Para poder evaluar la expresión (10.19) mediante la computadora se requiere discretizar el vector de entradas $\underline{u}(t)$, aproximando cada entrada $u(t)$ mediante pulsos o rectas como se muestra a continuación:

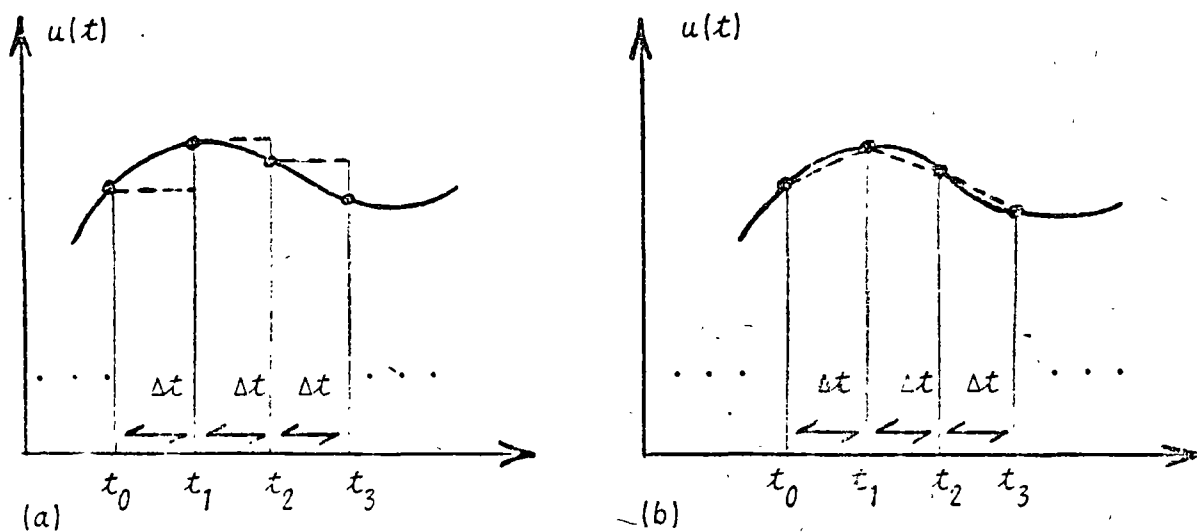


Fig. 10.10 Aproximación de una función mediante:
a) pulsos b) rectas

En el programa se aproxima la función $u(t)$ mediante rectas, las ecuaciones necesarias para la evaluación de (10.19) se desarrollan a continuación.

Sea la función $u(t)$ mostrada en la figura 10.11 :

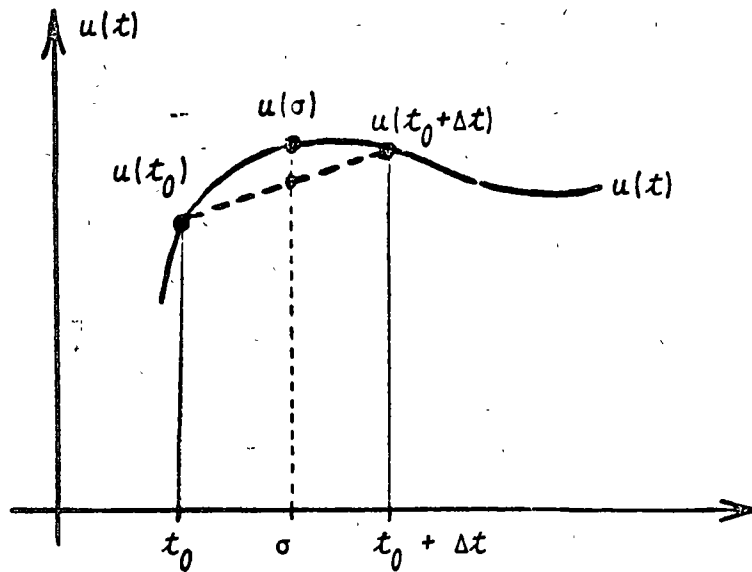


Fig. 10.11 Función $u(t)$ y su aproximación mediante una recta en el intervalo t_0 a $t_0 + t$

Se desea evaluar la expresión (10.19) pero:

$$\int_{t_0}^t e^{\underline{A}(t-\sigma)} \underline{B} \underline{u}(\sigma) d\sigma = e^{\underline{A}t} \int_{t_0}^t e^{-\underline{A}\sigma} \underline{B} \underline{u}(\sigma) d\sigma \quad (10.20)$$

evaluando de t_0 a $t_0 + \Delta t$:

$$\int_{t_0}^{t_0 + \Delta t} e^{\underline{A}(t-\sigma)} \underline{B} \underline{u}(\sigma) d\sigma = e^{\underline{A} \Delta t} \int_0^{\Delta t} e^{-\underline{A}\sigma} \underline{B} \underline{u}(\sigma) d\sigma \quad (10.21)$$

de la figura 10.11 se observa que:

$$\underline{u}(\sigma) \cong \frac{\underline{u}(t_0 + \Delta t) - \underline{u}(t_0)}{\Delta t} + \underline{u}(t_0) \quad (10.22)$$

substituyendo (10.22) en (10.21):

$$e^{\underline{A} \Delta t} \int_0^{\Delta t} e^{-\underline{A}\sigma} \underline{B} \underline{u}(\sigma) d\sigma = \left\{ e^{\underline{A} \Delta t} \int_0^{\Delta t} e^{-\underline{A}\sigma} d\sigma \right\} \cdot \underline{B} \frac{\underline{u}(t_0 + \Delta t) - \underline{u}(t_0)}{\Delta t} + \left\{ e^{\underline{A} \Delta t} \int_0^{\Delta t} e^{-\underline{A}\sigma} d\sigma \right\} \underline{B} \underline{u}(t_0) \quad (10.23)$$

empleando la siguiente relación:

$$e^{-A\sigma} = \underline{I} - \frac{A\sigma}{1!} + \frac{A^2 \sigma^2}{2!} - \frac{A^3 \sigma^3}{3!} + \dots \quad (10.24)$$

en los términos entre corchetes se llega a:

$$e^{\underline{A} \Delta t} \int_0^{\Delta t} e^{-\underline{A}\sigma} \sigma d\sigma = \frac{\underline{I}(\Delta t)^2}{2!} + \dots + \frac{\underline{A}^n (\Delta t)^{n+2}}{(n+2)!} + \dots \quad (10.25)$$

$$e^{\underline{A} \Delta t} \int_0^{\Delta t} e^{-\underline{A}\sigma} d\sigma = \underline{I}(\Delta t) + \dots + \frac{\underline{A}^n (\Delta t)^{(n+1)}}{(n+1)!} + \dots \quad (10.26)$$

Para la evaluación de las series (10.25) y (10.26) la cantidad de términos a emplear dependerá de la exactitud deseada. Se fija un criterio de convergencia ϵ tal que si \underline{Z} representa a la matriz de la serie (10.25) y \underline{W} a la matriz de la serie (10.26), se cumpla que:

$$\begin{aligned} \left| z_{ij}^{(n+1)} - z_{ij}^{(n)} \right| &< \epsilon, \text{ para toda } ij \\ \left| w_{ij}^{(n+1)} - w_{ij}^{(n)} \right| &< \epsilon, \text{ para toda } ij \end{aligned} \quad * \quad (10.27)$$

Como las series (10.25) y (10.26) solo dependen del espaciamiento, se tendrán que evaluar una sola vez.

En el programa para obtener la relación (10.24) hay que evaluar el vector de entradas $\underline{U}(t)$ en cada uno de los puntos en que se subdivide el intervalo de integración.

En términos generales el proceso a seguir es:

- ① evaluar la matriz de transición $e^{\underline{A}(\Delta t)}$
- ② evaluar las series de las ecuaciones (10.25) y (10.26)
- ③ obtener la respuesta debida a las condiciones iniciales para t_i

* $z_{ij}^{(n)}$ representa el elemento z_{ij} de la matriz \underline{Z} compuesta por la sumatoria de "n" términos.

- ④ evaluar $u(t)$ en t_i y t_{i+1}
- ⑤ obtener la respuesta debida a las excitaciones externas mediante la relación (10.23)
- ⑥ hacer $i=i+1$ y regresar al paso ③ hasta barrer todo el intervalo de integración.

10.4.3 Descripción del Programa

a) Subrutinas requeridas:

SUBROUTINE EXPMA(DELTA, M, A, EXPO), obtiene la matriz de transición. Consultar sección 10.3.3.

SUBROUTINE INTPE(DELTA, M, A, SUMA), obtiene la matriz de la serie (10.26); esta expresión se emplea para evaluar la respuesta debida a las excitaciones externas

SUBROUTINE INTRE(DELTA, M, A, RECTA), evalúa la expresión dada por la serie de la ecuación (10.25); esta expresión se utiliza para calcular la respuesta debida a las excitaciones externas.

SUBROUTINE MULTMA(A, B, N, M, L, X), obtiene el producto matricial AB. Consultar el capítulo 2.

SUBROUTINE GRAFI(A, N, M), grafica las soluciones de las variables dependientes y de las respuestas del sistema. Consultar el capítulo 1.

SUBROUTINE EXCITA(T, F), evalúa el vector de entradas $u(t)$ en el instante t_i .

b) Descripción de las variables:

Para la subrutina INTPE:

DELTA espaciamiento entre los valores de la variable independiente

M cantidad de ecuaciones diferenciales

A(I, J) matriz de coeficientes constantes del sistema de ecuaciones diferenciales

EPS criterio de convergencia

SUMA(I, J) matriz resultante de evaluar la serie

CN contador de iteraciones

DIV factorial divisor

TNEW incremento de la variable independiente elevado a la potencia "n"

CMA(I, J) matriz identidad

B(I, J) matriz A elevada a la potencia "n"
 X(I, J) matriz resultante del producto AB

Para la subrutina INTRE:

DELT espaciamento entre los valores de la variable independiente

M cantidad de ecuaciones diferenciales

A(I, J) matriz de coeficientes del sistema de ecuaciones diferenciales

RECTA(I, J) matriz resultante de evaluar la serie

CMA(I, J) matriz identidad

B(I, J) matriz A elevada a la potencia "n"

X(I, J) matriz resultante del producto AB

EPS criterio de convergencia

TNEW incremento de la variable independiente elevado a la potencia "n"

CN contador

DIV factorial divisor

Para la subrutina EXCITA:

T valor del instante de tiempo, en el cual se desea evaluar la expresión $\underline{U}(t)$

F(1, 1) valor del primer renglón de la expresión $\underline{U}(t)$ en el instante t_i

F(2, 1) valor del segundo renglón de la expresión $\underline{U}(t)$ en el instante t_i

F(3, 1) valor del tercer renglón de la expresión $\underline{U}(t)$ en el instante t_i

F(4, 1) valor del cuarto renglón de la expresión $\underline{U}(t)$ en el instante t_i

F(5, 1) valor del quinto renglón de la expresión $\underline{U}(t)$ en el instante t_i

Para el programa principal:

M cantidad de ecuaciones diferenciales

N cantidad de subintervalos en que se divide el intervalo de integración

NS cantidad de salidas del sistema

NU cantidad de entradas del sistema

PERIO intervalo de integración

DELT	magnitud de los subintervalos de integración
A(I, J)	matriz <u>A</u> del sistema de ecuaciones
B(I, J)	matriz <u>B</u> del sistema de ecuaciones
C(I, J)	matriz <u>C</u> del sistema de ecuaciones
D(I, J)	matriz <u>D</u> del sistema de ecuaciones
X(1, 1)	condición inicial de la variable independiente
X(1, J)	condiciones iniciales para cada una de las variables dependientes, $J > 1$
X(I, J)	solución del sistema de ecuaciones
VV(I, J)	valor de las salidas del sistema
SUMA(I, J)	integral del término constante de la ecuación de la recta empleada para aproximar la entrada
RECTA(I, J)	integral del término variable de la ecuación de la recta empleada para aproximar la entrada
SHOM(I, J)	solución del sistema debida a las excitaciones externas
V(I, J)	valor de la entrada en el instante t_{i-1}
PEND(I, J)	pendiente de la recta empleada para aproximar la entrada
RE(I, J)	variable de reemplazo
F(I, J)	variable de reemplazo

c) Dimensiones:

La proposición DIMENSION del programa principal y de las subrutinas deberá modificarse cuando:

$N > 100$ y/o $M > 5$ y/o $NS > 5$ y/o $NU > 5$

Si se modifica la extensión de M, deberán modificarse los argumentos de la subrutina EXCITA.

d) Formatos para los datos de entrada:

SEC. TARJETAS	FORMATO	INFORMACION
1	(4I5, F10.0)	M, N, NS, NU, PERIO
2	(8F10.0)	A(I, J), los elementos de la matriz se dan renglón por renglón. Emplear tan-

.
 .
 .
 3 (8F10.0) B(I,J), igual que para la
 . matriz A
 .
 .
 4 (8F10.0) C(I,J), igual que para la
 . matriz A
 .
 .
 5 (8F10.0) D(I,J), igual que para la
 . matriz A
 .
 .
 6 (8F10.0) X(1,J), el primer valor de
 . be corresponder a la condi-
 . ción inicial de la varia-
 . ble independiente.

 otros paquetes de datos (opcional)

n TARJETA EN BLANCO, al fina-
 lizar toda la información.

e) Diagrama de bloques:

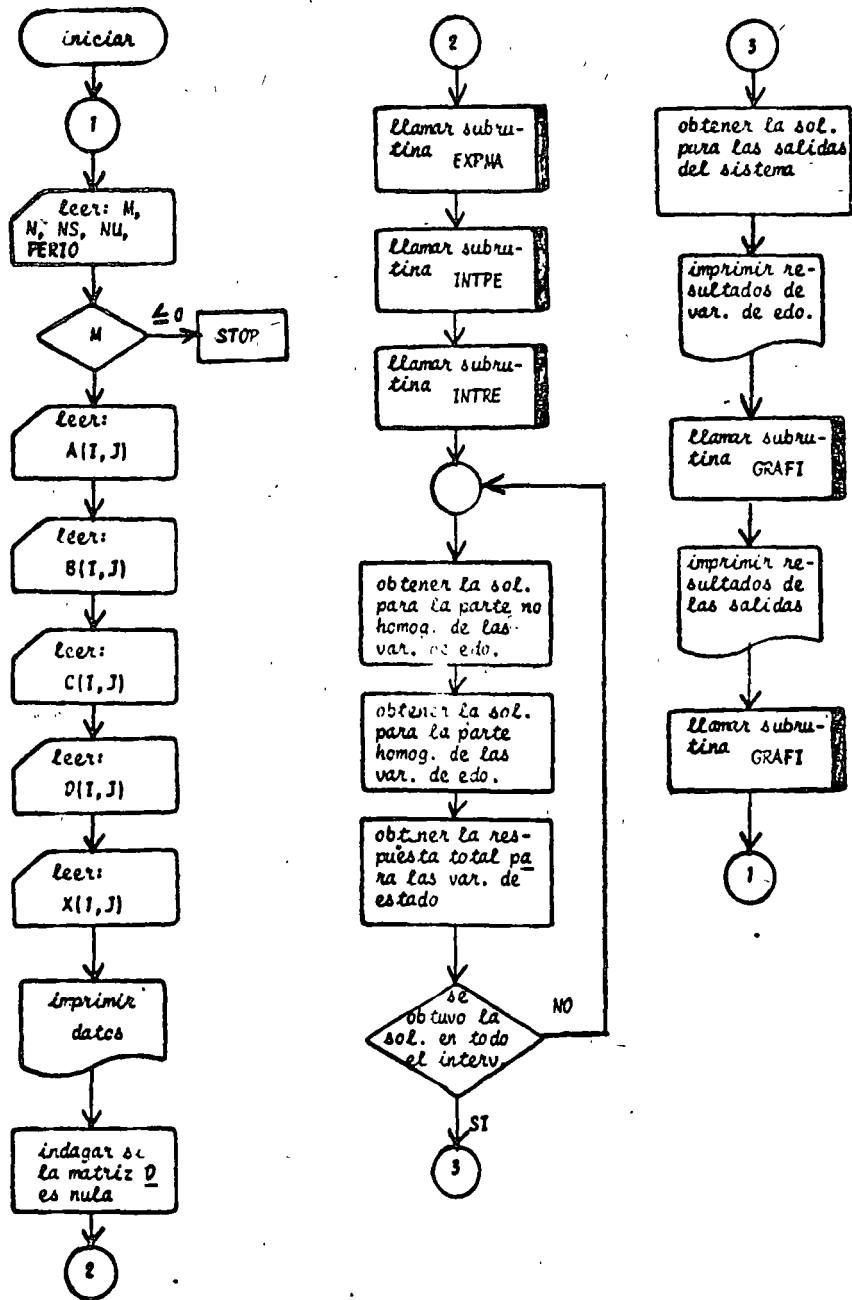


Fig. 10.12 Diagrama de bloques del programa principal

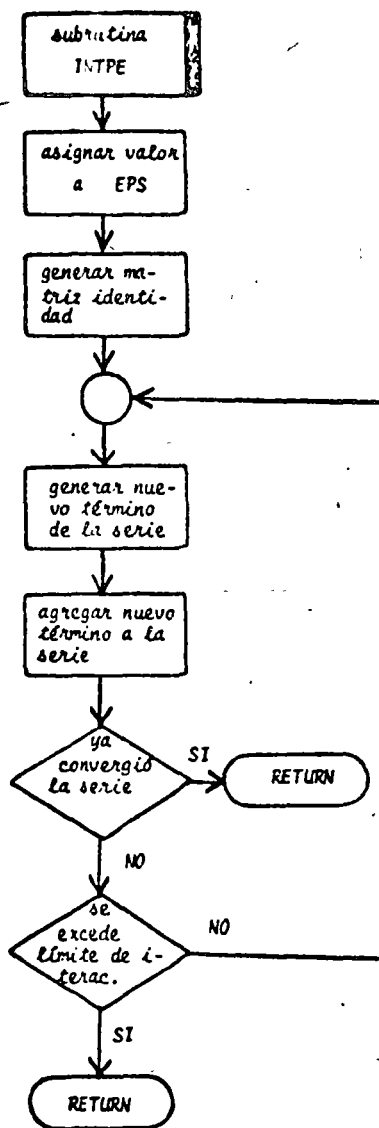


Fig. 10.13 Diagrama de bloques de la subrutina INTPE

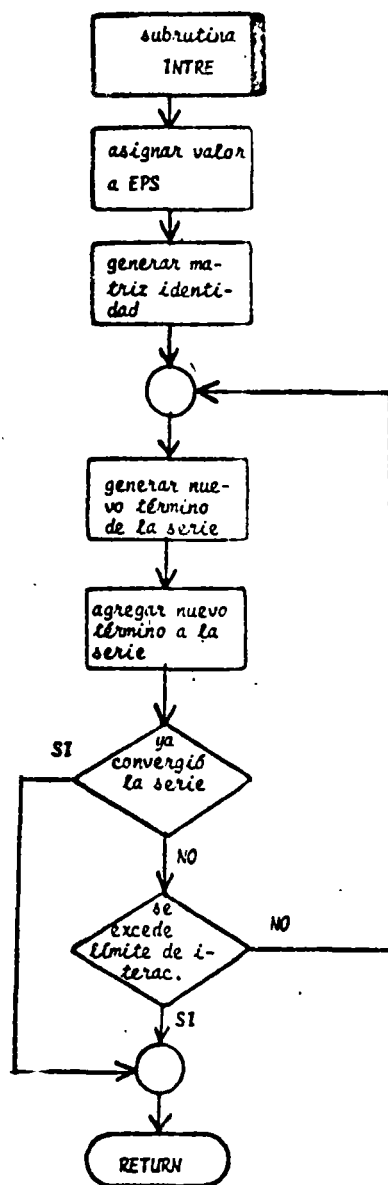


Fig. 10.14 Diagrama de bloques de la subrutina INTRE

6) Listado:

```

C   PROGRAMA PARA SIMULAR SISTEMAS DINAMICOS LINEALES EN UNA COMPUTA-
C   DORA DIGITAL MEDIANTE EL METODO DE VARIACION DE PARAMETROS
C   SIGNIFICADO DE LAS VARIABLES EMPLEADAS
C   M=CANTIDAD DE ECUACIONES DIFERENCIALES
C   NS=CANTIDAD DE PARTES EN QUE SE SUBDIVIDE EL INTERVALO DE INTEGRA-
C   CION
C   NS=CANTIDAD DE SALIDAS
C   NU=CANTIDAD DE ENTRADAS
C   PERIO=MAGNITUD DEL INTERVALO DE INTEGRACION
C   DELT=MAGNITUD DE LOS SUBINTERVALOS
C   A=MATRIZ A DEL SISTEMA DE ECUACIONES
C   B=MATRIZ B DEL SISTEMA DE ECUACIONES
C   C=MATRIZ C DEL SISTEMA DE ECUACIONES
C   D=MATRIZ D DEL SISTEMA DE ECUACIONES
C   X(1,I)=CONDICION INICIAL DE LA VARIABLE INDEPENDIENTE
C   X(1,J)=CONDICIONES INICIALES
C   X=SOLUCION PARA LAS VARIABLES DE ESTADO
C   Y=VALOR DE LAS SALIDAS
C   SUMA=RESPUESTA DEBIDA A LAS EXCITACIONES EXTERNAS
C   EXPO=MATRIZ DE TRANSICION
C   SUMA=INTEGRAL DE LA PARTE CONSTANTE DE LA ECUACION DE LA RECTA
C   RECTA=INTEGRAL DEL TERMINO VARIABLE DE LA ECUACION DE LA RECTA
C   Y=VALOR DE LA ENTRADA EN EL INSTANTE T(I-1)
C   F=VALOR DE LA ENTRADA EN EL INSTANTE T(I)
C   PE=PENDIENTE DE LA RECTA
C   MPI=CANTIDAD DE COLUMNAS DEL ARREGLO MATRICIAL X
C   RE=MATRIZ DE REEMPLAZO

-----
DIMENSION A(6,6),B(6,6),C(6,6),D(6,6),Y(101,6),EXPO(6,6),RECTA(6,6)
1),SUMA(6,6),Y(6,6),F(6,6),PE(6,6),F(6,6),Y(101,6)
I=5
I=6
C   LECTURA DE DATOS
1  READ(I0,100) M,M,NS,NU,PERIO
   IF(I) 2,2,3
2  CALL EXIT
3  MPI=M+1
   DO 4 I=1,M
4  READ(I0,150) (A(I,J),J=1,M)
   DO 5 I=1,M
5  READ(I0,150) (B(I,J),J=1,M)
   IF(NS.EQ.0) GO TO 71
   DO 6 I=1,NS
6  READ(I0,150) (C(I,J),J=1,M)
   DO 7 I=1,NS
7  READ(I0,150) (D(I,J),J=1,M)
71 READ(I0,150) (X(1,J),J=1,MPI)
C   IMPRESION DE DATOS
   WRITE(I0,200)
   DO 8 I=1,M
8  WRITE(I0,250) (A(I,J),J=1,M)
   WRITE(I0,300)
   DO 9 I=1,M
9  WRITE(I0,250) (B(I,J),J=1,M)
   IF(NS.EQ.0) GO TO 72
   WRITE(I0,350)
   DO 10 I=1,NS
10 WRITE(I0,250) (C(I,J),J=1,M)
   WRITE(I0,400)
   DO 11 I=1,NS
11 WRITE(I0,250) (D(I,J),J=1,M)
72 WRITE(I0,450)
   WRITE(I0,250) (X(1,J),J=1,MPI)
   IF(NS.EQ.0) GO TO 73
C   IMPRIMIR SI LA MATRIZ D ES NULA
   DO 12 I=1,NS
   DO 12 J=1,M
   IF(D(I,J).EQ.0.0) GO TO 12
   IO=1
   GO TO 13
12 CONTINUE
73 IO=0
C   OBTENER LA MATRIZ DE TRANSICION
13 DELT=PERIO/DELTA(M)
   CALL EXPO(DELT,M,A,EXPO)
C   OBTENER SUMA Y RECTA
   CALL SUMA(DELT,M,A,SUMA)
   CALL INTRE(DELT,M,A,RECTA)

```

```

C ... OBTENER LA SOLUCION TOTAL DE LAS VARIABLES DE ESTADO
NP1=1
DO 18 K=2, NP1
X(K,1)=X(K-1,1) + DELT
C ... OBTENER LA SOLUCION DEBIDA A LAS ENTRADAS
T=X(K-1,1)
CALL EXCITA(T,F)
CALL MULTA(D,F,N,HU,1,Y)
T=X(K,1)
CALL EXCITA(T,F)
CALL MULTA(R,F,N,HU,1,RE)
DO 14 I=1,N
14 PE(I,1)=(CF(I,1)-Y(I,1))/DELT
CALL MULTA(SU(A),Y,I,N,1,SHON)
CALL MULTA(RECTA,PE,D,N,H,1,RF)
DO 15 J=1,N
15 SUM(I,1)=SUM(I,1) + RE(I,1)
C ... OBTENER SOLUCION DEBIDA A LAS CONDICIONES INICIALES
DO 17 I=1,N
X(K,J+1)=0,J
DO 16 I=2, NP1
16 X(K,J+1)=X(K,J+1) + EXPO(J,I-1)*X(K-1,1)
C ... OBTENER LA SOLUCION TOTAL
17 X(K,J+1)=X(K,J+1) + SHON(J,1)
18 CONTINUE
IF(NS.F0.0) GO TO 30
C ... OBTENER EL VALOR DE LAS SALIDAS PARA TODOS LOS PUNTOS MUESTRALES
IF(ID.F0.0) GO TO 26
DO 21 K=1, NP1
DO 25 I=1, NS
25 YY(K,I)=0,
T=X(K,1)
CALL EXCITA(T,F)
CALL MULTA(D,F,NS,HU,1,A)
DO 31 I=1, NS
31 YY(K,I)=YY(K,I) + A(I,1)
DO 19 I=1,N
19 PE(I,1)=X(K,I+1)
CALL MULTA(C,PE,NS,H,1,A)
DO 20 I=1, NS
20 YY(K,I)=YY(K,I) + A(I,1)
21 CONTINUE
GO TO 30
26 DO 29 K=1, NP1
DO 27 I=1,N
27 RE(I,1)=X(K,I+1)
CALL MULTA(C,RE,NS,H,1,A)
DO 28 I=1, NS
28 YY(K,I)=YY(K,I) + A(I,1)
29 CONTINUE
C ... IMPRIMIR RESULTADOS CORRESPONDIENTES A LAS VARIABLES DE ESTADO
30 WRITE(I,500)
DO 22 J=1, NP1
22 WRITE(IH,250) (X(I,J),J=1, NP1)
CALL GRAF(X, PL, NP1)
IF(NS.F0.0) GO TO 1
C ... IMPRIMIR RESULTADOS CORRESPONDIENTES A LAS SALIDAS DEL SISTEMA
WRITE(I,550)
DO 23 J=1, NP1
23 WRITE(I,250) (X(I,1), (YY(I,J), J=1, NS))
DO 24 J=1, NP1
DO 24 J=1, NS
24 X(I,J+1)=YY(I,1)
NS=NS + 1
CALL GRAF(X, NP1, NS)
GO TO 1
C ... FORMATOS DE LECTURA E IMPRESION
100 FORMAT(4F10.0)
150 FORMAT(A10.0)
200 FORMAT(1=1.5(//,15X,'MATRIZ A',//)
250 FORMAT(/,15X,0(12.5,3X))
300 FORMAT(5(//,15X,'MATRIZ B',//)
350 FORMAT(5(//,15X,'MATRIZ C',//)
400 FORMAT(5(//,15X,'MATRIZ D',//)
450 FORMAT(5(//,15X,'LAS CONDICIONES INICIALES SON',//,18X,'TIEMPO',10X
1, X(1)',11X, X(2)',11X, X(3)',11X, X(4)',11X, X(5)',//)
500 FORMAT(5(//,15X,'LA SOLUCION PARA LAS VARIABLES DE ESTADO ES',//,18
1X, 'TIEMPO',10X, X(1)',11X, X(2)',11X, X(3)',11X, X(4)',11X, X(5)',
//)
550 FORMAT(5(//,15X,'EL VALOR DE LAS SALIDAS ES',//,18X,'TIEMPO',10X, Y
1(1)',11X, Y(2)',11X, Y(3)',11X, Y(4)',11X, Y(5)',//)
END

```

Fig. 10.15 Listado del programa principal

```

SUBROUTINE INTPE(DELTA,M,A,SUM)
C      SUBROUTINA PARA EVALUAR LA INTEGRAL DEL TERMINO CONSTANTE DE LA
C      RECTA
C      SIGNIFICADO DE LAS VARIABLES EMPLEADAS
C      DELTA=ESPACIAMIENTO ENTRE LOS VALORES DE LA VARIABLE INDEPENDIENTE
C      M=CANTIDAD DE ECUACIONES DIFERENCIALES
C      A=MATRIZ A DEL SISTEMA DE ECUACIONES
C      SUMA=MATRIZ RESULTANTE DE EVALUAR LA INTEGRAL
C      EPS=CRITERIO DE CONVERGENCIA
C      C=MATRIZ IDENTIDAD
C      C=CONTADOR DE ITERACIONES
C      DIV=FACTOREAL DIVISOR
C      T=INCREMENTO DE TIEMPO ELEVADO A LA POTENCIA N
C      B=MATRIZ A ELEVADA A LA POTENCIA N
C      X=MATRIZ RESULTANTE DEL PRODUCTO AB
C      DIMENSION A(6,6),C(6,6),B(6,6),X(6,6),SUMA(6,6)
C      EPS=0.00001
C      GENERAR MATRIZ IDENTIDAD
C      DO 3 I=1,M
C      DO 2 J=1,M
C      IF(I.EQ.J) GO TO 1
C      C(I,J)=0.0
C      GO TO 2
1      C(I,J)=1.0
2      CONTINUE
3      CONTINUE
C      OBTENER LOS DOS PRIMEROS TERMINOS DE LA SERIE
C      DO 4 I=1,M
C      DO 4 J=1,M
C      SUMA(I,J)=C(I,J)*DELTA + (A(I,J)*DELTA*DELTA)/2.0
4      B(I,J)=A(I,J)
C      T=DELTA*DELTA
C      C=3.0
C      DIV=2.0
C      OBTENER LOS TERMINOS RESTANTES DE LA SERIE
5      DIV=DIV*C
C      T=T*DELTA*DELTA
C      CALL MULTM(A,B,M,M,M,X)
C      DO 6 I=1,M
C      DO 6 J=1,M
6      B(I,J)=X(I,J)
C      C(I,J)=(X(I,J)+T*DELTA)/DIV
C      IF(C=6.0) 9,9,13
13      AMAX=C(I,1)
C      DO 8 I=1,M
C      DO 7 J=1,M
C      IF(ABS(C(I,J))-ABS(AMAX)) 7,7,14
14      AMAX=C(I,J)
7      CONTINUE
8      CONTINUE
C      VERIFICAR LA CONVERGENCIA DE LA SERIE
C      IF(ABS(AMAX)-EPS) 11,11,9
9      DO 10 I=1,M
C      DO 10 J=1,M
10      SUMA(I,J)=SUMA(I,J) + C(I,J)
C      IF(C=20.0) 15,15,11
15      C=C+1.0
C      GO TO 5
11      RETURN
C      END

```

Fig. 10.16 Listado de la subrutina INTPE

```

SUBROUTINE INTRE(DELTA,M,A,RECTA)
C SUBROUTINA PARA EVALUAR LA INTEGRAL DE LA PARTE VARIABLE DE LA 2ª
C CURCION DE LA RECTA
C SIGNIFICADO DE LAS VARIABLES EMPLEADAS
C DELTA=ESPACIAMIENTO ENTRE LOS VALORES DE LA VARIABLE INDEPENDIENTE
C RECHIZIDAD DE ECUACIONES DIFERENCIALES
C A=MATRIZ A DEL SISTEMA DE ECUACIONES
C RECTA=MATRIZ RESULTANTE DE EVALUAR LA SERIE
C CMA=MATRIZ IDENTIDAD
C B=MATRIZ A ELEVADA A LA POTENCIA II
C X=MATRIZ RESULTANTE DEL PRODUCTO AB
C EPS=CRITERIO DE CONVERGENCIA
C T=INCREMENTO DE LA VARIABLE INDEPENDIENTE ELEVADO A LA POTENCIA
C II
C LIMITE=CONTADOR DE ITERACIONES
C DIV=FACTORIAL DIVISOR
C DIMENSION A(6,6),RECTA(6,6),CMA(6,6),B(6,6),X(6,6)
C EPS=0.00001
C GENERAR MATRIZ IDENTIDAD
C 1 3 I=1,M
C 2 2 J=1,M
C IF(1.E9,J) GO TO 1
C CMA(I,J)=0.0
C GO TO 2
C 1 CMA(I,J)=1.0
C 2 CONTINUE
C 3 CONTINUE
C OBTENER LOS DOS PRIMEROS TERMINOS DE LA SERIE
C DO 4 I=1,M
C DO 4 J=1,M
C RECTA(I,J)=(CMA(I,J)*DELTA*DELTA)/2.0 + (A(I,J)*(DELTA**3))/6.0
C 4 R(I,J)=A(I,J)
C T=DELTA*DELTA*DELTA
C C=4.0
C DIVER=0
C OBTENER EL RESTO DE LOS TERMINOS DE LA SERIE
C 5 DIV=DIVER*C
C T=DELTA*DELTA*DELTA
C CALL MULT(A,B,C,M,M,X)
C DO 6 I=1,M
C DO 6 J=1,M
C R(I,J)=X(I,J)
C 6 CMA(I,J)=(X(I,J)*T)/DIV
C IF(C*LE.6.0) GO TO 9
C AMAX=CMA(1,1)
C DO 8 I=1,M
C DO 8 J=1,M
C IF(ABS(CMA(I,J)).LE.ABS(AMAX)) GO TO 7
C AMAX=CMA(I,J)
C 7 CONTINUE
C 8 CONTINUE
C VERIFICAR CONVERGENCIA DE LA SERIE
C IF(ABS(AMAX).LE.EPS) GO TO 11
C 9 DO 10 I=1,M
C DO 10 J=1,M
C 10 RECTA(I,J)=RECTA(I,J) + CMA(I,J)
C IF(C*GT.20.) GO TO 11
C C=C + 1.0
C GO TO 5
C 11 RETURN
C END

```

Fig. 10.17 Listado de la subrutina INTRE

```

SUBROUTINE EXCITA(T,F)
C DIMENSION F(6,6)
C F(1,1)=5.*EXP(-4.*T)
C F(2,1)=0.5*SIN(3.*T)
C F(3,1)=0.0
C F(4,1)=0.0
C F(5,1)=0.0
C RETURN
C END

```

Fig. 10.18 Listado de la subrutina EXCITA

10.4.4 Ejemplo

Para el siguiente circuito eléctrico:

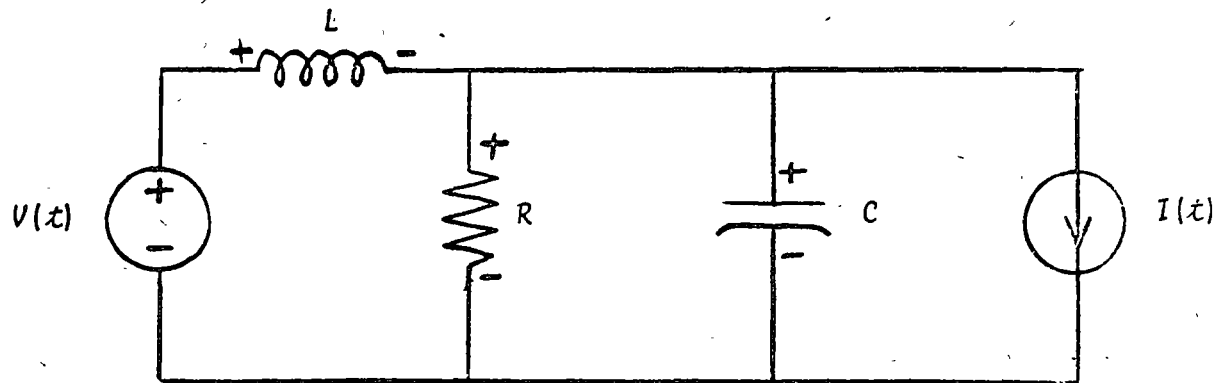


Fig. 10.19 Circuito eléctrico del problema del ejemplo 10.4.4

si se consideran como salidas I_c y V_R , su representación mediante variables de estado es:

$$\begin{bmatrix} \frac{dI_L}{dt} \\ \frac{dV_c}{dt} \end{bmatrix} = \begin{bmatrix} 0 & -1/L \\ 1/C & -1/RC \end{bmatrix} \begin{bmatrix} I_L \\ V_c \end{bmatrix} + \begin{bmatrix} 1/L & 0 \\ 0 & -1/C \end{bmatrix} \begin{bmatrix} V(t) \\ I(t) \end{bmatrix}$$

$$\begin{bmatrix} I_c \\ V_R \end{bmatrix} = \begin{bmatrix} 1 & -1/R \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I_L \\ V_c \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V(t) \\ I(t) \end{bmatrix}$$

Determine los valores de I_L , V_c , I_c y V_R para $t \geq 0$, cuando:

$$V(t) = 5e^{-4t}u_{-1}(t) \quad (V)$$

$$I(t) = 0.5\delta\text{en}(3t)u_{-1}(t) \quad (A)$$

$$R = 100 \text{ ohms}$$

$$C = 0.1 \text{ F}$$

$$L = 1.0 \text{ H}$$

$$t_0 = 0.$$

$$V_C(t_0) = 2 \text{ V}$$

$$I_L(t_0) = 0.3 \text{ A}$$

$$t_f = 10 \text{ s}$$

* SOLUCION

TABLA 10.5 Datos para el problema del ejemplo 10.4.4

$$M = 2$$

$$N = 100$$

$$NS = 2$$

$$NU = 2$$

$$\text{PERIO} = 10$$

$$A = \begin{bmatrix} 0 & -1 \\ 10 & -0.1 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 \\ 0 & -10 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & -0.01 \\ 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}$$

$$X(1, J) = (0 , 0.3 , 2)$$

$$F(1, 1) = 5.*\text{EXP}(-4.*T)$$

$$F(2,1) = 0.5 * \text{SIN}(3.0 * T)$$

$$F(3,1) = 0.$$

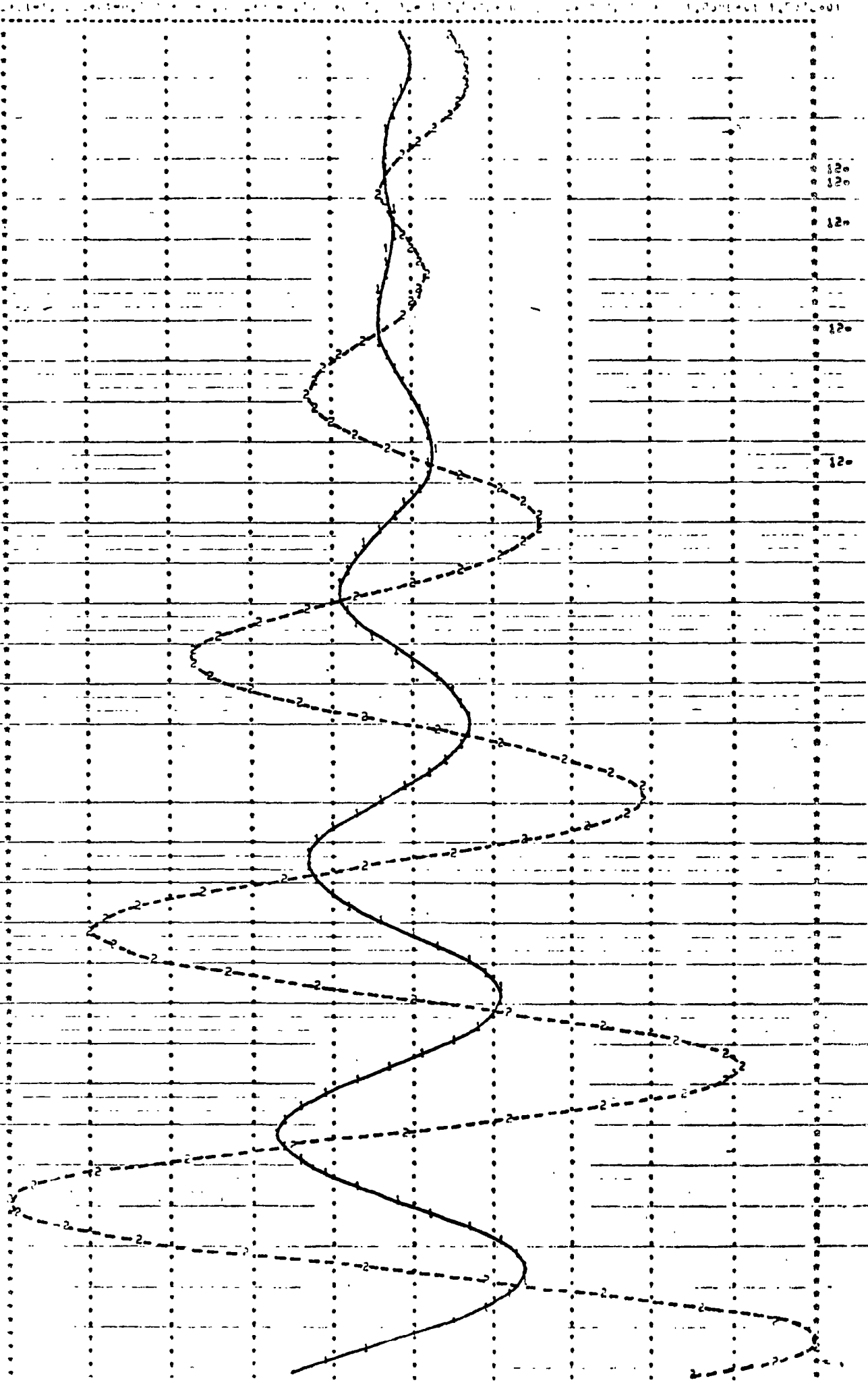
$$F(4,1) = 0.$$

$$F(5,1) = 0.$$

TABLA 10.6 Resultados del problema del ejemplo 10.4.4

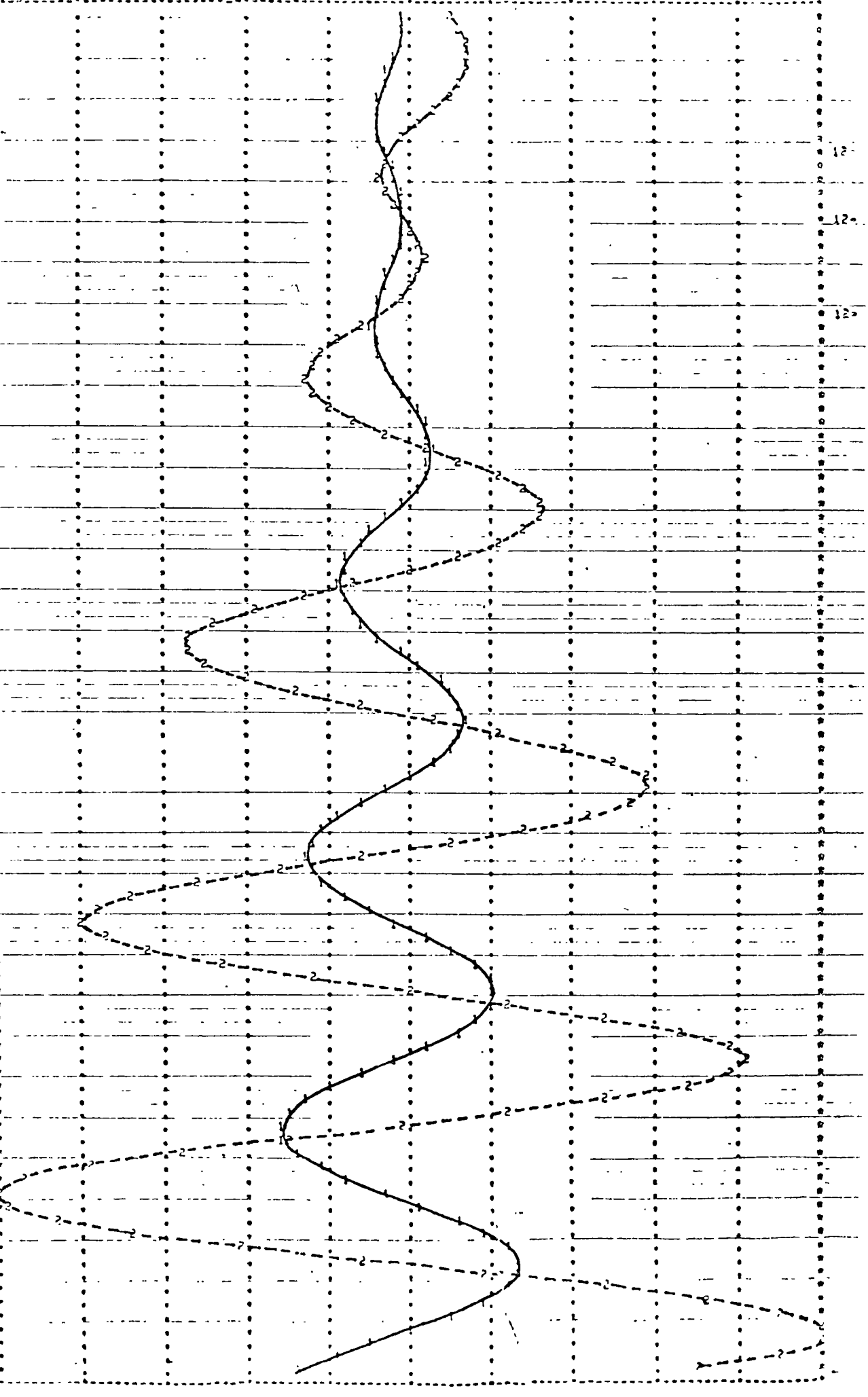
LA SOLUCION PARA LAS VARIABLES DE ESTADO EN
TIEMPO X(1) X(2)

0.	.30000E+00	.20000E+01
.10000E+00	.50191E+00	.23220E+01
.20000E+00	.53425E+00	.26119E+01
.30000E+00	.45225E+00	.27485E+01
.40000E+00	.30503E+00	.26745E+01
.50000E+00	.13400E+00	.23869E+01
.60000E+00	-.25673E-01	.19248E+01
.70000E+00	-.15241E+00	.13566E+01
.80000E+00	-.23291E+00	.76453E+00
.90000E+00	-.26486E+00	.23115E+00
.10000E+01	-.25499E+00	-.17406E+00
.11000E+01	-.21693E+00	-.43443E+00
.12000E+01	-.16776E+00	-.44242E+00
.13000E+01	-.12577E+00	-.30147E+00
.14000E+01	-.10631E+00	-.23558E+01
.15000E+01	-.11965E+00	.32723E+00
.16000E+01	-.16905E+00	.67420E+00
.17000E+01	-.25002E+00	.93923E+00
.18000E+01	-.35072E+00	.10544E+01
.19000E+01	-.45351E+00	.97213E+00
.20000E+01	-.53738E+00	.67331E+00
.21000E+01	-.58102E+00	.17120E+00
.22000E+01	-.56612E+00	-.49849E+00
.23000E+01	-.48037E+00	-.12319E+01
.24000E+01	-.31998E+00	-.19651E+01
.25000E+01	-.91039E-01	-.25859E+01
.26000E+01	.19019E+00	-.29957E+01
.27000E+01	.49838E+00	-.31137E+01
.	.	.
.88000E+01	.13221E+01	-.13823E+02
.89000E+01	.26339E+01	-.12196E+02
.90000E+01	.37213E+01	-.93734E+01
.91000E+01	.44771E+01	-.59132E+01
.92000E+01	.45239E+01	-.12545E+01
.93000E+01	.47215E+01	.32984E+01
.94000E+01	.41721E+01	.76164E+01
.95000E+01	.32299E+01	.11295E+02
.96000E+01	.19038E+01	.13946E+02
.97000E+01	.47348E+01	.15353E+02
.98000E+01	-.16695E+01	.15297E+02
.99000E+01	-.25350E+01	.13778E+02
.10000E+02	-.37140E+01	.10994E+02



EL VALOR DE LAS SALIDAS ES		
Tiempo	Y(1)	Y(2)
0.	.29000E+00	.20000E+01
.10000E+00	.33003E+00	.23270E+01
.20000E+00	.22931E+00	.26119E+01
.30000E+00	.31105E+01	.27485E+01
.40000E+00	-.15773E+01	.26745E+01
.50000E+00	-.33902E+00	.23969E+01
.60000E+00	-.53124E+01	.19238E+01
.70000E+00	-.59759E+00	.13566E+01
.80000E+00	-.57828E+00	.76453E+00
.90000E+00	-.43086E+00	.23115E+01
.10000E+01	-.32381E+00	-.17406E+00
.11000E+01	-.13392E+00	-.40443E+00
.12000E+01	.57922E+01	-.44242E+00
.13000E+01	.22113E+00	-.30147E+00
.14000E+01	.32972E+00	-.23598E+01
.15000E+01	.36535E+00	.32723E+00
.16000E+01	.32229E+00	.67420E+00
.17000E+01	.20350E+00	.93973E+00
.18000E+01	.25120E+01	.10594E+01
.19000E+01	-.18739E+00	.97213E+00
.20000E+01	-.40441E+00	.67331E+00
.21000E+01	-.59114E+00	.17120E+00
.22000E+01	-.71700E+00	-.49849E+00
.23000E+01	-.75727E+00	-.12311E+01
.24000E+01	-.69716E+00	-.19611E+01
.25000E+01	-.53418E+00	-.25359E+01
.26000E+01	-.27913E+00	-.20957E+01
.27000E+01	.44537E+01	-.31137E+01
.28000E+01	.40303E+00	-.28877E+01
.29000E+01	.75556E+00	-.23033E+01
.30000E+01	.10589E+01	-.13882E+01
.31000E+01	.12724E+01	-.21239E+00
.	.	.
.92000E+01	.45247E+01	-.12545E+01
.93000E+01	.45057E+01	.32984E+01
.94000E+01	.40588E+01	.76164E+01
.95000E+01	.32100E+01	.11285E+02
.96000E+01	.21622E+01	.13946E+02
.97000E+01	.64767E+00	.15330E+02
.98000E+01	-.77116E+00	.15247E+02
.99000E+01	-.21788E+01	.13702E+02
.10000E+02	-.34007E+01	.10944E+02

2.00E+00
2.10E+00
2.20E+00
2.30E+00
2.40E+00
2.50E+00
2.60E+00
2.70E+00
2.80E+00
2.90E+00
3.00E+00
3.10E+00
3.20E+00
3.30E+00
3.40E+00
3.50E+00
3.60E+00
3.70E+00
3.80E+00
3.90E+00
4.00E+00
4.10E+00
4.20E+00
4.30E+00
4.40E+00
4.50E+00
4.60E+00
4.70E+00
4.80E+00
4.90E+00
5.00E+00
5.10E+00
5.20E+00
5.30E+00
5.40E+00
5.50E+00
5.60E+00
5.70E+00
5.80E+00
5.90E+00
6.00E+00
6.10E+00
6.20E+00
6.30E+00
6.40E+00
6.50E+00
6.60E+00
6.70E+00
6.80E+00
6.90E+00
7.00E+00
7.10E+00
7.20E+00
7.30E+00
7.40E+00
7.50E+00
7.60E+00
7.70E+00
7.80E+00
7.90E+00
8.00E+00
8.10E+00
8.20E+00
8.30E+00
8.40E+00
8.50E+00
8.60E+00
8.70E+00
8.80E+00
8.90E+00
9.00E+00
9.10E+00
9.20E+00
9.30E+00
9.40E+00
9.50E+00
9.60E+00
9.70E+00
9.80E+00
9.90E+00
1.00E+01



12
12
12

10.5 Bibliografía

1. CANALES Roberto y BARRERA Renato, "Apuntes de Ingeniería de Control I". México: Fac. de Ingeniería, UNAM, 1973.
pp.1-36 (cap. 3), 1-37 (cap. 5).
2. CARNAHAN B., LUTHER H., WILKES J., "Applied Numerical Methods". New York: John Wiley & Sons Inc., 1969.
pp.361-380.
3. DORF C. Richard, "Modern Control Systems". Reading Mass.: Addison-Wesley Co., 1970.
pp.250-301, 374-379.
4. GEREZ G. Víctor y MURRAY-LASSO M.A., "Teoría de Sistemas y Circuitos". México: Representaciones y Servicios de Ingeniería S.A., 1972.
pp.330-385, 459-463.
5. KUO S. Shan, "Computer Applications of Numerical -- Methods". Reading Mass.: Addison-Wesley Co., 1972.
pp.128-166.
6. OGATA Katsuhiko, "Modern Control Theory". Englewood Cliffs N.J.: Prentice-Hall Inc., 1970.
pp.663-704.
7. SMITH G., JAMES M., WOLFORD J., "Applied Numerical Methods for Digital Computation with FORTRAN". Scranton Penn.: International Textbook Co., 1967.
pp.350-356.



centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



METODOS NUMERICOS Y APLICACIONES CON LA COMPUTADORA
DIGITAL

C O M P L E M E N T O

SOLUCION NUMERICA DE ECUACIONES DIFERENCIALES

RUTA CRITICA

PROGRAMACION LINEAL.

PROF. M. en C. VERONICA CZITROM.

ABRIL, 1978.

SOLUCIÓN NUMÉRICA ECUACIONES DIFERENCIALES

$$y = ? = y(x)$$

1 SOLA VARIABLE INDEPENDIENTE

EJEMPLO: POBLACIÓN

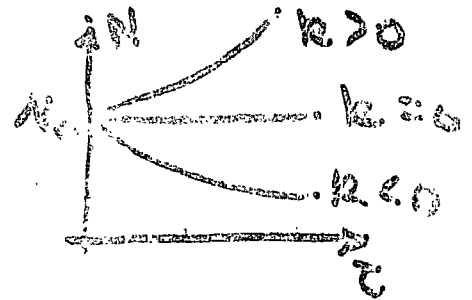
$N = N(t)$ = NUM. HABITANTES TIEMPO t

$$\frac{dN}{dt} \propto N$$

$$\Rightarrow$$

$$\frac{dN}{dt} = k \cdot N$$

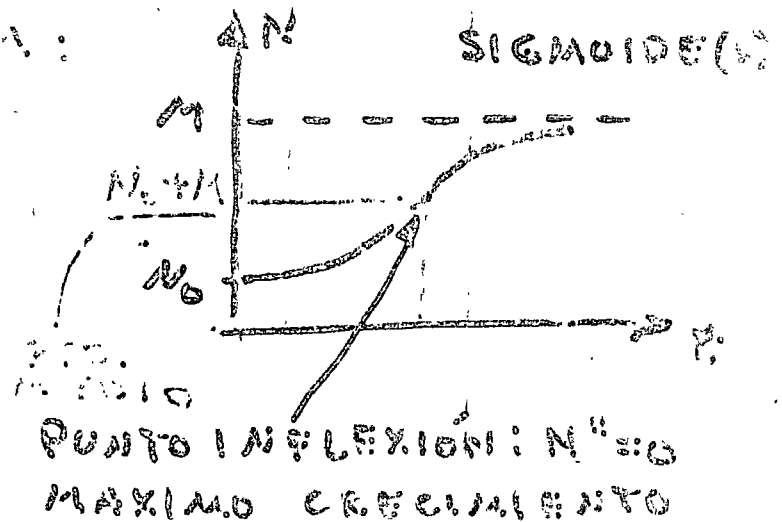
$$N(t) = N_0 \cdot e^{kt}$$



MIX POBLACIÓN M:

$$\frac{dN}{dt} = k \cdot N (M - N)$$

$$N(t) = \frac{M}{1 + C \cdot e^{-kt}}$$



SOLUCIÓN NUMÉRICA ECU. DIF.:

- CUANDO NO EXISTE SOLUCIÓN EXACTA
- SI HAY COMPLICACIONES OBTENERLA

ECUACIÓN DIFERENCIAL:

ORDEN: MAYOR DERIVADA

GRADO: MAYOR POTENCIA DE $y, y', y'' \dots$

$$y'''' - x^5 (y')^2 + (\cos x) y = \cos x$$

ORDEN: 4 (y'''')

GRADO: 2 ($(y')^2$)

EC. DIF. LINEAL:

- NO HAY PRODUCTOS $y^{(n)}$ $y^{(m)}$

- SOLO PRIMERA POTENCIA: $(y^{(n)})'$

EC. DIF. DE ORDEN N: SOLUCION CON TIENE N CONSTANTES ARBITRARIAS.

SOLUCION ÚNICA: EC. DIF. ORDEN N. N CONDICIONES INICIALES

CONDICIONES INICIALES: $y(x_0) = p$ $y'(x_0) = 3$ $y''(x_0) = 5$ \vdots MISMO PUNTO

CONDICIONES A LA FRONTERA: $y(x_0) = p$ $y(x_1) = 3$ DIFERENTES PUNTOS

ECUACIONES DIFERENCIALES LINEALES HOMOGÉNEAS DE COEFICIENTES CONSTANTES

$\frac{d^n y}{dx^n} + a_{n-1} \frac{d^{n-1} y}{dx^{n-1}} + \dots + a_1 \frac{dy}{dx} + a_0 y = 0$ HOMOGÉNEA

SE PROPONE: $y = e^{x \cdot \lambda}$

ES SOLUCION SI:

(SUSTITUYENDO)

$\lambda^n + a_{n-1} \lambda^{n-1} + \dots + a_1 \lambda + a_0 = 0$

POLINOMIO CARACTERISTICO:

RAÍCES:

a) REALES Y DISTINTAS $\alpha_1 \neq \alpha_2 \neq \dots$

$$c_1 e^{\alpha_1 t} + c_2 e^{\alpha_2 t} + \dots$$

CONSTANTES ARBITRARIAS

b) REALES Y REPETIDAS $\alpha_i = \alpha_{i+1} = \dots = \alpha_{i+k}$

$$(c_1 + c_{i+1} x + \dots + c_{i+k} x^{k-1}) e^{\alpha_i t}$$

c) COMPLEJAS CONJUGADAS $A \pm iB$

$$e^{At} (c_2 \cos Bt + c_1 \sin Bt)$$

d) COMPLEJAS CONJUGADAS REPETIDAS

$$A \pm iB, A \pm iB, A \pm iB$$

$$e^{At} (c_1 + c_2 x + c_3 x^2) \cos Bt + e^{At} (c_4 + c_5 x + c_6 x^2) \sin Bt$$

METODO NUMERICO DE SOL. DE

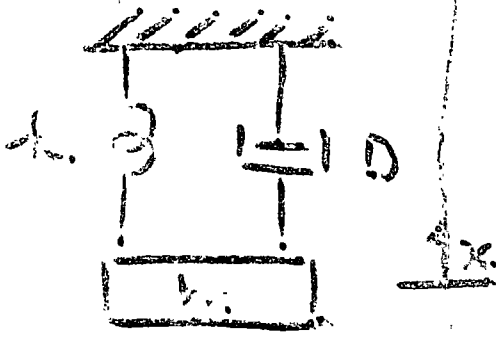
$$y^{(n)} + a_{n-1} y^{(n-1)} + \dots + a_1 y' + a_0 y = 0$$

EVALÚA RAÍCES DE EC. CARACTERÍSTICA

$$\alpha^n + a_{n-1} \alpha^{n-1} + \dots + a_1 \alpha + a_0 = 0$$

EJEMPLO: SISTEMA MECÁNICO

(4)



$$m\ddot{x} + D\dot{x} + kx = 0$$

E.C. CARACTERÍSTICA:

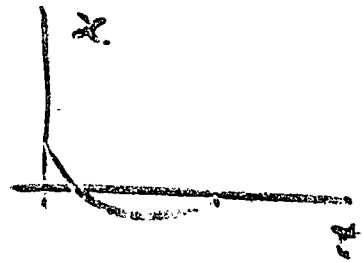
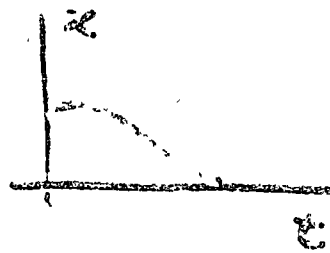
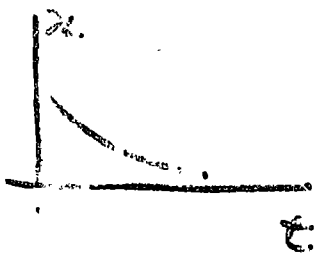
$$m\lambda^2 + D\lambda + k = 0$$

$$\lambda = \frac{-D \pm \sqrt{D^2 - 4mk}}{2m}$$

1) CASO SOBREAAMORTIGUADO ($D^2 - 4mk > 0$)

$$\lambda_1 \neq \lambda_2$$

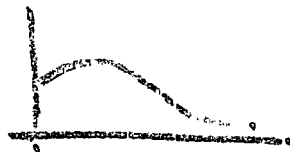
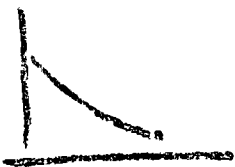
$$\text{sol: } x(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t}$$



2) CASO CRÍTICAMENTE AMORTIGUADO ($D^2 - 4mk = 0$)

$$\alpha_1 = \alpha_2 = -\frac{D}{2m}$$

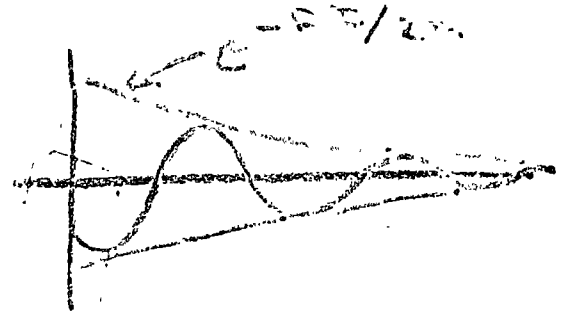
$$\text{sol: } x(t) = c_1 e^{-\frac{D}{2m}t} + c_2 x e^{-\frac{D}{2m}t}$$



3) CASO SUBAMORTIGUADO ($D^2 - 4mk < 0$)

$$\lambda = \underbrace{-\frac{D}{2m}}_A \pm i \underbrace{\frac{\sqrt{4mk - D^2}}{2m}}_B$$

$$\text{sol: } x(t) = e^{-\frac{D}{2m}t} (c_1 \cos(Bt) + c_2 \sin(Bt))$$



EC. DIF. ORDEN $N \rightarrow N$ ECS. DIFS. 1^{er} ORDEN

MÉTODO NUMÉRICO DE EULER

LEONHARD EULER (1707-1783)

MAT. MAS PRÁCTICO SIGLO XVIII
GEOMETRÍA, CÁLCULO, MECÁNICA, TRIG. NUMÉRICA,
LOGARITMO NUMÉRICO, NEGATIVOS Y COMPLEJOS.

NOTACION: $\Sigma, e, f(x), \pi, i$

MÉTODOS ALGEBRAICOS

MOV. LUNA (3 CUERPOS: SOL, LUNA, TIERRA)

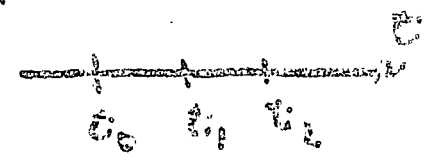
PERDIO VISTA 1 OJO; LUEGO EL OJO

RESOLVER

$$\frac{dy}{dt} = f(t, y)$$

SUJETO A

$$y(t_0) = y_0$$

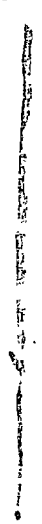


$$\frac{dy}{dt} = y'$$

$$dy = y' dt$$

$$\Delta y = y' \Delta t$$

$$y_{i+1} - y_i = y'_i \Delta t$$



$$y' = f(t, y)$$

$$\int_{t_0}^{t_1} y' dt = \int_{t_0}^{t_1} f(t, y) dt$$

$$y_1 - y_0$$

$$y_1 = y_0 + \int_{t_0}^{t_1} f(t, y) dt$$

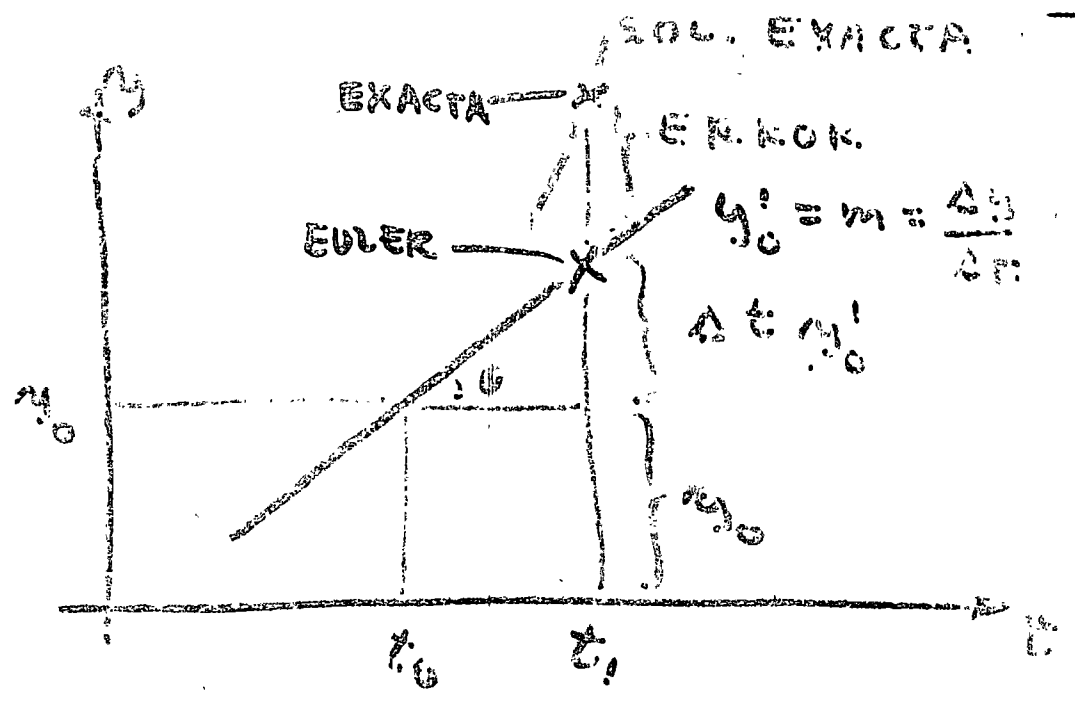
$$= y_0 + \Delta t \cdot f(t_0, y_0)$$

$$y_{i+1} = y_i + \Delta t \cdot f(t_i, y_i)$$

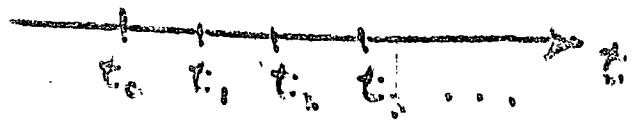
SE RELANCA con $y(t_0) = y_0$

EULER $\sim (\Delta t)^2$

$$y_1 = y_0 + \Delta t \cdot y'_0$$

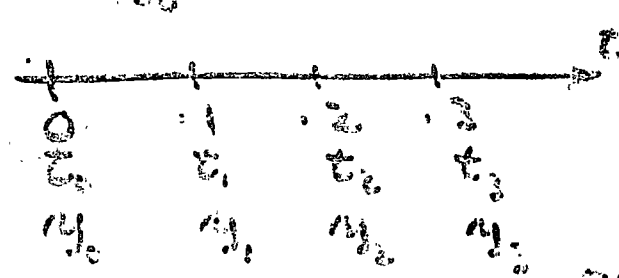


ERRORES CHICOS $\Rightarrow \Delta t$ CHICA \Rightarrow MUCHO TIEMPO COMPUTACION



EJEMPLO.
$$\begin{cases} \frac{dy}{dt} = 2t + y \\ y(0) = 1 \end{cases} f(t, y)$$

SEA $\Delta t = 0.1$



- $y_1 = y_0 + \Delta t f(t_0, y_0) = 1 + 0.1(2 \times 0 + 1) = 1.1$ ERROR .016
- $y_2 = y_1 + \Delta t f(t_1, y_1) = 1.1 + 0.1(2 \times 0.1 + 1.1) = 1.22$.034
- $y_3 = y_2 + \Delta t f(t_2, y_2) = 1.23 + 0.1(2 \times 0.2 + 1.22) = 1.393$.057
- $y_4 = y_3 + \Delta t f(t_3, y_3) = 1.593 + 0.1(2 \times 0.3 + 1.593) = 1.791$.083

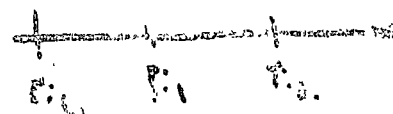
ERRORE CREECE AL

(SOL. EXACTA ...)

MÉTODO NUMÉRICO DE:
EULER MODIFICADO

RESOLVER: $\frac{dy}{dt} = F(t, y)$

SUJETO A: $y(t_0) = y_0$



PARA CADA t_i SE EFECTUAN ITERACIONES PARA OBTENER y_i MÁS EXACTA.

EN t_0 : $y' = F(t, y)$

$y_1 = y_0 + \int_{t_0}^{t_1} F(t, y) dt$

$y_1 = y_0 + \frac{\Delta t}{2} [F(t_0, y_0) + F(t_1, y_1)]$

PROBLEMA:

EULER: $y_1^{(1)} = y_0 + \Delta t F(t_0, y_0)$

$y_1^{(1)} = y_0 + \Delta t F(t_0, y_0)$

$y_1^{(2)} = y_0 + \frac{\Delta t}{2} [F(t_0, y_0) + F(t_1, y_1^{(1)})]$

$y_1^{(3)} = y_0 + \frac{\Delta t}{2} [F(t_0, y_0) + F(t_1, y_1^{(2)})]$

$y_1^{(4)} = \dots$

$y_1^{(5)} = \dots$

\vdots

CUANDO $|y_1^{(i)} - y_1^{(i-1)}| < \epsilon$
 SE TIENE y_1 EN t_1 .

EN t_2 : $y_2^{(1)} = y_1 + \Delta t F(t_1, y_1)$

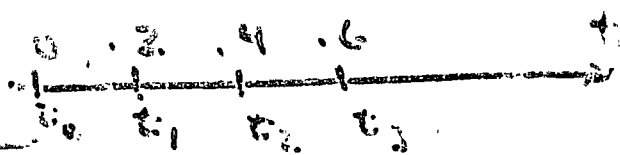
$y_2^{(2)} = y_1 + \frac{\Delta t}{2} [F(t_1, y_1) + F(t_2, y_2^{(1)})]$

$y_2^{(3)} = y_1 + \frac{\Delta t}{2} [F(t_1, y_1) + F(t_2, y_2^{(2)})]$

$y_2^{(4)}$

EJEMPLO: $\begin{cases} \frac{dy}{dt} = 2t + y \\ y(0) = 1 \end{cases}$

SEA $\Delta t = 0.2$



EN $t_1 = 0.2$

$$y_1^{(1)} = y_0 + \Delta t \cdot f(t_0, y_0) = 1 + 0.2 \cdot (2 \cdot 0 + 1) = 1.2$$

$$y_1^{(2)} = y_0 + \frac{\Delta t}{2} [f(t_0, y_0) + f(t_0, y_1^{(1)})] =$$

$$= 1 + \frac{0.2}{2} [(2 \cdot 0 + 1) + (2 \cdot 0 + 1.2)] = 1.26$$

$$y_1^{(3)} = \dots = 1.267$$

$$y_1^{(4)} = \dots = 1.267$$

$$y_1^{(5)} = \dots = 1.267 \quad \therefore y_1 = 1.267$$

EN $t_2 = 0.4$

$$y_2^{(1)} = y_1 + \Delta t \cdot f(t_1, y_1) = 1.267 + 0.2 \cdot (2 \cdot 0.2 + 1.267) = 1.66$$

$$y_2^{(2)} = y_1 + \frac{\Delta t}{2} [f(t_1, y_1) + f(t_1, y_2^{(1)})] = \dots = 1.674$$

$$y_2^{(3)} = 1.681$$

$$y_2^{(4)} = 1.682$$

$$y_2^{(5)} = 1.682 \quad \therefore y_2 = 1.682$$

t	SOL. EXACTA	EUL. ER. $\Delta t = 0.1$		EUL. ER. MODIF. $\Delta t = 0.2$	
		APROX.	ERROR	APROX.	ERROR
0.2	1.264	1.230	0.034	1.267	0.003
0.4	1.675	1.592	0.083	1.687	0.012

EUL. ER. MODIFICADO: MENOR ERROR EXACTO ≈ 12 VEC. APROX.

METODO NUMÉRICO DE RUNGE-KUTTA ORDEN 4

$$\begin{cases} \frac{dy}{dt} = f(t, y) \\ y(t_0) = y_0 \end{cases}$$

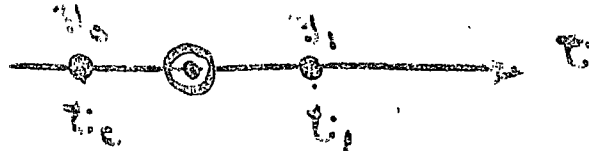
SE OBTIENE TRUNCANDO
SERIE DE TAYLOR

PARA $t_1 = t_0 + \Delta t$

$$\begin{aligned} k_1 &= \Delta t f(t_0, y_0) \\ k_2 &= \Delta t f\left(t_0 + \frac{\Delta t}{2}, y_0 + \frac{k_1}{2}\right) \\ k_3 &= \Delta t f\left(t_0 + \frac{\Delta t}{2}, y_0 + \frac{k_2}{2}\right) \\ k_4 &= \Delta t f(t_0 + \Delta t, y_0 + k_3) \end{aligned}$$

ENTONCES $y_1 = y_0 + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4)$

F: PENDIENTES (DERIVADAS) EVALUADAS EN
DIFERENTES PUNTOS DE LA CURVA



PARA $t_2 = t_1 + \Delta t$

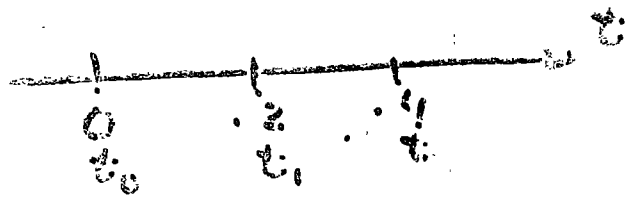
$$\begin{aligned} k_1 &= \Delta t f(t_1, y_1) \\ k_2 &= \Delta t f\left(t_1 + \frac{\Delta t}{2}, y_1 + \frac{k_1}{2}\right) \\ k_3 &= \Delta t f\left(t_1 + \frac{\Delta t}{2}, y_1 + \frac{k_2}{2}\right) \\ k_4 &= \Delta t f(t_1 + \Delta t, y_1 + k_3) \end{aligned}$$



ENTONCES: $y_2 = y_1 + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4)$

EJEMPLO:
$$\begin{cases} \frac{dy}{dt} = 2t + y \\ y(0) = 1 \end{cases}$$

SEA $\Delta t = .2$



EN $t_1 = .2$

$K_1 = \Delta t f(t_0, y_0) = .2 (2 \times 0 + 1) = .2$

$K_2 = \Delta t f(t_0 + \frac{\Delta t}{2}, y_0 + \frac{K_1}{2}) = .2 [2(0 + \frac{.2}{2}) + (1 + \frac{.2}{2})]$
 $0 + \frac{.2}{2} = .1$
 $= .26$

$K_3 = \Delta t f(t_0 + \frac{\Delta t}{2}, y_0 + \frac{K_2}{2}) = .2 [2 \times .1 + (1 + \frac{.26}{2})] = .26$

$K_4 = \Delta t f(t_0 + \Delta t, y_0 + K_3) = .2 [2(0 + .2) + (1 + .266)] = .332$

$y_1 = y_0 + \frac{1}{6} (K_1 + 2K_2 + 2K_3 + K_4) = 1.2642$

EN $t_2 = .4$, $y_2 = 1.6754$

ERRORES EN $t_1 = .2$: 0.00000 } MUY
 $t_2 = .4$: 0.00001 } EXACTO

METODO DE MILNE

$$\begin{cases} \frac{dy}{dx} = f(t, y) \\ y(t_0) = y_0 \end{cases}$$

$y(t_0)$: EN TÉRMINOS DE y_0, y_1, y_2, y_3, y_4



RUTA CRÍTICA.

- PLANEACIÓN DE PROYECTOS
- ALOCACIÓN DE RECURSOS
- ACTIVIDADES CRÍTICAS
- TIEMPOS DE EJECUCIÓN
- UTIL EN PROYECTOS COMPLEJOS
- FLUJO DE DINERO
- SECUENCIA DE ACTIVIDADES
- RUTA CRÍTICA (CPM) Y EVALUACIÓN DE PROGRAMAS Y TÉCNICAS DE REVISIÓN (PERT) MÉTODOS SEMEJANTES
- USADO EN CONSTRUCCIÓN
- DESARROLLADO EN 1956
- PLANEADOR SE CONCENTRA EN ASPECTOS IMPORTANTES

EJEMPLO.

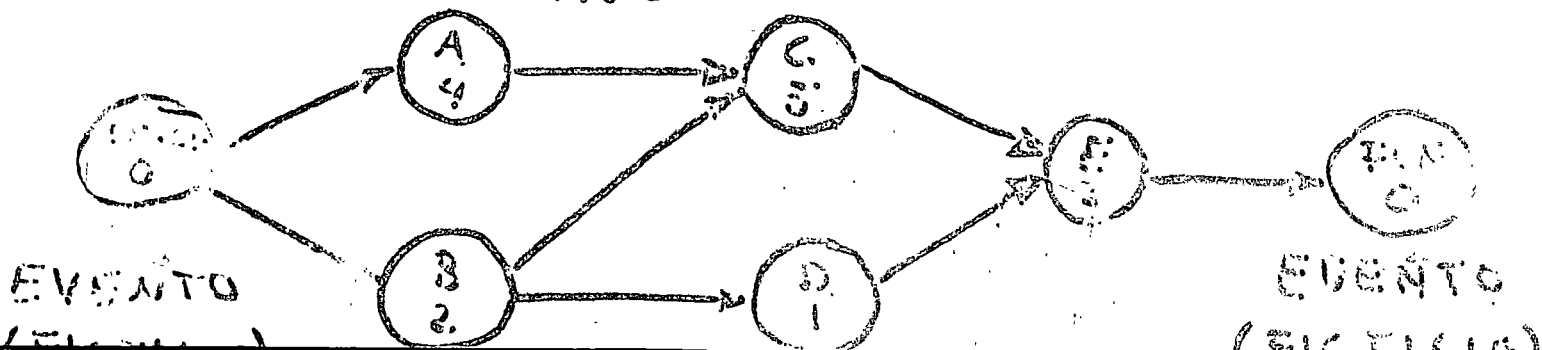
ACTIVIDADES A Y B PRECEDEN A C.

D SIGUE DE B

E EMPIEZA DESPUÉS DE C Y D

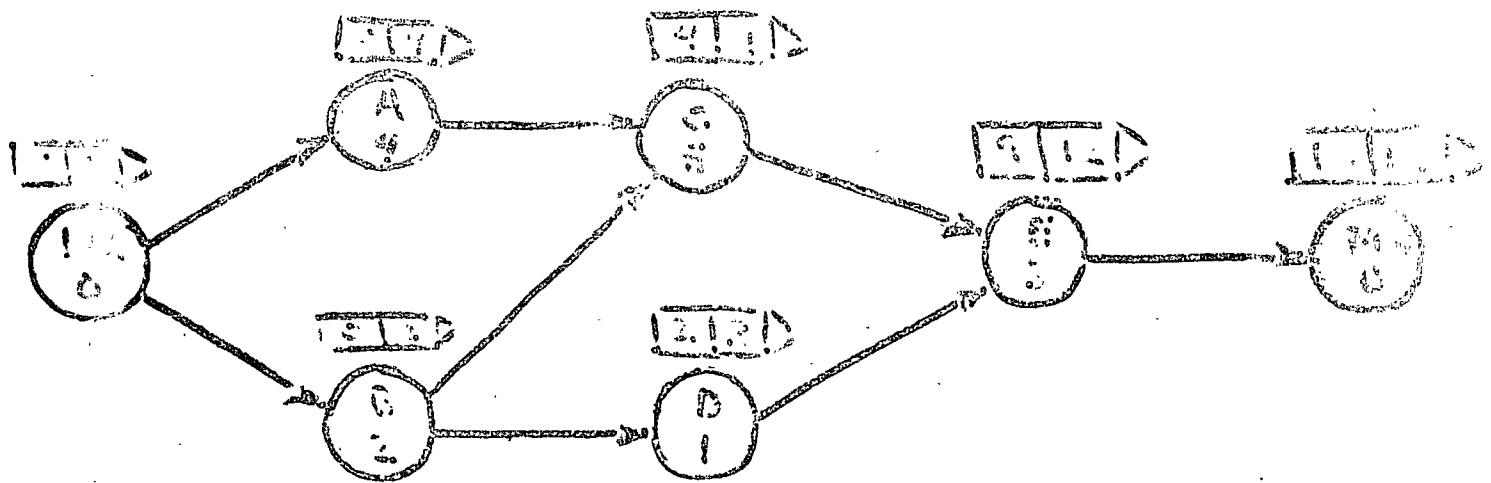
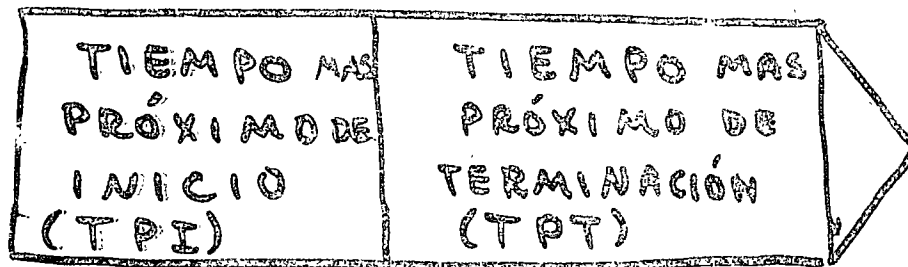
DURACIONES: $D(A)=4$, $D(B)=3$, $D(C)=5$, $D(D)=4$

GRÁFICA LINEAL



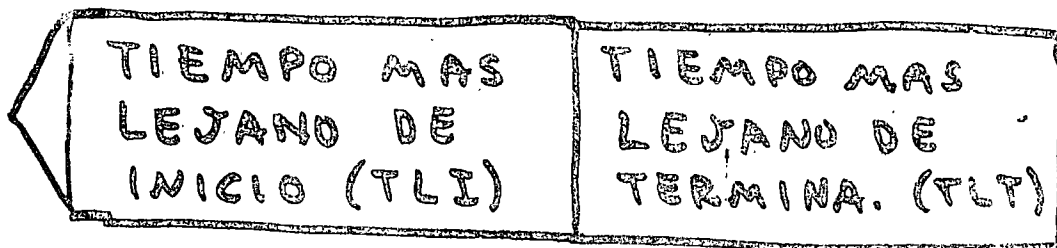
- TIEMPO DE EJECUCIÓN ?

(2)



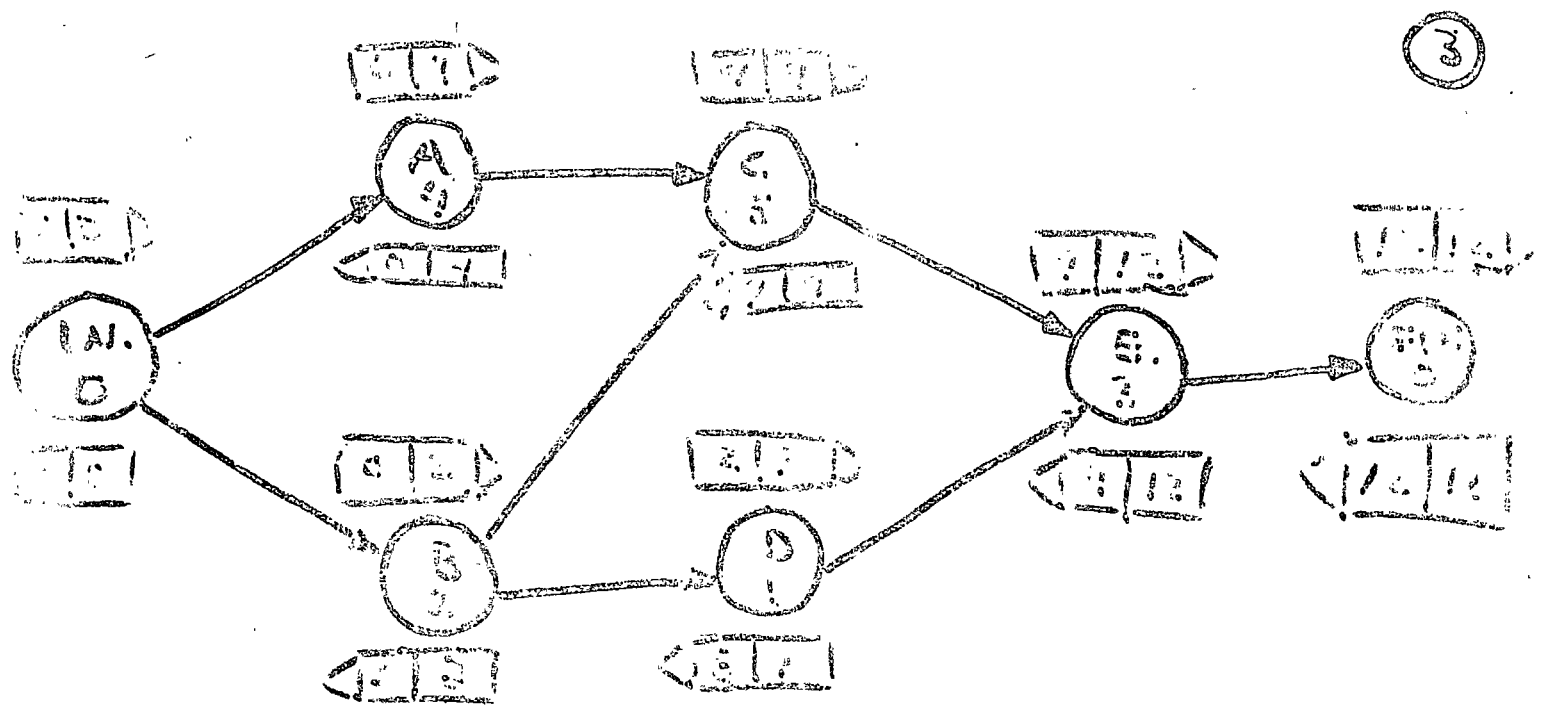
TPI = MÁXIMO DE LOS TPT ACTIVIDADES INMEDIATAMENTE PRECEDENTES

$TPT = TPI + D$



TLI = MÍNIMO DE LOS TLI ACTIVIDADES INMEDIATAMENTE PRECEDENTES

$TLI = TLT - D$



— —: RUTA CRÍTICA

$$\text{HOLGURA TOTAL (HT)} = \text{TLE} - \text{TPI} = \text{TLT} - \text{TPT}$$

= TIEMPO QUE SE PUEDE ATRASAR UNA ACTIVIDAD SIN ATRASAR PROYECTO

HT=0: ACTIVIDADES CRÍTICAS

$$\text{HOLGURA LIBRE (HL)} = (\text{MIN TPI ACTIVIDADES SUCESIVAS}) - \text{TPT}$$

= TIEMPO QUE SE PUEDE ATRASAR UNA ACTIVIDAD SIN ATRASAR OTRA ACTIVIDAD

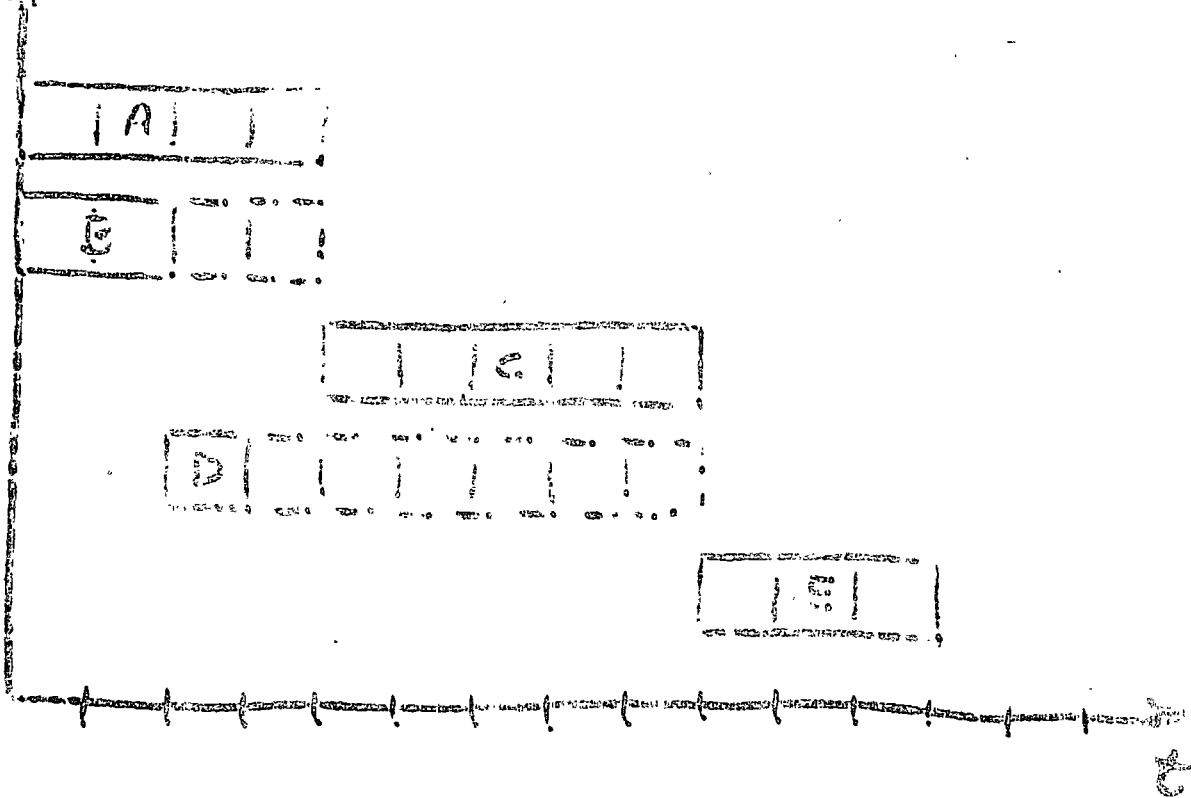
	HT	HL		COMPUTACIONES
INICIO	0	0	←	
A	0	0	←	
B	2	0		
C	0	0	←	
D	6	6	←	
E	0	0	←	
FINAL	0	0	←	

CRÍTICAS

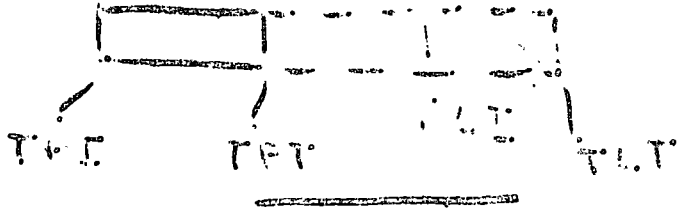
HT	TLE	TPI	TLT	TPT
0	0	0	0	0
1	0	0	0	0
1	0	0	0	0
0	1	1	0	0
0	0	1	0	0
0	0	1	1	0
0	0	0	0	0

DIAGRAMA DE BARRAS O GANTT

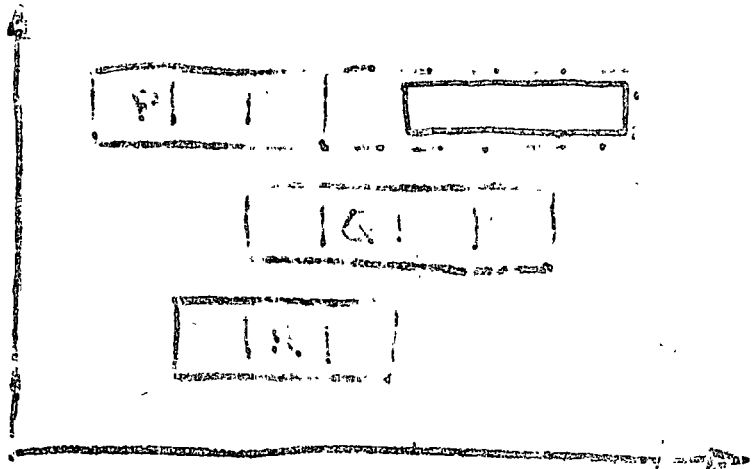
ACTIVIDADES



$||| = HT$



ALOCACION DE RECURSOS:
3 GRUPOS, 3 ACTIVIDADES



PROGRAMACIÓN LINEAL

①

- METODO DE OPTIMIZACIÓN (MAXIM. O MINIM.)
- OPTIMIZA FUNCIÓN SUJETA A RESTRICCIONES
- FUNCIÓN } LINEALES
RESTRICCIONES }
- UN BUEN PROGRAMA BASTA

EJEMPLO: TRANSPORTE

x_1 = NUM. CAMIONETAS 2 TONELADAS = ?

x_2 = NUM. CAMIONETAS 4 TONELADAS = ?

$$\text{MAX. TRANSPORTE} = 2x_1 + 4x_2$$

RESTRICCIONES:

1) 24 DÍAS DE MECÁNICO/MES

1 DIA SERVICIO CAM. 2 TONELADA.

4 DIAS " " 4 " "

$$x_1 + 4x_2 \leq 24$$

2) 9 ANDENES DE CARGA

$$x_1 + x_2 \leq 9$$

3) 21 PERSONAS PARA CARGAR DISPONIBLE

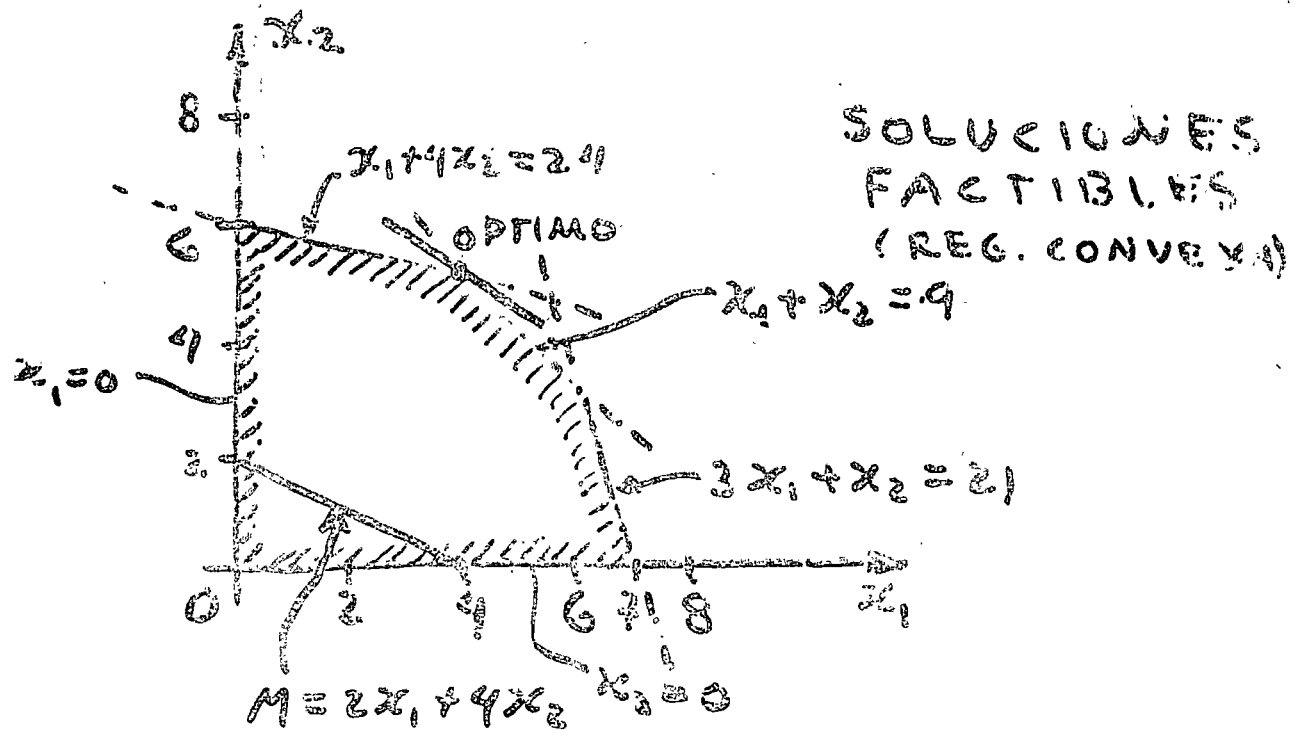
3 " " " CAM. 2 TON.

1 " " " " 4 "

$$3x_1 + x_2 \leq 21$$

NO NEGATIVIDAD: $x_1 \geq 0$

$$x_2 \geq 0$$



$\text{MAX: } M = 2x_1 + 4x_2$ ACTIVIDADES
FUNCIÓN OBJETIVO
 RESTRICCIONES: $\begin{cases} x_1 + 4x_2 \leq 24 \\ x_1 + x_2 \leq 9 \\ 3x_1 + x_2 \leq 21 \end{cases}$
 $\begin{cases} x_1 \geq 0 \\ x_2 \geq 0 \end{cases}$ } CONDS. NO NEGATIVAS

$$\begin{pmatrix} 1 & 4 \\ 1 & 1 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 24 \\ 9 \\ 21 \end{pmatrix}$$

$$\boxed{A \quad \underline{x} \leq \underline{b}}$$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\boxed{\underline{x} \geq \underline{0}}$$

$$\text{MAX: } M = (2 \quad 4) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\boxed{M = \underline{c}^T \underline{x}}$$

1 SOL : VÉRTICE
 ∞ SOL : LADO

FUNCION LINEAL: $F(x_1, \dots, x_n)$

1) HOMOGENEIDAD

$F(kx_1, \dots, kx_n) = k \cdot F(x_1, \dots, x_n)$

2) ADITIVIDAD

$F(x_1 + x'_1, \dots, x_n + x'_n) = F(x_1, \dots, x_n) + F(x'_1, \dots, x'_n)$

EQUIVALENTEMENTE:

$F(kx_1 + k'x'_1, \dots, kx_n + k'x'_n) = k \cdot F(x_1, \dots, x_n) + k' \cdot F(x'_1, \dots, x'_n)$

EJEMPLOS:

$F(x_1, x_2) = x_1 + 4x_2$

LINEAL.

$= 3$

$= x_1^2$

$= x_1 \cdot x_2$

} NO LINEALES

METODO SIMPLEX:

MÉTODO NUMÉRICO SOLUCION ÓPTIMA DE:

$M = \underline{C}^T \underline{x}$

SUJETA A RESTRICCIONES: $A \underline{x} \leq \underline{b}$

$\underline{x} \geq \underline{0}$

SIMPLEX: DESIGUALDADES \rightarrow IGUALDADES

\leq HOLGURA

\geq HOLGURA, ARTIFICIALES

$=$ ARTIFICIALES

FUNCION OBJETIVO: ARTIFICIALES

ZONA CONVEXA: SOL. EN PERIFERIA

BÚSQUEDA DE VÉRTICE EN VÉRTICE

EJEMPLO MÉTODO SIMPLEX: PRODUCCIÓN

TIPO MAQUINA	HORAS PARA PRODUCCIÓN		TOTAL HORAS DISPONIBLES
	PRODUCTO 1	PRODUCTO 2	
A	2	1	70
B	1	1	40
C	1	3	90
GANANCIA POR UNIDAD		40	60

MAXIMIZAR GANANCIA.

$x_1 = \text{NUM. ARTÍCULOS 1} = ?$
 $x_2 = \text{NUM. ARTÍCULOS 2} = ?$

MAQ. A: $2x_1 + x_2 \leq 70$
 MAQ. B: $x_1 + x_2 \leq 40$
 MAQ. C: $x_1 + 3x_2 \leq 90$

MAX: GANANCIA = $M = 40x_1 + 60x_2$

SIMPLEX: DESIGUALDADES \rightarrow IGUALDADES:

$$\begin{cases} 2x_1 + x_2 + x_3 = 70 \\ x_1 + x_2 + x_4 = 40 \\ x_1 + 3x_2 + x_5 = 90 \end{cases} \quad x_3, x_4, x_5 \geq 0$$

SISTEMA DE 3 ECUACIONES 5 INCÓGNITAS.

si: $x_1 = 0$
 $x_2 = 0$
 ENTONCES: $x_3 = 70$
 $x_4 = 40$
 $x_5 = 90$
 Y $M = 40 \times 0 + 60 \times 0 = 0$

PRIMERA SOLUCIÓN FACTIBLE

VARIABLES DE LA BASE: 70 x_3, x_4, x_5

$M = 40x_1 + 60x_2$

SI $x_1 \nearrow 1$, $M \nearrow 40$
 $x_2 \nearrow 1$, $M \nearrow 60$ \leftarrow

$x_1 = 0$, $x_3 = 70 - x_2$ $= 0$ $x_2 = 70$
 $x_4 = 40 - x_2$ $= 0$ $x_2 = 40$
 $x_5 = 90 - x_1 - 3x_2$

SEGUNDA SOLUCIÓN FACTIBLE:

$$\begin{array}{lll}
 x_1 = 0 & x_2 = 30 & M = 40 \times 0 + 60 \times 30 \\
 x_5 = 0 & x_3 = 40 & = 1800 \\
 & x_4 = 10 &
 \end{array}$$

OBTENIÉNDOLA CON SIST. ECS.

x_2, x_3, x_4 : UNA EN CADA ECUACIÓN

$$\begin{cases}
 \frac{5}{3}x_1 + x_2 - \frac{1}{3}x_5 = 40 \\
 \frac{2}{3}x_1 + x_3 - \frac{1}{3}x_5 = 10 \\
 \frac{1}{3}x_1 + x_4 + \frac{1}{3}x_5 = 30
 \end{cases}$$

$$M = 40x_1 + 60(30 - \frac{1}{3}x_5 - \frac{1}{3}x_1) = 1800 + 20x_1 - 20x_5$$

SI $x_1 \nearrow 1$, $M \nearrow 20$ ←
 $x_5 \nearrow 1$, $M \searrow 20$ X

$$\begin{array}{lll}
 x_5 = 0: & x_3 = 40 - \frac{2}{3}x_1 = 0 & x_1 = 24 \\
 & x_4 = 10 - \frac{1}{3}x_1 = 0 & x_1 = 15 \leftarrow \\
 & x_2 = 30 - \frac{1}{3}x_1 = 0 & x_1 = 90
 \end{array}$$

TERCERA SOLUCIÓN FACTIBLE:

$$\begin{array}{lll}
 x_4 = 0 & x_1 = 15 & M = 1800 + 20(15) - 0 \\
 x_5 = 0 & x_2 = 17.5 & = 2100 \\
 & x_3 = 15 &
 \end{array}$$

SIST ECS:

$$\begin{cases}
 x_2 & x_2 - 2.5x_1 + 0.5x_5 = 17.5 \\
 x_1 & + 1.5x_1 - 0.5x_5 = 15 \\
 x_3 & - 0.5x_1 + 0.5x_5 = 2.5
 \end{cases}$$

$$M = 1800 + 20(15 - 1.5x_1 + 0.5x_5) - 20x_5 = 2100 - 30x_1 - 10x_5$$

$$M = 1100 - 30x_3 - 10x_5$$

$$s! \quad x_4 \nearrow 1, \quad M \leq 30$$

$$x_5 \nearrow 1, \quad M \leq 10$$

YA SE TIENE LA SOLUCIÓN ÓPTIMA
(MAXIMA)

$x_1 = 15$: SE DEBE FABRICAR 15 ARTICULOS 1
 $x_2 = 25$: " " " " 25 " 2

GANANCIA : 2100

EN TERMINOS DE MATRICES (TABLEAU):

1ª ITERACIÓN:

	x_1	x_2	x_3	x_4	x_5	b	
	2	1	1	0	0	70	$70/1 = 70$
	1	1	0	1	0	40	$40/1 = 40$
	1	3	0	0	1	90	$90/3 = 30$ ← MÁS CHICO
-M	-40	-60	0	0	0	0	

↑ MÁS -

PIVOTE : 3 ←

↑

2ª ITERACIÓN:

	x_1	x_2	x_3	x_4	x_5	b	
	$5/3$	0	1	0	$-1/3$	40	$40/(5/3) = 24$
	$2/3$	0	0	1	$-1/3$	10	$10/(2/3) = 15$ ←
	$1/3$	1	0	0	$1/3$	30	$30/(1/3) = 90$
	-20	0	0	0	2.0	1500	

↑

3^a ITERACIÓN:

$$\begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & b \\ \left[\begin{array}{cccccc} 0 & 0 & 1 & -1.5 & 0.5 & 15 \\ 1 & 0 & 0 & 1.5 & -0.5 & 15 \\ 0 & 1 & 0 & -0.5 & 0.5 & 25 \\ 0 & 0 & 0 & 30 & 10 & 2100 \end{array} \right] \end{array}$$

TOODS + \Rightarrow SOLUCIÓN ÓPTIMA

$$\begin{cases} x_3 = 15 \\ x_1 = 15 \\ x_2 = 25 \\ M = 2100 \end{cases}$$

SE LEEN DE LA
MATRIZ.

INTERPRETACIÓN:

$x_1 = 15 =$ NUM. ARTICULOS 1

$x_2 = 25 =$ " " 2

CAJANANCIA = $M = 2100$

MÁQUINA A: $2x_1 + x_2 + x_3 = 70$

$x_3 = 30 \Rightarrow$ MÁQ. A NO SE EMPLEA
DURANTE 30 HORAS (HOLGURA)

MÁQUINA B: $x_1 + x_2 + x_3 = 40$

$x_4 = 0 \Rightarrow$ MÁQ. B PLENAAMENTE
APROVECHADA

MÁQUINA C: $x_1 + 3x_2 + x_3 = 90$

$x_5 = 0 \Rightarrow$ MÁQ. C PLENAAMENTE
APROVECHADA

SIMPLEX: DESIGUALDADES → IGUALDADES

	<u>VARIABLES</u>
≤	+ HOLGURA
≥	- HOLGURA, + ARTIFICIAL
=	+ ARTIFICIAL

FUNCION OBJETIVO :

MAXIMIZACIÓN: -M (ARTIF.)

MINIMIZACIÓN: +M (ARTIF.)

M GRANDE

EJEMPLO: MAX: $Z = 2x_1 + 7x_2$

$$2x_1 + 5x_2 \leq 150$$

$$x_1 + 2x_2 \geq 20$$

$$2x_1 - x_2 = 30$$

$$2x_1 + 5x_2 + h_1 = 150$$

$$x_1 + 2x_2 - h_2 + a_1 = 20$$

$$2x_1 - x_2 + a_2 = 30$$

MAX: $Z = 2x_1 + 7x_2 - 1000(a_1 + a_2)$

EJEMPLO: DIETA

AL MENOS 21 UNIDADES VITAMINA A
 " " " 12 " " " B

ALIMENTO	POR UNIDAD DE ALIMENTO		COSTO
	UNIDADES DE VITAMINA		
	A	B	
1 (MILANINA)	1	0	20
2 (MAREANA)	0	1	20
3 (LECHUGA)	1	2	31
4 (CHICHARRO)	1	1	11
5 (ZUVAMORIN)	2	1	12

MINIMIZAR COSTOS.

x_i = CANTIDAD (UNIDADES) DE ALIMENTO i .

MIN: COSTO = $20x_1 + 20x_2 + 31x_3 + 11x_4 + 12x_5$

REST: $x_1 + x_2 + x_3 + 2x_4 + 2x_5 \geq 21$ COEF.S. COSTOS

$x_2 + 2x_3 + x_4 + x_5 \geq 12$ RESTRICCIONES

COEF.S. ESTRUCTURALES

$x_i \geq 0$

DUAL: CIA. FARMACEUTICA "LA CAMPANA":

λ_1 = PRECIO PILDORA VITAMINA A = ?

λ_2 = " " " " " B = ?

MAXIMIZAR GANANCIA (PRECIOS COMPETITIVOS)

MAX: GANANCIA = $21\lambda_1 + 12\lambda_2$

! P.M. 0000 → λ₁ ∈ ℝ — PRECIO NA REAJA
 VIT. A λ₂ ∈ ℝ
 λ₁ + λ₂ ∈ ℝ
 λ₁ + λ₂ ∈ ℝ
 2λ₁ + λ₂ ∈ ℝ

PROBLEMA PRIMO:

MIN: w = (20 20 31 11 12) $\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}$

RESTRIÇ: $\begin{pmatrix} 1 & 0 & 1 & 1 & 2 \\ 0 & 1 & 2 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \leq \begin{pmatrix} 21 \\ 12 \end{pmatrix}$

$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

PROBLEMA DUAL:

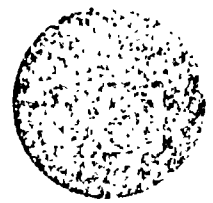
MAX: z = (21 12) $\begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix}$

RESTRIÇ: $\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 2 \\ 1 & 1 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} \leq \begin{pmatrix} 20 \\ 20 \\ 31 \\ 11 \\ 12 \end{pmatrix}$

$\begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$



centro de educación continua
división de estudios superiores
facultad de Ingeniería, unam



METODOS NUMERICOS Y APLICACIONES CON LA COMPUTADORA
DIGITAL

O P T I M I Z A C I O N

PROF. DR. VICTOR GEREZ GREISER.

ABRIL, 1978.

6.5. PROGRAMACION LINEAL

6.5.1 Ejemplos

Existen muchos problemas de optimización cuyo modelo matemático es de tal naturaleza que se pueden resolver con la técnica de optimización conocida con el nombre de programación lineal. Se han desarrollado algoritmos y basados en ellos, programas de computadora digital para la solución de estos problemas.

*La estructura de los problemas que pueden resolverse con esta técnica es siempre la misma, de manera que contando con un buen programa para la solución de éstos, pueden resolverse sin necesidad de tener que escribir programas especiales para la solución de problemas particulares. Los problemas de optimización que se pueden resolver con *la técnica de programación dinámica por otra parte no tiene esta característica y con frecuencia es necesario desarrollar programas particulares para obtener la solución de un problema específico.

En esta sección se empezará a ilustrar con ejemplos la formulación de modelos matemáticos que permiten aplicar la programación lineal. A continuación, la ilustración geométrica de la solución del problema de programación lineal, sirve para introducir el método simplex de solución de problemas.

El primer ejemplo ilustra un problema de transporte. Supóngase que una embotelladora tiene dos plantas, una en Tlaxcala y otra en Tehuacán, con capacidad de 7 000 y 13 000 cajas de refrescos al día, además tiene dos centros de consumo que son Puebla y Orizaba, que pueden consumir hasta 12 000 y 8 000 cajas diarias respectivamente. El costo de envío de una caja de refrescos de los diferentes lugares de producción a los diferentes destinos está dado en la tabla 6.5.1.

*Todos los problemas de programación lineal tienen el mismo modelo matemático.

*No existen modelos generales para problemas de programación dinámica.

Ejemplo 6.5.1

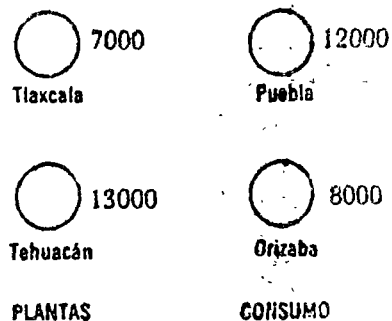


Tabla 6.5.1 Costos de transporte en el ejemplo 6.4.1.

de a	Tlaxcala 1	Tehuacán 2
Puebla 1	0.8	1.00
Orizaba 2	1.30	0.90

El administrador de la empresa debe determinar cuántas cajas deben enviarse de cada embotelladora a cada centro de consumo, de manera que se satisfagan las siguientes condiciones:

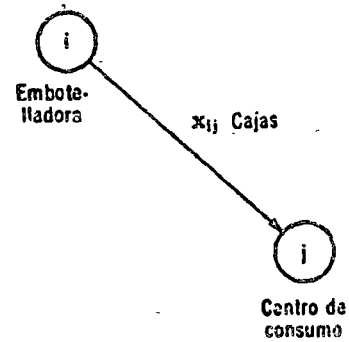
- 1) Cada embotelladora no puede enviar más cajas que el máximo que puede producir.
- 2) Cada centro de consumo puede obtener tantas cajas como puede consumir.
- 3) Deben minimizarse los gastos de transporte.

Para plantear este problema en el marco de las ecuaciones (6.1.1) y (6.1.2).

$$M = M(x_1, x_2, \dots, x_n) \quad (6.1.1)$$

$$\begin{aligned} C_1 &= C_1(x_1, x_2, \dots, x_n) \geq 0 \text{ para } i = 1, 2, \dots, p \\ C_2 &= C_2(x_1, x_2, \dots, x_n) \leq 0 \text{ para } i = p + 1, \dots, r \\ C_3 &= C_3(x_1, x_2, \dots, x_n) = 0 \text{ para } i = r + 1, \dots, n \end{aligned} \quad (6.1.2)$$

es necesario definir la siguiente variable: x_{ij} es el número de cajas enviadas de la embotelladora situada en la localidad i 'sima ($i = 1$ corresponde a Tlaxcala e $i = 2$ a Tehuacán) al centro consumidor j 'simo (1 es el índice de Puebla y 2 el de Orizaba). Con la introducción de esta variable el problema puede plantearse de la siguiente forma:



Las cajas enviadas de la localidad 1 (Tlaxcala) al centro de consumo 1 (Puebla), que se ha acordado representar con x_{11} más las cajas enviadas de la localidad 1 al centro de consumo 2 (Orizaba), x_{12} , no deben exceder la capacidad de la embotelladora de la localidad 1 que es de 7 000 cajas, es decir,

$$x_{11} + x_{12} \leq 7000 \quad (6.5.1)$$

La figura 6.5.1 ilustra el planteamiento de esta ecuación:

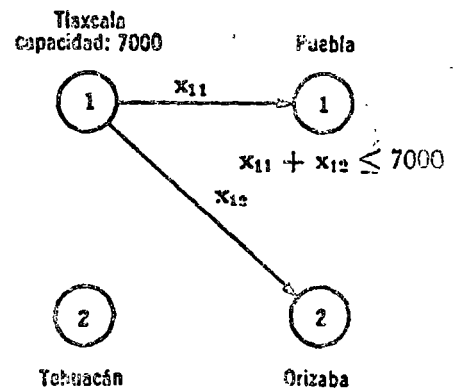


Fig. 6.5.1 Cajas enviadas desde la embotelladora en Tlaxcala.

310 Optimización

En forma similar puede establecerse la siguiente ecuación que limite la producción total de la embotelladora de la 2da. localidad a 13 000 cajas, a saber:

La figura 6.5.2 ilustra el planteamiento de otras ecuaciones.

$$x_{21} + x_{22} \leq 13\,000 \quad (6.5.2)$$

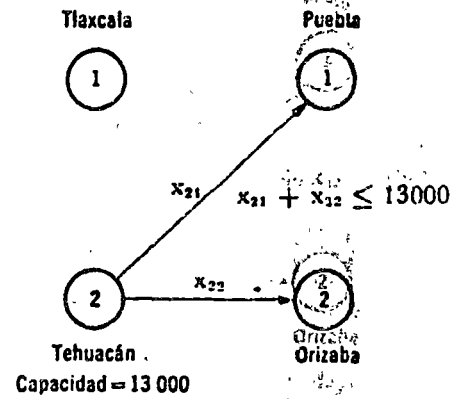


Fig. 6.5.2 Cajas enviadas desde la embotelladora en Tehuacán.

Por otra parte, se ha señalado que cada centro de consumo puede obtener tantas cajas como desea.

Al centro consumidor 1, Puebla, le llegan x_{11} cajas de Tlaxcala y x_{21} cajas de Tehuacán tal como ilustra la fig. 6.5.3. Por lo tanto, como el consumo de Puebla es de 12 000 cajas:

$$x_{11} + x_{21} \geq 12\,000 \quad (6.5.3)$$

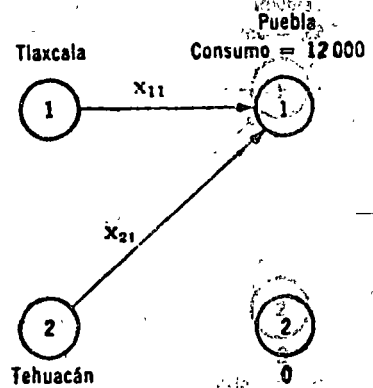


Fig. 6.5.3 Cajas recibidas en Puebla.

Finalmente como última restricción se tiene que las cajas que recibe Orizaba, centro consumidor 2, deben ser iguales o mayor a 8 000 cajas. Se tiene por lo tanto;

$$x_{12} + x_{22} \geq 8\,000 \quad (6.5.4)$$

La figura 6.5.4 ilustra el significado de esta ecuación.

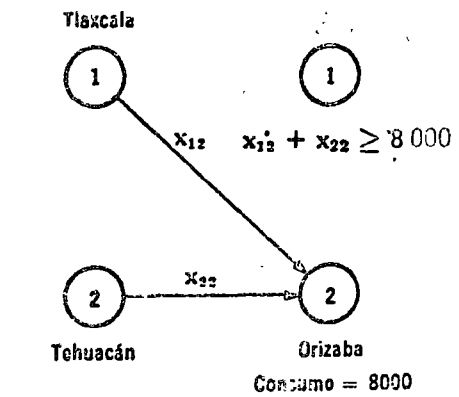


Fig. 6.5.4 Cajas recibidas por Orizaba.

Para terminar con el establecimiento del modelo matemático de este problema es necesario establecer la función objetivo.

El objetivo de análisis es minimizar los costos de transporte que están dados por:

$$M = 0.8 x_{12} + 1 x_{21} + 1.3 x_{21} + 0.9 x_{22} \quad (6.5.5)$$

Debe además imponerse la siguiente condición:

$$x_{ij} \geq 0 \quad \forall i, \forall j \quad (6.5.6)$$

ya que no tendrán significado valores negativos de envíos de cajas.

En resumen puede decirse que el problema consiste en minimizar la función objetivo.

$$M = 0.8 x_{12} + 1. x_{21} + 1.3 x_{12} + 0.9 x_{22} \quad (6.5.5)$$

Sujeto a las restricciones

$$x_{11} + x_{12} \leq 7,000 \quad (6.5.1)$$

$$x_{21} + x_{22} \leq 13,000 \quad (6.5.2)$$

$$x_{11} + x_{21} \leq 12,000 \quad (6.5.3)$$

$$x_{12} + x_{22} \geq 8,000 \quad (6.5.4)$$

$$x_{ij} \geq 0, \quad \forall i \text{ y } \forall j. \quad (6.5.6)$$

Todos los modelos matemáticos de problemas de programación lineal tienen precisamente esta forma.

Antes de continuar conviene recordar algunas definiciones introducidas en la sección 6.1.2.

*Un conjunto de valores de las variables que satisface todas las restricciones del problema se llama una *solución factible* del problema de programación lineal. Empleando la definición anterior, puede decirse que la solución del problema consiste en encontrar una solución factible que sea óptima. En este caso del problema del transporte una solución factible que minimice la función objetivo (6.5.5).

*La solución factible satisface todas las restricciones.

312 Optimización

*Este problema tiene cuatro variables que hay que determinar, x_{11} , x_{12} , x_{21} y x_{22} . Con objeto de visualizar geoméricamente la solución de los problemas de programación lineal e introducir otro tipo de problemas de optimización de este tipo, se incluye un segundo ejemplo:

*Supóngase que una compañía de transporte tiene x_1 camionetas de 2 toneladas y x_2 camionetas de 4 toneladas y desea maximizar su capacidad de transporte. La función objetivo es y el problema consiste en maximizar dicha expresión.

*Además la compañía tiene las siguientes restricciones:

*La primera es la siguiente: Las camionetas chicas requieren 1 día de mantenimiento al mes, y las grandes 4 días y la compañía sólo tiene disponibles 24 días de mecánico al mes. Matemáticamente esta restricción se expresa de la siguiente forma:

*La segunda restricción en este problema se refiere a la disponibilidad de andenes de carga. Ambos tipos de vehículo, requieren de igual número de andenes de carga, y que la compañía sólo cuenta con 9 andenes. Empleando las variables x_1 y x_2 esta restricción establece:

*La última restricción se refiere al personal que se requiere para cargarlas. Este personal está restringido a 21 personas. Las camionetas chicas requieren tres personas para cargarlas y las grandes solamente una persona. Se tiene por lo tanto

*Desde luego que las variables x_1 y x_2 , número de camionetas de 2 toneladas y de 4 toneladas con que cuenta la compañía respectivamente, no pueden ser negativas, por lo tanto las últimas restricciones en este problema son:

Desde luego existen otros muchos problemas donde puede aplicarse la programación lineal. Entre ellos pueden citarse problemas de mezclado y planeación de la producción como el ejemplo 6.5.4 de la sección 6.5.5.

Después de estos ejemplos se procederá a planear en forma normal el problema de programación lineal y se estudiarán las condiciones que debe satisfacer tanto la función objetivo como las restricciones.

*Variables del problema x_{11} , x_{12} , x_{21} y x_{22}

Ejemplo 6.5.2

* x_1 camionetas de 2 ton. x_2 camionetas de 4 ton.

$$m = 2x_1 + 4x_2 \quad (6.5.7)$$

*Restricciones.

*Mantenimiento:
24 días mecánico/mes.

$$x_1 + 4x_2 \leq 24 \quad (6.5.8)$$

*2da. Andenes de carga:
9 andenes.

$$x_1 + x_2 \leq 9 \quad (6.5.9)$$

*3ra. Cargado:
21 personas.

$$3x_1 + x_2 \leq 21 \quad (6.5.10)$$

*Ultima:
no negatividad.

$$x_1 \geq 0; x_2 \geq 0 \quad (6.5.11)$$

6.5.2. Planteamiento formal

*Si se analiza la formulación de los problemas de los dos ejemplos introducidos en la sección anterior, pueden detectarse ciertas variables que se llaman en forma genérica *actividades*.

*En el ejemplo 6.5.1 las actividades consisten en enviar cajas de refrescos de la embotelladora al centro consumidor y se han representado con los símbolos:

*En el ejemplo 6.5.2 estas actividades consisten en operar camiones de carga y se han empleado los símbolos x_1 y x_2 para representarlas.

*Cada actividad queda caracterizada por una variable que se designa como *nivel de actividad*.

*Además se observa que los problemas de los ejemplos anteriores satisfacen las siguientes condiciones:

*Tanto las restricciones como la función objetivo son funciones lineales de los niveles de actividad. Al ser lineales estas funciones son *homogéneas y aditivas*.

Una función es lineal si dados dos conjuntos:

*y dos constantes cualquiera K y K' se tiene:

$$f(Kx_1, + K'x'_1, \dots, Kx_n + K'x'_n) = Kf(x_1, x_2, \dots, x_n) + K'f(x'_1, x'_2, \dots, x'_n) \quad 6.5.12$$

*La condición de linealidad (6.5.12) es equivalente a dos condiciones. En primer lugar una función lineal tiene un factor constante de escala, es decir,

*y en segundo lugar es aditiva:

$$f(x_1 + x'_1, x_2 + x'_2, \dots, x_n + x'_n) = f(x_1, x_2, \dots, x_n) + f(x'_1, x'_2, \dots, x'_n) \quad 6.5.14$$

Un ejemplo servirá para ilustrar este importante concepto y señalar que funciones del tipo

*no son lineales. Es decir, si en las funciones hay cargos fijos (el término a) no es posible aplicar directamente el concepto de programación lineal.

*Actividades.

*Envío de cajas de refresco.

$$x_{ij}, i, j = 1, 2$$

*Operación de camiones de carga

$$x_1, x_2$$

*Nivel de actividad.

1. No negatividad de los niveles, es decir

$$x_i \geq 0, \forall i$$

*Funciones objetivo y restricciones son lineales \rightarrow homogéneas y aditivas.

$$f(x_1, x_2, \dots, x_n)$$

*Conjuntos de variables

$$x_i, i = 1, 2, \dots, n \text{ y } x'_i, i = 1, 2, \dots$$

*Constantes K y K'

*Condición de linealidad \rightarrow factor constante de escala

$$f(Kx_1, Kx_2, \dots, Kx_n) = Kf(x_1, x_2, \dots, x_n) \quad (6.5.13)$$

*Condición de linealidad \rightarrow aditividad.

$$f(x) = a + bx \quad (6.5.15)$$

*Función no lineal.

Ejemplo 6.5.3.

314 Optimización

Determine si las siguientes funciones son lineales y justifique la respuesta.

a) $y = 3x_1 + 2x_2$

b) $y = 3x + 5$

Solución:

a) Como $a3x_1 + b3x_1 + a2x_2 + b2x_2$
 $= a(3x_1 + 2x_2) + b(3x_1 + 2x_2)$

b) Como $a3x + 5 + b3x + 5 \neq a(3x + 5) + b(3x + 5)$

se cumple la condición (6.5.12) y la función es lineal.
 la función no es lineal.

El problema de programación lineal por lo tanto puede plantearse de la siguiente forma.

*Hay que determinar el valor de los niveles de actividad x_1, x_2, \dots, x_n , que maximicen a la función objetivo:

sujeito a las siguientes restricciones:

*Encontrar x que maximice:

$$m = c_1 x_1 + c_2 x_2 + \dots + c_n x_n \quad (6.5.6a)$$

y satisfaga:

$$a_{i1} x_1 + a_{i2} x_2 + \dots + a_{in} x_n = b_i \quad i = 1, 2, \dots, p$$

$$a_{i1} x_1 + a_{i2} x_2 + \dots + a_{in} x_n \leq b_i \quad i = p + 1, \dots, r$$

$$a_{i1} x_1 + a_{i2} x_2 + \dots + a_{in} x_n \geq b_i \quad i = r + 1, \dots, m$$

$$x_j \geq 0 \quad j = 1, 2, \dots, n$$

(6.5.16b)

*Los coeficientes C_i de la función objetivo se conocen con el nombre de *coeficientes de costo*, y los coeficientes a_{ij} de las ecuaciones de restricción se llaman *coeficientes estructurales*.

* c_i = coeficientes de costo

a_{ij} = coeficientes estructurales.

Como se ilustra en el ejemplo 6.5.3 un problema de maximización puede siempre convertirse en uno de minimización. Como muestra el sistema de ecuaciones (6.5.16) las restricciones pueden ser del tipo de desigualdad o igualdad. *Para la solución del problema de programación lineal conviene convertir todas las desigualdades en igualdades introduciendo *variables de holgura*, que de preferencia deben de ser positivas. La siguiente desigualdad:

*Variables de holgura ≥ 0 para convertir desigualdades en igualdades.

Desigualdad

$$a_{q1} x_1 + a_{q2} x_2 + \dots + a_{qn} x_n \leq b_q$$

+
Variable de holgura

$$x_{n+q} \geq 0$$

↓
Igualdad

$$a_{q1} x_1 + a_{q2} x_2 + \dots + a_{qn} x_n + x_{n+q} = b_q$$

puede convertirse en una igualdad introduciendo una variable positiva x_{n+q} llamada de holgura. En efecto:

*Desigualdad

$$a_{q1} x_1 + a_{q2} x_2 + \dots + a_{qn} x_n \geq b_q$$

+
Variable de holgura

$$x_{n+q} \geq 0$$

↓
Igualdad

*Si por otra parte se tiene en la ecuación de restricción la desigualdad en sentido contrario.

la introducción de la variable de holgura positiva x_{n+q} , convierte la desigualdad en una igualdad, ya que:

$$a_{q1}x_1 + a_{q2}x_2 + \dots + a_{qn}x_n - x_{n+q} = D_1$$

Además, los métodos de solución del problema de programación lineal exigen que los niveles de actividad sean positivos, es decir, $x_i \geq 0, \forall i$. * Si un nivel de actividad no está sujeto a esta restricción se le puede sustituir por la diferencia de dos niveles de actividad positivos. Supongamos que el nivel x_1 no está restringido. Si se introducen las variables

* Si nivel de actividad

$$x_1 \leq \text{ó} \geq 0$$

$$x_1 = x_1^+ - x_1^- \quad (6.5.17)$$

x_1^+ y x_1^- relacionadas con la variable x_1 mediante la siguiente diferencia.

$$x_1^+ \geq 0, x_1^- \geq 0$$

la variable o nivel de actividad original puede ser mayor, igual o menor que cero, sin que las variables x_1^+ y x_1^- tomen valores negativos. El siguiente ejemplo ilustra tanto la introducción de variable de holgura como el empleo de la relación (6.5.17) y la transformación de un problema de minimización en uno de maximización.

Ejemplo 6.5.3

Convierta el siguiente problema de minimización en un problema de maximización, transforme todas las ecuaciones de restricción en igualdades mediante la introducción de variables de holgura y transforme todas las variables en no negativas:

$$\begin{aligned} \min : m &= 3x_1 + 5x_2 \\ 3x_1 + 2x_2 &\geq 6 \\ x_1 - 6x_2 &\leq 4 \\ x_1 &\geq 0; x_2 \text{ sin restricción} \end{aligned}$$

se sabe que:

Definiendo una nueva función objetivo.

Solución:

$$\begin{aligned} \text{Min. } m &= 3x_1 + 5x_2 \text{ es equivalente a:} \\ \text{Max. } -m &= -3x_1 - 5x_2. \end{aligned}$$

la función objetivo se convierte en:

* Nueva función objetivo n:

$$\begin{aligned} n &= -m \\ \downarrow \\ \max : n &= -3x_1 - 5x_2 \end{aligned}$$

Para convertir las dos desigualdades de restricción en igualdad es necesario introducir dos nuevas variables x_3 y x_4 para realizar los siguientes cambios en las restricciones.

$$x_1 - 6x_2 \leq 4 \rightarrow x_1 - 6x_2 + x_4 = 4$$

$$3x_1 + 2x_2 \geq 6 \rightarrow 3x_1 + 2x_2 - x_3 = 6$$

* Finalmente la variable x_2 , no restringida debe sustituirse por la diferencia de dos variables no negativas

* x_2 variable sin restricción

$$x_2 = x_2^+ - x_2^-$$

Realizando esta sustitución, las ecuaciones o condiciones de restricción tienen la siguiente forma:

Ep:

$$3x_1 + 2x_2^+ - 2x_2^- - x_3 = 6$$

$$x_1 - 6x_2^+ + 6x_2^- + x_4 = 4$$

$$x_1, x_2^+, x_2^-, x_3, x_4 \geq 0$$

316 Optimización

y la función objetivo es:

También es posible resolver un problema de minimización recurriendo a su formulación dual que se estudia en la sección 6.5.5.

* La estructura del problema de programación lineal se presta para el empleo de la notación matricial. Si se definen *la matriz de coeficientes estructurales

y * los vectores de actividades:

de *costos

y *de restricciones

El problema de programación lineal queda planteado de la siguiente forma:.

Sujeto a las restricciones

En la siguiente sección se ilustra gráficamente la forma de obtener la solución del problema de programación lineal.

6.5.3 Solución gráfica

En esta sección ilustraremos gráficamente la solución del problema de programación lineal. Como es difícil representar gráficamente funciones de más de dos variables, se empleará el ejemplo 6.5.2 para realizar esta representación.

El modelo matemático de este problema es el siguiente:

$$\max : m = -3x_1 - 5x_2$$

*Formulación matricial

*Coeficientes estructurales

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & \dots & & \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad (6.5.17)$$

*Actividades

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (6.5.18)$$

*Costos

$$c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \quad (6.5.19)$$

*Restricciones

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (6.5.20)$$

$$\max : m = c^T x \quad (6.5.21)$$

$$Ax \leq b \quad (6.5.22)$$

$$x \geq 0 \quad (6.5.23)$$

$$\max : m = 2x_1 + 4x_2 \quad (6.5.7)$$

Sujeto a las restricciones

$$\begin{aligned} x_1 + 4x_2 &\leq 24 && (6.5.8) \\ x_1 + x_2 &\leq 9 && (6.5.9) \\ 3x_1 + x_2 &\leq 21 && (6.5.10) \\ x_1, x_2 &\geq 0 \end{aligned}$$

Las restricciones de este problema establecen una zona del plano (x_1, x_2) donde deben encontrarse las soluciones factibles, tal como se señaló en la sección 6.1.2. Observe que la ecuación $x_1 + 4x_2 = 24$, corresponde a una recta, que divide al plano en dos regiones. En la inferior se cumple $x_1 + 4x_2 \leq 24$, por lo tanto, la solución factible debe estar "abajo" de dicha recta. La figura 6.5.5 ilustra la zona definida por esta restricción.

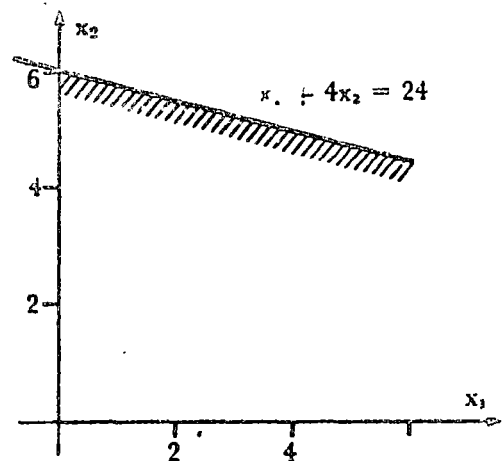


Fig. 6.5.5 Zona con restricción $x_1 + 4x_2 \leq 24$.

Un razonamiento similar lleva a concluir que la solución factible también debe estar a la "izquierda" de las rectas $x_1 + x_2 = 9$ y $3x_1 + x_2 = 21$ (fig. 6.5.6).

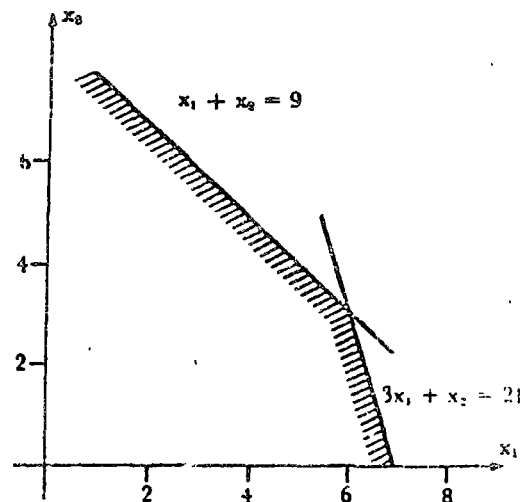


Fig. 6.5.6 Zona con restricciones $x_1 + x_2 \leq 9$ y $3x_1 + x_2 \leq 21$.

318 Optimización

Además, la condición $x_1 \geq 0$ y $x_2 \geq 0$ impone que debe estar en el primer cuadrante. La región del plano donde se cumplen todas las restricciones es por lo tanto polígono convexo OABCDO que aparece en la figura 6.5.7.

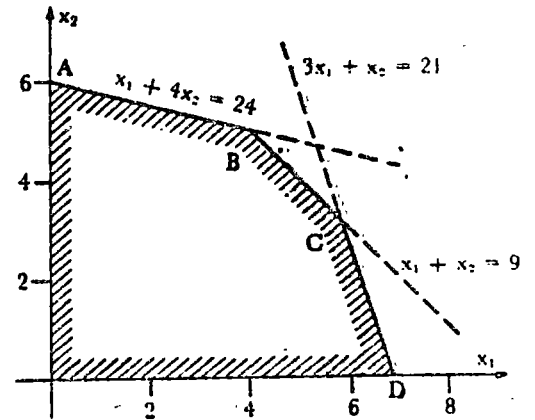


Fig. 6.5.7 Zona de soluciones factibles del ejemplo 6.5.2.

El siguiente paso en la solución consiste en encontrar dentro de los puntos de dicho polígono, que son soluciones factibles todos ellos, aquel punto para el cual la función objetivo 6.5.7 $2x_1 + 4x_2$ es máxima. Nótese primero que cualquier recta dependiente $-1/2$ cumple con la condición $2x_1 + 4x_2$. Además, entre mayor sea la distancia al origen de una recta dependiente $-1/2$, tanto mayor es $2x_1 + 4x_2$ tal como se ilustra en la figura 6.5.8.

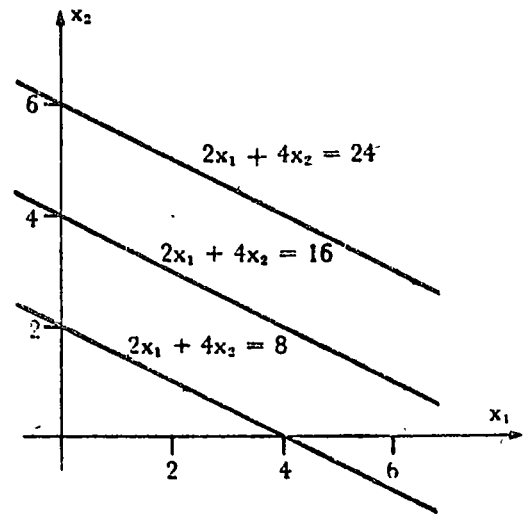


Fig. 6.5.8 Funciones objetivo del ejemplo 6.5.2.

Para obtener el valor máximo de la función objetivo $2x_1 + 4x_2$ es necesario desplazar una recta dependiente $-1/2$ de manera que su distancia al origen sea máxima, pero tenga por lo menos un punto dentro de la región OABCDO. En la figura 6.5.9 se ilustra este procedimiento de búsqueda del máximo. En el punto B de coordenadas (4, 5) el valor de la función objetivo $2x_1 + 4x_2$ es de 28 y se cumplen todas las restricciones. Por lo tanto $x_1 = 4$, $x_2 = 5$ es la solución del problema de programación lineal. Haciendo referencia a la fig. 6.5.9 obsérvese además que para dicho punto, tiene las características resumidas en el cuadro de la tabla 6.5.1.

Problema	
Función objetivo.	
$M = 2x_1 + 4x_2$ (max.)	
Restricciones.	
$x_1 + 4x_2$	≤ 24 (a)
$x_1 + x_2$	≤ 9 (b)
$3x_1 + x_2$	≤ 21 (c)
x_1	≥ 0 (d)
x_2	≥ 0 (e)

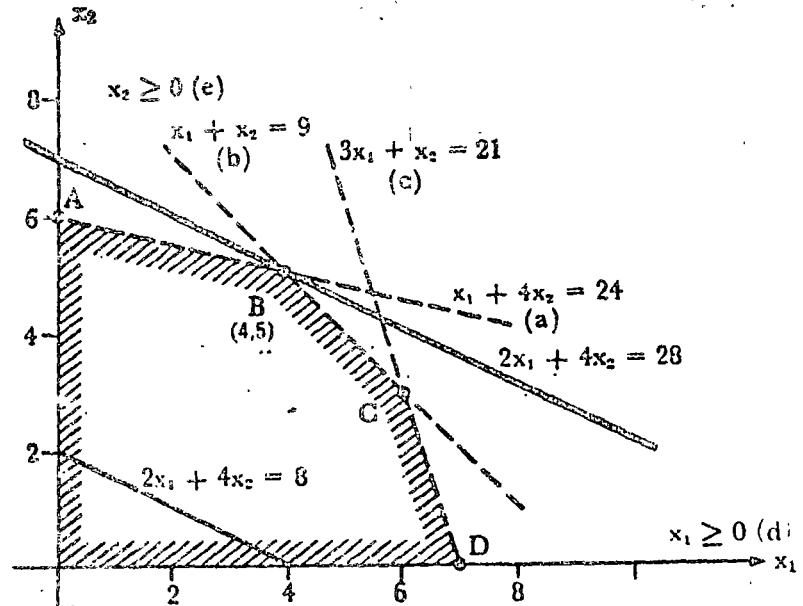


Fig. 6.5.9 Ilustración de la solución gráfica del problema de programación lineal.

Tabla 6.5.1 Propiedades de punto óptimo B del ejemplo 6.5.2.

Restricción	Holgura
$x_1 + 4x_2 = 24$	0
$x_1 + x_2 = 9$	0
$3x_1 + x_2 = 17 \leq 21$	4

Es decir, el recurso mecánico "del que se cuenta con 24 días más, el de "andenes de carga" con el que se cuenta con 9, se emplea plenamente si se usan 4 camionetas de dos toneladas y 5 de 4 toneladas. Mientras que de tercer recurso, del que se cuenta con 21 unidades, sólo se usan 17. Sin embargo, ninguna otra combinación de x_1 y x_2 permite obtener mayor volumen de carga sin violar las restricciones (6.5.8), (6.5.10). Antes de continuar, nótese que la región definida por las restricciones (6.5.8-6.5.10) es convexa, como muestra la figura 6.5.10, ya que cualquier recta que une dos puntos cualquiera de la periferia de la zona se encuentra en la frontera o dentro de la región.

En la sección 6.5.4 se empleará la representación gráfica de la solución de programación lineal para visualizar fácilmente diversos casos especiales de problemas de este tipo.

*El método gráfico de solución del problema de programación lineal está restringido a modelos con dos variables. Prácticamente todos los problemas de interés para el analista tienen más de dos

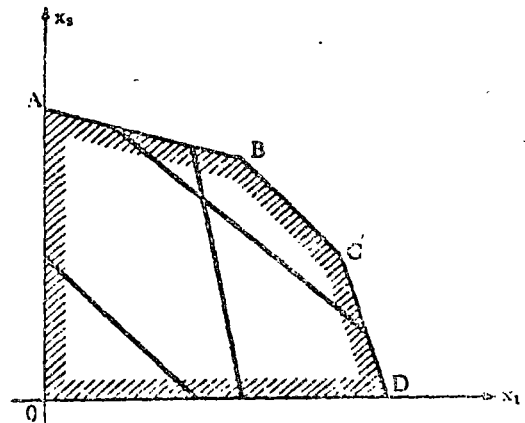


Fig. 6.5.10 Zona convexa de soluciones factibles.

*Método gráfico para problemas con dos variables.

variables, por lo cual el método gráfico no se puede emplear en estos casos. *Es necesario contar con métodos algebraicos que se puedan programar en una computadora digital, con objeto de resolver problemas con un gran número de variables, como son la mayoría de los que se encuentran en la práctica. El método simplex que se introduce en la siguiente sección tiene esta propiedad. Sin embargo, es importante familiarizarse con la solución gráfica estudiada en esta sección, ya que ayuda a entender la naturaleza de la solución del problema.

Al ir desarrollando el método simplex de solución analítica, continuamente se hará referencia a la solución gráfica. Los autores consideran que de esta forma el lector lo comprenderá con mayor facilidad.

6.5.4 Solución analítica

El método analítico más importante para la solución de este tipo de problemas es el *método simplex, que introduciremos resolviendo el ejemplo 6.5.2.

*La función objetivo de este ejemplo es:
 $\max : m = 2x_1 + 4x_2$

*Sujeto a las restricciones

*El primer paso en este método consiste en introducir variables de holgura x_3, x_4, x_5 para convertir las desigualdades de las ecuaciones de restricción en igualdades, tal como se señaló en la sección 6.5.2.

*Debido al signo de las desigualdades, las variables de holgura deben ser positivas, es decir:

*El problema consiste en encontrar los valores de las variables x_j que maximicen a la función objetivo (6.5.7).

*Como el sistema (6.5.24) tiene 3 ecuaciones con 5 incógnitas pueden expresarse 3 de ellas cualesquiera en función de las dos restantes.

*Como la variable x_3 sólo aparece en la 1er. ecuación, la x_4 en la 2da. y x_5 en la 3er. ecuación, lo más conveniente es tomar $x_1 = 0$ y $x_2 = 0$, obteniéndose de inmediato del sistema (6.5.24) que $x_3 = 24, x_4 = 9$ y $x_5 = 21$. Esta solución se conoce con el

*Métodos algebraicos para resolver sistemas con muchas variables.

*Método simplex.

*Función objetivo.

$$\max : m = 2x_1 + 4x_2 \quad (6.5.7)$$

*Restricciones.

$$x_1 + 4x_2 \leq 24 \quad (6.5.8)$$

$$x_1 + x_2 \leq 9 \quad (6.5.9)$$

$$3x_1 + x_2 \leq 21 \quad (6.5.10)$$

$$x_1, x_2 \geq 0$$

$$x_1 + 4x_2 + x_3 = 24$$

$$x_1 + x_2 + x_4 = 9 \quad (6.5.24)$$

$$3x_1 + x_2 + x_5 = 21$$

*Variables de holgura positivas.

$$x_3, x_4, x_5 \geq 0 \quad (4.5.3)$$

*Encontrar x_j para maximizar $2x_1 + 4x_2$.

*Sistema de 3 ecuaciones con 5 incógnitas.

$$*x_1 + 4x_2 + x_3 = 24$$

$$x_1 + x_2 + x_4 = 9$$

$$3x_1 + x_2 + x_5 = 21$$

$$\text{Si } x_1 = x_2 = 0$$

nombre de una *solución básica*, y *las variables cuyo valor se ha fijado reciben el nombre de *variables base*. Teniendo presente la definición de solución factible, se nota que el conjunto $x_1 = 0, x_2 = 0, x_3 = 24, x_4 = 9$ y $x_5 = 21$ es una *solución factible* aunque no óptima, ya que en este caso la función objetivo vale $m = 0$.

Haciendo referencia a la figura 6.5.11 que muestra gráficamente la región donde se cumplen las restricciones (6.5.8) a (6.5.10), se observa que la solución básica $x_1 = x_2 = 0$ y $x_3 = 24, x_4 = 9$ y $x_5 = 21$ corresponde al origen del sistema. Nótese además que el valor de las variables de holgura indica que no se está empleando ningún recurso en este punto.

$$x_3 = 24, x_4 = 9, x_5 = 21$$

*Variables cuyo valor se fija ($x_1, x_2 = 0$) se llaman variables base.

$$x_1 = x_2 = 0, x_3 = 24$$

$x_4 = 9, x_5 = 21$ son una solución factible no óptima.

Restricción	Valor en 0	Holgura
$x_1 + 4x_2 \leq 24$	$x_1 + 4x_2 = 0$	24
$x_1 + x_2 \leq 9$	$x_1 + x_2 = 0$	9
$3x_1 + x_2 \leq 21$	$3x_1 + x_2 = 0$	21

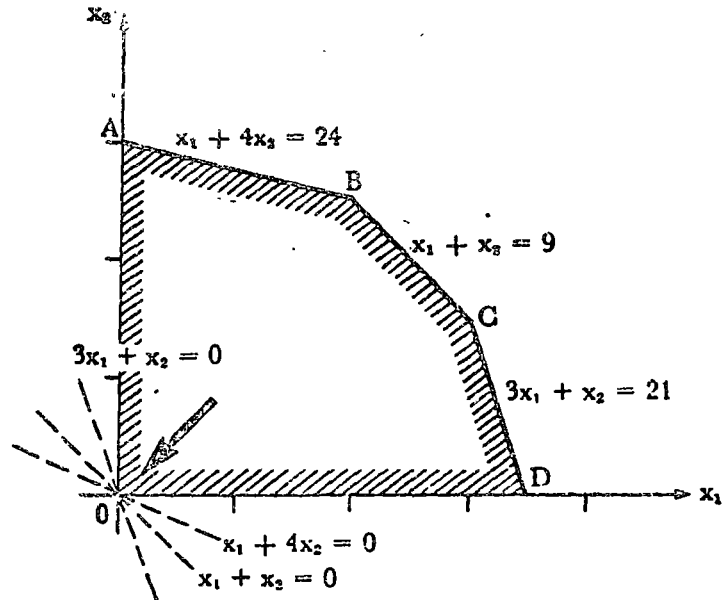
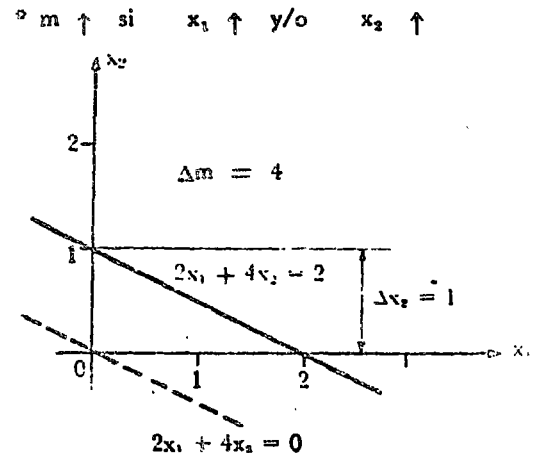


Fig. 6.5.11 Valor de las funciones de restricción en el punto de solución básica.

*Para incrementar el valor de la función objetivo se puede incrementar el valor de x_1 o el de x_2 o ambas. Se empieza por determinar en cuál variable un incremento unitario aumenta más la función objetivo. La fig. 6.5.12 ilustra que una unidad de incremento en x_2 aumenta en 4 el valor de m y un incremento unitario en x_1 sólo aumenta a m en 2 unidades, por lo tanto conviene, para encontrar el máximo lo más rápido posible aumentar el valor de x_2 , manteniendo $x_1 = 0$.



* $m \uparrow$ si $x_1 \uparrow$ y/o $x_2 \uparrow$

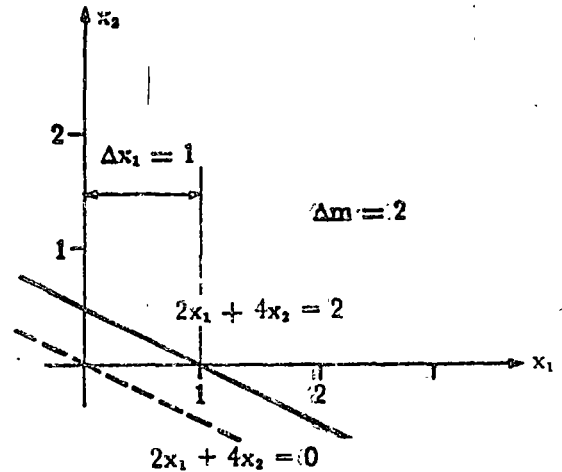


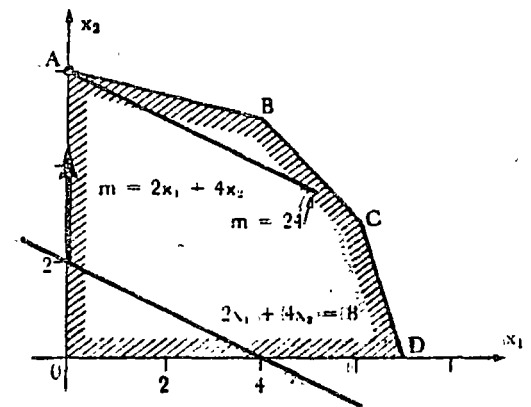
Fig. 6.5.12 Incremento de la función objetivo.

Para $x_1 = 0$, de (6.5.24) se obtiene:

$$\begin{aligned} x_3 &= 24 - 4x_2 \\ x_4 &= 9 - x_2 \\ x_5 &= 21 - x_2 \end{aligned} \quad (6.5.25)$$

El máximo valor de x_2 puede ser 6, ya que si es mayor de 6, $x_3 \leq 0$ y se violaría la condición $x_i \geq 0$, $i = 1, 2, \dots, 5$. Gráficamente, al ir moviendo la recta $2x_1 + 4x_2$ que representa la función objetivo paralelamente a sí misma, a lo largo de la recta $x_1 = 0$, se llega al punto A, otra esquina del polígono ABCD que define a la región de soluciones factibles.

La figura 6.5.13 muestra este traslado de la función objetivo $m = 2x_1 + 4x_2$.



A es la intersección de

$$x_1 + 4x_2 = 24 \text{ y } x_1 = 0$$

$$\text{En A } x_3 = 0$$

$$x_4 = 3$$

$$x_5 = 15$$

Fig. 6.5.13 Búsqueda del máximo de la función objetivo a lo largo de la recta $x_1 = 0$.

Del sistema de ecuaciones (6.5.25) para:

$$x_1 = 0 \text{ y } x_2 = 0$$

se tiene:

$$x_6 = 3$$

$$x_3 = 0$$

$$x_2 = 15$$

Como x_3 era la holgura de la ecuación de restricción (6.5.8) se deduce que en el punto *A el recurso limitado correspondiente a esa ecuación de restricción se ha empleado en su totalidad. En efecto el punto A se encuentra sobre la recta de ecuación

y la ecuación

*En resumen, en el punto A el valor de todas las variables del problema son:

*Las nuevas variables no básicas, es decir, las que son diferentes de cero son:

El valor de la función objetivo es:

*que resulta mayor que el valor de esta función en el 1er. punto explorado, el origen, donde valía cero.

*Cuando las variables básicas eran x_1 y x_2 , o sea en el 1er. paso de la solución del problema, también llamada *1ra. iteración*, el sistema de ecuaciones algebraicas que hubo que resolver eran:

*En este sistema las variables no básicas x_3 , x_4 y x_5 aparecían en una ecuación cada una, y esto facilitó su evaluación.

*Para proseguir con igual facilidad, se debe manipular algebraicamente a las ecuaciones (6.5.24) para que en cada una de

$$x_1 + 4x_2 + x_3 = 24 \quad (6.5.8)$$

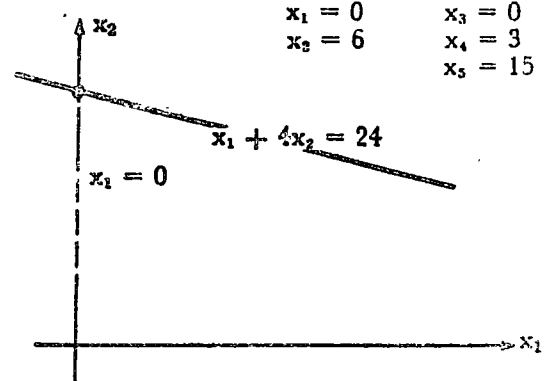
*A en la intersección de:

$$x_1 + 4x_2 = 24$$

$$x_1 = 0$$

*Valor de las variables en el punto A:

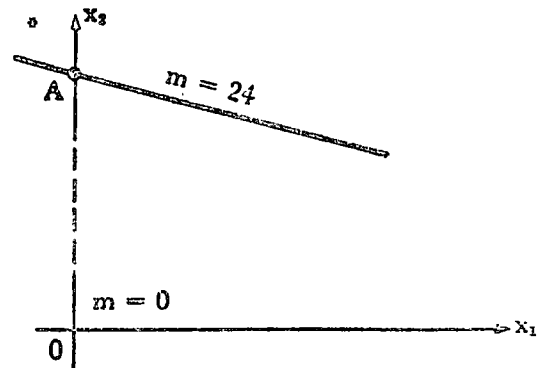
$$\begin{array}{ll} x_1 = 0 & x_3 = 0 \\ x_2 = 6 & x_4 = 3 \\ & x_5 = 15 \end{array}$$



*Variables no básicas $\neq 0$.

x_3 , x_4 y x_5

$$m = 2x_1 + 4x_2 = 24 \quad (6.5.26)$$



*Con $x_1 = x_2 = 0$ (variables básicas) el sistema de ecuaciones era:

$$\begin{array}{rcl} x_1 + 4x_2 + x_3 & = & 29 \\ x_1 + x_2 + x_4 & = & 9 \\ 3x_1 + x_2 + x_5 & = & 21 \end{array} \quad (6.5.24)$$

*Las variables x_3 , x_4 , $x_5 \neq 0$ (no básicas) aparecen una en cada ecuación.

*Manipule las ecuaciones para que las variables no básicas (x_3 , x_4 , x_5) aparezcan en una sola ecuación.

324 Optimización

ellas aparezca solamente una de las nuevas variables no básicas x_2, x_4 y x_5 de preferencia con coeficiente unitario. *De la 1er. ecuación del sistema (6.5.24).

se tiene al dividir entre 4

*Esta ecuación ya tiene una sola variable no básica x_2 con coeficiente unitario. *En la 2da. ecuación del sistema (6.5.24)

*aparecen dos variables no básicas, x_2 y x_4 .

Como x_2 ya quedó en la ecuación anterior se debe dejar en esta x_4 .

Restando de la ecuación

la ecuación anterior

*se elimina la variable x_2 . En efecto se tiene:

*Finalmente la última ecuación del sistema (6.5.24)

*Contiene las variables no básicas x_2 y x_5 . Hay que eliminar x_2 para que sólo quede una. Restando a esta ecuación la ecuación *(6.5.26) se elimina en efecto x_2 .

Realizando esta operación se obtiene:

*El sistema de ecuaciones de restricción ha quedado de la forma deseada:

*En este sistema de ecuaciones en cada una de ellas aparece solamente una de las variables no básicas.

*1er. ecuación.

$$x_1 + x_2 + 4x_2 = 24$$

$$\frac{1}{4} x_1 + \frac{1}{4} x_2 + x_2 = 6 \quad (6.5.26)$$

*Única v.n.b. x_2

*2da. ecuación

$$x_1 + x_2 + x_4 = 9$$

*v.n.b. x_2 y x_4

elimine x_2

$$(1) x_1 + x_2 + x_4 = 9$$

$$(2) \frac{1}{4} x_1 + \frac{1}{4} x_2 + x_2 = 24$$

*Se elimina x_2

$$(1) - (2) \frac{3}{4} x_1 - \frac{1}{4} x_2 + x_4 = 3$$

*Última ecuación

$$3x_1 + x_2 + x_5 = 21$$

*V.n.b. x_2 y x_5

elimine x_2

$$3x_1 + x_2 + x_5 = 21$$

$$- \left(\frac{1}{4} x_1 + x_2 + \frac{1}{4} x_3 = 6 \right)$$

$$\frac{11}{4} x_1 + x_5 - \frac{1}{4} x_3 = 15$$

*Nuevas ecuaciones de restricción.

$$\frac{1}{4} x_1 + \frac{1}{4} x_3 + x_2 = 6 \quad (6.5.26)$$

$$\frac{3}{4} x_1 - \frac{1}{4} x_3 + x_4 = 3 \quad (6.5.27)$$

$$\frac{11}{4} x_1 - \frac{1}{4} x_3 + x_5 = 15 \quad (6.5.28)$$

*En cada ecuación aparece una sola v.n.b. (x_2, x_4, x_5)

*La función objetivo hay que expresarla en función de las variables básicas, x_1 y x_3 , es decir, hay que eliminar x_2 . En la etapa anterior, esta función estaba expresada en función de x_1 y x_2 : despejando de la ecuación (6.5.26).

*despejando a x_2 se tiene:

*sustituyendo a x_2 en la función objetivo por este valor, se tiene:

La función objetivo ha quedado expresada en función de las variables básicas x_1 y x_3 y su valor para $x_1 = 0$ y $x_3 = 0$ es $m = 24$.

A continuación hay que determinar qué pasa con la función objetivo si se aumenta x_1 ó x_3 . Para incrementar m , y seguir satisfaciendo la condición de no negatividad de las variables debe mantenerse $x_3 = 0$, ya que dado el coeficiente negativo de x_3 , si x_3 aumenta, m disminuye. Debe incrementarse a x_1 . *Del sistema de ecuaciones de restricción para $x_3 = 0$, las variables no básicas en función de la variable base x_2 , quedan expresadas en la siguiente forma:

*Deben analizarse las ecuaciones (6.5.30) para determinar cuál es el máximo valor de x_1 , para el cual todas las variables no básicas x_2 , x_4 y x_5 sean mayores o iguales a cero. Se tiene

*Se ve que el máximo valor posible de la variable x_1 , sin que

Expresar m en función de las v.b.

$$\begin{aligned} (x_1, x_2) \\ m &= 2x_1 + 4x_2 \\ \frac{1}{4}x_1 + \frac{1}{4}x_3 + x_2 &= 6 \end{aligned} \quad (6.5.26)$$

*despejando a x_2

$$x_2 = 6 - \frac{1}{4}x_1 - \frac{1}{4}x_3$$

*Sustituyendo en la

$$\begin{aligned} m &= 2x_1 + 4\left(6 - \frac{1}{4}x_1 - \frac{1}{4}x_3\right) \\ m &= x_1 - x_3 + 24 \end{aligned} \quad (6.5.29)$$

* $m = f(\text{v.b.}, x_2 \text{ y } x_3)$

$$\text{para } x_1 = x_3 = 0 \rightarrow m = 24$$

* $m = x_1 - x_3 + 24$

$$\text{Si } x_3 \uparrow \quad m \downarrow$$

*Para $x_3 = 0$ las ecuaciones de restricción son:

$$\begin{aligned} x_2 &= 6 - \frac{x_1}{4} \\ x_4 &= 3 - \frac{3}{4}x_1 \\ x_5 &= 15 - \frac{11}{4}x_1 \end{aligned} \quad (6.5.30)$$

*Máximo valor de x_1 sin violar condiciones de no negatividad.

$$\begin{aligned} x_2 &= 6 - \frac{x_1}{4} & x_{1 \text{ máx}} &= 24 & x_2 &= 0 \\ x_4 &= 3 - \frac{3}{4}x_1 & x_{1 \text{ máx}} &= 4 & x_4 &= 0 \\ x_5 &= 15 - \frac{11}{4}x_1 & x_{1 \text{ máx}} &= \frac{60}{11} & x_5 &= 0 \end{aligned}$$

*Máximo valor de posible de $x_1 = 4 \rightarrow x_4 = 0$

326 Optimización

ninguna de las variables x_2 , x_4 y x_5 se vuelvan negativas es 4, para lo cual $x_4 = 0$.

*Se finaliza este paso y se tiene que $x_3 = x_4 = 0$. Estas variables se toman como base para el siguiente paso.

*Para $x_3 = x_4 = 0$, y $x_1 = 4$ el valor del resto de las variables es $x_2 = 5$ y $x_5 = 4$. Estos valores se obtienen del sistema de restricciones.

*El siguiente conjunto de valores de las variables constituye una nueva solución factible.

*Antes de continuar resulta ilustrativo interpretar gráficamente este segundo paso de solución. En este paso de solución se incrementa el valor de la variable básica x_1 de 0 a 4 *manteniendo a la otra variable básica $x_3 = 0$. Como x_3 es la variable de holgura de la 1er. ecuación de restricción $x_1 + 4x_2 \leq 24$ esta búsqueda de un mayor valor en la función objetivo se realiza a lo largo de la frontera AB de la zona de soluciones factibles tal como se ilustra en la figura 6.5.14.

*Para continuar debe volverse a manipular el sistema de ecuaciones (6.5.25) - (6.5.28), para dejar en cada ecuación una sola de las variables no básicas x_1 , x_2 y x_5 . Realizando operaciones algebraicas elementales sobre ese sistema similares a las descritas previamente se obtiene:

*Nuevas v.b. $x_3 = x_4 = 0$.

*Del sistema de restricciones: Si $x_3 = x_4 = 0$ y $x_1 = 4 \rightarrow x_2 = 5, x_5 = 4$

*Nueva solución factible.

$$x_1 = 4, x_2 = 5, x_3 = 0, x_4 = 0 \text{ y } x_5 = 4$$

*Interpretación gráfica de la 2da. iteración.
 $x_1 \uparrow$ de 0 a 4

*v.b. $x_3 = 0$ cst.

$$x_3 \text{ holgura de } x_1 + 4x_2 \leq 24$$

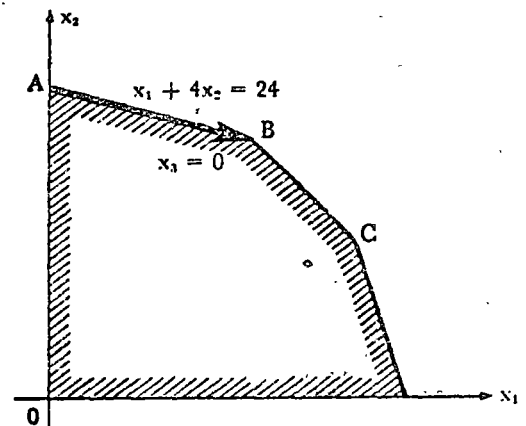


Fig. 6.5.14 Búsqueda del máximo de la función objetivo a lo largo de la recta AB. (2da. iteración).

*En cada ecuación de restricción una sola v.b. (x_1, x_2, x_5).

$$\begin{aligned} x_2 + \frac{1}{3} x_3 - \frac{1}{3} x_4 &= 5 \\ x_1 - \frac{1}{3} x_3 + \frac{4}{3} x_4 &= 4 \quad (6.5.31) \\ + \frac{2}{3} x_3 - \frac{11}{3} x_4 + x_5 &= 4 \end{aligned}$$

y volviendo a expresar la función objetivo en relación a las nuevas variables base x_3 y x_4 se tiene:

$$m = 23 - \frac{2}{3}x_3 - \frac{4}{3}x_4 \quad (6.5.32)$$

*No puede ↑ x_3 y x_4 porque m ↓

solución factible del paso anterior es óptima.

$$x_1 = 4, x_2 = 5, x_3 = 0, x_4 = 0 \text{ y } x_5 = 4$$

*Problema

$$m = 2x_1 + 4x_2 \quad (6.5.7)$$

restricciones

$$\text{(mecánicas)} \quad x_1 + 4x_2 + x_3 = 24 \quad (6.5.8)$$

$$\text{(andenes)} \quad x_1 + x_2 + x_4 = 20 \quad (6.5.9)$$

$$\text{(cargadores)} \quad 3x_1 + x_2 + x_5 = 21 \quad (6.5.10)$$

*Como en el último paso x_3 y x_4 eran nulas, la única forma de alterarlas, satisfaciendo simultáneamente la condición de no negatividad de las variables es incrementándolas, pero esto disminuiría el valor de m . Por lo tanto, la solución factible que a su vez es óptima es precisamente la solución obtenida en el paso anterior a saber:

*Recuérdese el planteamiento del problema:

Maximizar la función objetivo.

*sujeto a las restricciones

donde se recordará que la primera restricción la imponía la disponibilidad de mecánicos, la segunda estaba relacionada con la existencia de andenes y la tercera con la disponibilidad de cargadores.

*Al operar 4 camionetas chicas y 5 grandes, como indica la solución de problema ($x_1 = 4, x_2 = 5$), la primer variable de holgura es nula ($x_3 = 0$), la segunda también es nula ($x_4 = 0$), y la tercera vale 4 ($x_5 = 4$). Este conjunto de valores de la variable de holgura significa que el primer recurso (mecánicos) se aproveche en su totalidad al igual que el segundo (andenes). Mientras que del tercer recurso se emplea la cantidad disponible menos la holgura, es decir

* $x_1 = 4 \equiv$ operar 4 camionetas chicas $x_2 = 5 \equiv$ operar 5 camionetas grandes $x_3 = 0 \equiv$ se emplean todos los mecánicos $x_4 = 0 \equiv$ se emplean todos los andenes $x_5 = 4 \equiv$ se emplean $21 - 4$ mecánicos.

$$21 - x_5 = 21 - 4 = 17$$

*Para resolver un problema de programación lineal empleando el método simplex es necesario realizar repetitivamente diversas operaciones, como se acaba de ilustrar. Es posible sistematizar el método solución expuesto empleando la notación matricial.

*Sistematización del método empleando matrices.

Se empieza por formar una tabla o matriz cuyas columnas menos la última tienen por valor los coeficientes de las variables en las ecuaciones de restricción y en la función objetivo. En esta última ecuación debe cambiarse el signo de los coeficientes. La última columna tiene por valor los recursos disponibles y un cero en la última posición. Los elementos del último renglón de esta tabla, exceptuando el último, se llaman *los indicadores* del problema. Como se señala a continuación, si después de realizar las operaciones que se indican posteriormente, todos los indicadores son positivos, la búsqueda del óptimo ha terminado. Con objeto de familiarizar al lector con el método se presentan las ecuaciones en forma explícita y en notación matricial, tal como aparecen a continuación:

328 Optimización

Formulación explícita
1er. Etapa

$$\begin{aligned} x_1 + 4x_2 + x_3 &= 24 \\ x_1 + x_2 + x_4 &= 9 \\ 3x_1 + x_2 + x_5 &= 21 \\ 2x_1 + 4x_2 &= m \end{aligned}$$

Formulación matricial

$$\left[\begin{array}{ccccc|c} x_1 & x_2 & x_3 & x_4 & x_5 & b \\ 1 & \textcircled{4} & 1 & 0 & 0 & 24 \\ 1 & 1 & 0 & 1 & 0 & 9 \\ 3 & 1 & 0 & 0 & 1 & 21 \\ \hline -2 & -4 & 0 & 0 & 0 & 0 \end{array} \right] \begin{array}{l} 6 \leftarrow \\ 9 \\ 21 \end{array} \quad (6.5.32)$$

indicadores
*variables base

*El problema se inicia buscando una solución factible.

En este caso puede ser

Posteriormente se señala cómo puede sistematizarse la búsqueda de la 1er. solución factible.

*En seguida debe seleccionarse la columna con el término más negativo en el último renglón o sea el correspondiente a la función objetivo, con objeto de determinar el incremento de cuál variable hace crecer más rápidamente a la función objetivo. En este primer paso la segunda columna, correspondiente a x_2 tiene

esta propiedad, o sea más negativo el último renglón. *A continuación se dividen los elementos correspondientes a la disponibilidad de recursos, es decir, los elementos de la última columna, exceptuando el último, entre los correspondientes elementos de la columna seleccionada anteriormente, en este caso la segunda. El valor de estos cocientes se anota en una última columna y se selecciona el renglón con el valor menor de esta columna, en este caso el primero. En este ejemplo estos valores fueron 6, 9 y 21. Este problema se inicializó con $x_1 = x_2 = 0$.

*Los incrementos en x_2 resultan aumentar más la función objetivo que los de x_1 . La columna adicional indica que el máximo valor que puede darse a x_2 es de 6, sin hacer negativa alguna de las variables x_2, x_4 ó x_5 .

*Se ha encontrado hasta el momento que el primer renglón tiene el elemento más pequeño en la última columna y la segunda columna el más negativo en el último renglón. La intersección de este renglón (el primero) y esta columna, (la segunda) definen el elemento llamado pivote, en este caso 4.

*1er. solución factible: $x_1 = x_2 = 0$ (v. b.)
 $x_3 = 24, x_4 = x_5 = 21$ (v. n. b.)
 $x_1 = x_2 = 0, x_3 = 24, x_4 = 9$ y $x_5 = 21$.

*Busque columnas con último elemento más negativo.

$$\left[\begin{array}{ccccc|c} 1 & 4 & 1 & 0 & 0 & 24 \\ 1 & 1 & 0 & 1 & 0 & 9 \\ 3 & 1 & 0 & 0 & 1 & 21 \\ \hline -2 & -4 & 0 & 0 & 0 & 0 \end{array} \right]$$

↑

*Divida última columna entre columnas seleccionada anteriormente.

$$\left[\begin{array}{ccccc|c} \textcircled{2} & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ \hline & & & & & -4 \end{array} \right] \begin{array}{l} \textcircled{6} \quad \textcircled{6} \quad \textcircled{2} \\ 24 \\ 9 \\ 21 \\ 21 \end{array} \begin{array}{l} 6 \leftarrow \\ 9 \\ 21 \end{array}$$

* x_2 incrementa más rápidamente a m que x_1 .

6 ←
6 máximo valor de x_2
que no viola $x_1 \geq 0, x_4$

elemento más pequeño

$$\left[\begin{array}{ccccc|c} \textcircled{4} \text{ Pivote} & & & & & \\ & & & & & \\ & & & & & \\ \hline & & & & & -4 \end{array} \right] \begin{array}{l} 6 \leftarrow \\ 9 \\ 21 \end{array}$$

elemento más negativo

*Posteriormente se divide el renglón del pivote, en este caso el primero entre el pivote como se ilustra a continuación:

°División del renglón del pivote entre el pivote.

Formulación explícita

$$x_1 + 4x_2 + x_3 = 24$$

$$\frac{1}{4}x_1 + x_2 + \frac{1}{4}x_3 = 6$$

Formulación matricial

x_1	x_2	x_3	x_4	x_5	
1	4	1	0	0	24
$\frac{1}{4}$	1	$\frac{1}{4}$	0	0	6

Después se emplea esta última ecuación para eliminar la variable x_2 de las ecuaciones restantes del sistema. Como en la 2da. y 5er. ecuación x_2 tiene uno por coeficiente hasta restar el renglón o sea la ecuación del pivote de cada una de esas ecuaciones, se tiene:

Formulación explícita

2da. Esc. $x_1 + x_2 + x_4 = 9$

Ecs. del Pivote norm. $\frac{1}{4}x_1 + x_2 + \frac{1}{4}x_3 = 6$

Resta $\frac{3}{4}x_1 - \frac{1}{4}x_3 + x_4 = 3$

3er. Ecs. $3x_1 + x_2 + x_5 = 21$

Ecs. del Pivote norm. $\frac{1}{4}x_1 + x_2 + \frac{1}{4}x_3 = 6$

Resta $\frac{11}{4}x_1 - \frac{1}{4}x_3 + x_5 = 15$

Formulación matricial

x_1	x_2	x_3	x_4	x_5	
1	1	0	1	0	9
$\frac{1}{4}$	1	$\frac{1}{4}$	0	0	6
$\frac{3}{4}$	0	$-\frac{1}{4}$	1	0	3
3	1	0	0	1	21
$\frac{1}{4}$	0	$\frac{1}{4}$	0	0	6
$\frac{11}{4}$	0	$-\frac{1}{4}$	0	0	15

Para eliminar a x_2 de la función objetivo, es necesario multiplicar la ecuación del pivote por -4 y restada de la función objetivo, ya que -4 es el coeficiente de x_2 en la función objetivo, tal como se ilustra:

Formulación explícita

Función objetivo $-x_1 - 4x_2 = 0$

Renglón del piv. norm. $x(-4) -x_1 - 4x_2 - x_3 = -24$

Resta $-x_1 + x_3 = 24$

Formulación matricial

x_1	x_2	x_3	x_4	x_5	
-2	-4	0	0	0	0
-1	-4	-1	0	0	-24
-1	0	1	0	0	24

Después de realizadas las funciones anteriores x_2 , x_4 y x_5 , las variables no básicas, quedan multiplicadas por 1, tal como muestra el siguiente cuadro:

Formulación explícita	=	6	Formulación matricial					24	
$\frac{1}{4}x_1 + x_2 + \frac{1}{4}x_3$			$\frac{1}{4}$	x_2	$\frac{1}{4}$	x_4	x_5	6	24
$\frac{3}{4}x_1 - \frac{1}{4}x_3 + x_4$		3	$\frac{3}{4}$	0	$-\frac{1}{4}$	1	0	3	4
$\frac{11}{4}x_1 - \frac{1}{4}x_3 + x_5$		15	$\frac{11}{4}$	0	$-\frac{1}{4}$	0	1	15	$\frac{60}{11}$
$-x_1 + x_3$		24	-1	0	1	0	0	24	

*
↑

En este cuadro la columna con el elemento más negativo es la 1ra. y el cociente de la primera columna entre la de las restricciones es $24, 4, \frac{60}{11}$, o sea los elementos de la columna auxiliar situada

fuera de las llaves de la matriz. Como el elemento más pequeño es 4; el del segundo renglón, el pivote es el aumento de la 1ra. columna y del segundo renglón y se va a emplear para eliminar x_1 de las ecuaciones restantes. Se empieza dividiendo el renglón del pivote entre el pivote, tal como se ilustra, para obtener la *ecuación del pivote normalizada. (e.p.n)

*Ecuación del pivote normalizada: e.p.n.

Formulación explícita	=	x_1	Formulación matricial						
Ecs. del pivote	$\frac{3}{4}x_1 - \frac{1}{4}x_3 + x_4$	3	$\frac{3}{4}$	x_2	$\frac{1}{4}$	x_4	x_5	0	3
Ecs. del piv. norm.	$x_1 - \frac{1}{3}x_3 + \frac{4}{3}x_4$	4	1	0	$-\frac{1}{3}$	$\frac{4}{3}$	0	0	4

Para eliminar x_1 de la primer ecuación es necesario restarle la ecuación del pivote multiplicada por $\frac{1}{4}$ que es el coeficiente de x_1 en la 1er. ecs.

Formulación explícita	=	x_1	Formulación matricial						
1er. Ecs.	$\frac{1}{4}x_1 + x_2 + \frac{1}{4}x_3$	6	$\frac{1}{4}$	x_2	$\frac{1}{4}$	x_4	x_5	0	6
E.p.n $\times \left(\frac{1}{4}\right)$	$\frac{1}{4}x_1 - \frac{1}{12}x_3 + \frac{1}{3}x_4$	1	$\frac{1}{4}$	0	$-\frac{1}{12}$	$\frac{1}{3}$	0	0	1
Resta	$0x_2 + \frac{1}{3}x_3 - \frac{1}{3}x_4$	5	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	0	0	5

Para eliminar x_1 de la 3er. ecuación hay que restarle la del pivote normalizada multiplicada por $\frac{11}{4}$.

Formulación explícita				Formulación matricial					
				x_1	x_2	x_3	x_4	x_5	
3er. Ecs.	$\frac{11}{4} x_1 - \frac{1}{4} x_3$	$+ x_5 = 15$		$\frac{11}{4}$	0	$-\frac{1}{4}$	0	1	15
E.p.n x	$\frac{11}{4} x_1 - \frac{11}{12} x_3 + \frac{11}{3} x_4$	$= 11$		$\frac{11}{4}$	0	$-\frac{11}{12}$	$\frac{11}{3}$	0	11
Resta	$\frac{2}{3} x_3 - \frac{11}{3} x_4 + x_5$	$= 4$		0	0	$\frac{2}{3}$	$-\frac{11}{3}$	1	4

Y finalmente para eliminar x_1 de la función objetivo debe restársele la normalizada del pivote multiplicada por -1 , coeficiente de x_1 en la función objetivo, tal como se muestra:

Formulación explícita				Formulación matricial					
				x_1	x_2	x_3	x_4	x_5	
Func. obj.	$-x_1 + x_3$	$= 24$		-1	0	1	0	0	$+ 24$
E.p.n x (-1)	$-x_1 + \frac{1}{3} x_3 - \frac{4}{3} x_4$	$= -4$		-1	0	$\frac{1}{3}$	$-\frac{4}{3}$	0	$- 4$
Resta	$\frac{2}{3} x_3 + \frac{4}{3} x_4$	$= 28$		0	0	$\frac{2}{3}$	$\frac{4}{3}$	0	28

Una vez realizadas estas operaciones el cuadro queda como se muestra:

3er. Etapa

Formulación explícita				Formulación matricial					
				x_1	x_2	x_3	x_4	x_5	
$x_2 + \frac{1}{3} x_3 - \frac{1}{3} x_4$	$= 5$			0	1	$\frac{1}{3}$	$-\frac{1}{3}$	0	5
$x_1 + \frac{1}{3} x_3 - \frac{4}{3} x_4$	$= 4$			1	0	$-\frac{1}{3}$	$\frac{4}{3}$	0	4
$\frac{2}{3} x_3 - \frac{11}{3} x_4 + x_5$	$= 4$			0	0	$\frac{2}{3}$	$-\frac{11}{3}$	1	4
$\frac{2}{3} x_3 + \frac{4}{3} x_4$	$= 28$			0	0	$\frac{2}{3}$	$\frac{4}{3}$	0	28

indicadores

*Antes de continuar es necesario revisar el signo de los elementos del último renglón, exceptuando el último o sea el de los indicadores. Cuando todos son positivos se ha encontrado el óptimo.

°Reviste indicadores, si todos ≥ 0 se ha encontrado el óptimo \equiv último elemento de la tabla.

*El valor del óptimo está dado por el último elemento del último renglón:

°Óptimo.

332 Optimización

Como en esta etapa ya todos los indicadores son positivos, la búsqueda del óptimo ha terminado. El valor óptimo es 28. *Las variables con coeficiente diferente de cero en el último renglón valen cero, en este caso $x_3 = x_4 = 0$.

Del último sistema de ecuaciones:

para

se obtiene:

*Es posible obtener el valor de los niveles de actividad en el punto óptimo a partir de la última tabla del método simplex.

*La llamada última tabla del método simplex tiene indicadores únicamente positivos.

Para obtener los niveles óptimos de actividad de la última tabla basta numerar los renglones de acuerdo con la posición donde se encuentra una columna unitaria, es decir, una columna con un solo uno y el resto ceros.

Para aclarar este paso nos referimos a la fig. 6.5.15 donde aparece la tabla terminal del ejemplo. El 1er. renglón tiene su 1er. columna unitaria en la segunda posición, por eso se le designa con 2, y el 2do. renglón tiene la 1er. columna unitaria en la primer posición. Se le designa con 1. Se continúa hasta terminar con todos los renglones menos el último. En el punto óptimo las variables *diferentes de cero* tienen por índice el número con el que se han designado los renglones y su valor está dado en la última columna. En este caso $x_2 = 5$, $x_1 = 4$ y $x_5 = 4$.

*Último renglón

$$0 \quad 0 \quad \frac{2}{3} \quad \frac{4}{3} \quad 0 \quad 28$$

$$x_3 = x_4 = 0$$

$$x_2 + \frac{1}{3}x_3 - \frac{1}{3}x_4 = 5$$

$$x_1 - \frac{1}{3}x_3 + \frac{4}{3}x_4 = 4$$

$$-\frac{2}{3}x_3 - \frac{11}{3}x_4 + x_5 = 4$$

$$x_3 = x_4 = 0$$

$$x_2 = 5, x_1 = 4, x_5 = 4$$

*Obtención de los niveles óptimos de actividad a partir de la última tabla simplex.

*Indicadores de la última tabla:

$$0 \quad 0 \quad \frac{2}{3} \quad \frac{4}{3} \quad -0$$

$$\text{indicadores} \geq 0$$

		Columnas unitarias				
		↓	↓			↓
		x_1	x_2	x_3	x_4	x_5
designación de los renglones	2	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	5
	1	1	0	$-\frac{1}{3}$	$\frac{4}{3}$	4
	5	0	0	$\frac{2}{3}$	$-\frac{11}{3}$	4
		0	0	$\frac{2}{3}$	$\frac{4}{3}$	28

Fig. 6.5.15 Tabla terminal del problema.

En la tabla 6.5.2 se resumen los diferentes pasos que se siguen en la solución de un problema de programación lineal mediante el método simplex. En las matrices de esta tabla la columna y el renglón marcados con una flecha definen la posición del pivote y las columnas marcadas con un asterisco (*) corresponden a las variables base, es decir, que se han tomado como nulas.

Tabla 6.5.2 Solución del ejemplo 6.5.2 por el método simplex

Solución analítica del problema de programación lineal

Problema

Función objetivo:

$$m = 2x_1 + 4x_2 \text{ (máx.)}$$

Restricciones

$$\begin{aligned} x_1 + 4x_2 &\leq 24 & \text{(a)} \\ x_1 + x_2 &\leq 9 & \text{(b)} \\ 3x_1 + x_2 &\leq 21 & \text{(c)} \\ x_1 &\geq 0 & \text{(d)} \\ x_2 &\geq 0 & \text{(e)} \end{aligned}$$

Formulación explícita

1er. Etapa

$$\begin{aligned} x_1 + 4x_2 &+ x_3 \\ x_1 + x_2 &+ x_4 \\ x_1 + x_2 &+ x_5 \\ +2x_1 + 4x_2 &+ 0 \end{aligned}$$

Formulación matricial

$$= \begin{matrix} 24 \\ 9 \\ 21 \\ m \end{matrix} \left[\begin{array}{cc|cc|c} x_1 & x_2 & x_3 & x_4 & x_5 & & \\ \hline 1 & 4 & 1 & 0 & 0 & 24 & 6 \leftarrow \\ 1 & 1 & 0 & 1 & 0 & 9 & 9 \\ 3 & 1 & 0 & 0 & 1 & 21 & 21 \\ \hline -2 & -4 & & & & 0 & \\ * & * & & & & & \end{array} \right]$$

Variables base

$$x_1 = x_2 = 0 *$$

Solución factible

$$x_3 = 24 \quad x_4 = 9 \quad x_5 = 21$$

Incremento unitario en $x_1 \rightarrow$ Incremento en $m = 2$

Incremento unitario en $x_2 \rightarrow$ Incremento en $m = 4$

$$\begin{array}{ll} x_1 = 0 & x_2 \text{ max} \\ x_3 = 24 - 4x_2 = 0 & 6 \leftarrow \\ x_4 = 9 - x_2 = 0 & 9 \\ x_5 = 21 - x_2 = 0 & 21 \\ \text{Con } x_2 = 6 \rightarrow x_3 = 0 & \end{array}$$

334 Optimización

2da. Etapa

$$\frac{1}{4}x_1 + x_2 + \frac{1}{4}x_3$$

$$\frac{3}{4}x_1 - \frac{1}{4}x_3 + x_4$$

$$\frac{11}{4}x_1 - \frac{1}{4}x_3 + x_5$$

$$x_1 - x_3 + 24$$

Nuevas variables base
 $x_1 = 0 \quad x_3 = 0$ *

$$= 6 \left[\begin{array}{ccccc|c} \frac{1}{4} & 1 & \frac{1}{4} & 0 & 0 & 6 \\ \frac{3}{4} & 0 & -\frac{1}{4} & 1 & 0 & 3 \\ \frac{11}{4} & 0 & -\frac{1}{4} & 0 & 1 & 15 \\ \hline -1 & 0 & 1 & 0 & 0 & +24 \end{array} \right] \begin{array}{l} 24 \\ 4 \leftarrow \\ \frac{60}{11} \\ \end{array}$$

Solución factible

$$x_2 = 6, x_4 = 3, x_5 = 15$$

Incremento unitario en $x_1 \rightarrow$ Incremento en $m = 1$
 Incremento unitario en $x_3 \rightarrow$ Decremento en m

$x_3 = 0$	$x_1 \text{ max}$
$x_2 = 6 - \frac{1}{4}x_1$	24
$x_4 = 3 - \frac{3}{4}x_1$	4 \leftarrow
$x_5 = 15 - \frac{11}{4}x_1$	$\frac{60}{11}$

Nuevas variables base
 $x_3 = x_4 = 0$

3er. Etapa

$$x_2 + \frac{1}{3}x_3 + \frac{1}{3}x_1$$

$$x_1 - \frac{1}{3}x_3 + \frac{4}{3}x_4$$

$$\frac{2}{3}x_3 - \frac{11}{3}x_4 + x_5$$

$$-\frac{2}{3}x_3 + \frac{4}{3}x_4 + 28$$

$$= 5 \left[\begin{array}{ccccc|c} 0 & 1 & \frac{1}{3} & -\frac{1}{3} & 0 & 5 \\ 1 & 0 & -\frac{1}{3} & \frac{4}{3} & 0 & 4 \\ 0 & 0 & \frac{2}{3} & -\frac{11}{3} & 1 & 4 \\ \hline 0 & 0 & +\frac{2}{3} & +\frac{4}{3} & 0 & +28 \end{array} \right]$$

Solución factible

$$x_1 = 4, x_2 = 5 \text{ y } x_5 = 4$$

*Antes de continuar es necesario indicar cómo se obtiene la 1ra. solución factible en el problema de programación lineal.

Recuérdese que el problema de programación lineal tiene n incógnitas, los niveles de actividad y existen m ecuaciones de restricción. Si todas las ecuaciones de restricción son desigualdades de tipo "menor o igual a cero" se introducen m variables de holgura, y en el primer paso de solución, igualando a cero las variables del problema (niveles de actividad) las variables de holgura toman determinados valores, que forman una 1er. solución factible. En este caso se encuentra el problema del ejemplo anterior, cuyas restricciones eran:

*Obtención de la 1er. solución factible.

$$\begin{aligned} x_1 + 4x_2 &\leq 24 \rightarrow x_1 + 4x_2 + x_3 &= 24 \\ x_1 + x_2 &\leq 9 \rightarrow x_1 + x_2 + x_4 &= 9 \\ 3x_1 + x_2 &\leq 21 \rightarrow 3x_1 + x_2 + x_5 &= 21 \end{aligned}$$

Niveles de actividad x_1, x_2
 Variables de holgura x_3, x_4, x_5
 1er. Solución factible
 Niveles de actividad $x_1 = x_2 = 0$
 Variables de holgura $x_3 = 24; x_4 = 9$ y $x_5 = 21$

En algunos casos algunas restricciones son mayores que cero o igualdades. *En este caso habrá menos de m variables de holgura y no se puede formar la 1er. solución factible igualando los niveles de actividad a cero, como se ilustra a continuación.

*Con restricciones de igualdad.

Sujeto a las siguientes restricciones:

$$\text{Max : } m = 2x_1 + 4x_2 + x_3$$

$$\begin{aligned} x_1 + 2x_2 + x_3 &\leq 4 \\ 2x_1 + 4x_2 + x_3 &= 8 \\ 4x_1 + 2x_2 - x_3 &\geq 6 \end{aligned}$$

*La introducción de dos variables de holgura, ya que sólo hay dos desigualdades convierte a las ecuaciones de restricción en:

o Con dos variables de holgura.

$$\begin{aligned} x_1 + 2x_2 + x_3 + x_4 &= 4 \\ 2x_1 + 4x_2 + x_3 &= 8 \\ 4x_1 + 2x_2 + x_3 - x_5 &= 6 \end{aligned}$$

Si se da a los niveles de actividad x_1, x_2, x_3 el valor cero, como en el caso anterior, para inicializar el problema, se viola la segunda restricción, ya que

$$2x_1 + 4x_2 + x_3 \left\{ \begin{aligned} &\neq 8 \\ &x_1 = x_2 = x_3 = 0 \end{aligned} \right.$$

De manera que no es posible obtener en esta forma la 1er. solución factible para iniciar la solución del problema de programación lineal; si se presenta un problema de este tipo es necesario incluir *variables artificiales* en el problema. Una por cada ecuación de restricción que sea una igualdad y una desigualdad del tipo "mayor o igual que cero". En el ejemplo es necesario introducir las variables artificiales x_6 y x_7 ya que hay una igualdad y una desigualdad del tipo "mayor o igual que cero" entre las restricciones. El sistema de ecuaciones de restricción, después de introducir estas variables queda:

*Variables artificiales.

$$\begin{aligned} x_1 + 2x_2 + x_3 + x_4 &= 4 \\ 2x_1 + 4x_2 + x_3 + x_6 &= 8 \\ 4x_1 + 2x_2 - x_3 + x_5 + x_7 &= 6 \end{aligned}$$

336 Optimización

En este caso asignando a las variables estructurales y a una de las de holgura el valor cero, se puede obtener la primer solución factible.

*En efecto con $x_1 = x_2 = x_3 = x_5 = 0$, el resto de las variables asume el valor de $x_4 = 4$, $x_6 = 8$ y $x_7 = 12$.

*Las variables artificiales no deben aparecer en la solución final en la función objetivo. Para asegurarse de que esto no suceda, se deben incluir en la función objetivo con grandes coeficientes negativos en problemas de maximización. Estos grandes coeficientes negativos aseguran que las variables artificiales deben ser nulas para maximizar la función objetivo.

Antes de terminar con esta sección para estudiar el problema dual en la siguiente, es necesario enunciar un importante *teorema **de programación lineal y explicar por qué este método es un método de gradiente.

*Debido a este teorema la búsqueda del óptimo se realiza a lo largo de la frontera de la zona definida por las restricciones, como se ilustró en la solución gráfica y analítica del problema del ejemplo 6.5.2. En la figura 6.5.9 referente a este problema, se recuerda que el método simplex empezó por calcular el valor de la función objetivo en O, después en A y finalmente en B. No fue, sin embargo, necesario evaluarla en todos los vértices del polígono OABCD. Faltaron los puntos C y D. El método permite ir buscando valores siempre crecientes (en un problema de maximización) de la función objetivo en los vértices del polígono. *Esta búsqueda se realiza siempre a lo largo de aquella arista donde el valor de la función objetivo crece (o decrece) con mayor rapidez. Por esta razón se trata de un método de gradiente. El método permite descubrir cuándo se ha encontrado el valor óptimo, sin necesidad de tener que evaluar en general la función objetivo en todos los vértices del polígono y tener finalmente que buscar el valor óptimo de la función objetivo entre estos valores.

Un método de fuerza bruta para encontrar el óptimo consistiría en evaluar la función objetivo en todos los vértices y después buscar el máximo ó mínimo de ésta entre todos estos valores

*Con $x_1 = x_2 = x_3 = x_5 = 0$,
 $x_4 = 4$, $x_6 = 8$ y $x_7 = 12$.

*Introducir en problemas de maximización las variables artificiales con grandes coeficientes negativos.

*Teorema.

Puede demostrarse **que en un problema de programación lineal con las restricciones definiendo una zona convexa, el punto óptimo (ya sea máximo o mínimo de la función objetivo) se encuentra siempre en la frontera de la zona convexa definida por las restricciones.

*Búsqueda a lo largo de la frontera.

*Búsqueda en dirección de máxima variación de la función objetivo.

El método simplex no solamente reduce el número de vértices donde hay que calcular la función objetivo, sino al ir de paso en paso incrementando (o decrecentando) el valor de la función objetivo hace innecesaria la búsqueda final del óptimo. *En problemas con gran número de variables, el no tener que explorar todos los vértices y no tener que almacenar para una búsqueda final del óptimo el valor de las coordenadas de los vértices y de la función objetivo, ahorra mucho tiempo y requerimiento de memoria al procesarse digitalmente estos problemas. Desde luego que esta ventaja computacional tiene como precio las restricciones que impone al modelo matemático el problema, las de linealidad en sus ecuaciones y de convexidad y de la zona de soluciones factibles. Afortunadamente existen múltiples problemas, de gran interés para el analista de sistemas, en los que puede plantearse un modelo matemático con las restricciones anteriores.

*Se ahorra tiempo se computa y memoria.

6.5.5 Problema dual

*Se indicó en la sección 6.5.2 que el problema de programación lineal puede plantearse en forma matricial de la siguiente manera:

*Sujeto a las restricciones

donde x el vector de niveles de actividad, b el de restricciones y c el de costos. La matriz A tiene por elementos los coeficientes estructurales del problema.

La ilustración del problema dual puede realizarse con el ejemplo 6.5.2, sin embargo no se le empleará, con objeto de introducir otro tipo de problema.

*En un taller se cuenta con tres máquinas A, B, y C. Se emplean para fabricar dos productos 1 y 2. La tabla 6.5.3 muestra las horas de maquinado que requiere cada producto, las horas disponibles en cada tipo de máquina y la ganancia que se obtiene en la venta de cada producto.

°Formulación matricial.

$$\text{máx: } m = c^T x \quad (6.5.21)$$

°restricciones

$$Ax \leq b \quad (6.5.22)$$

$$x \geq 0 \quad (6.5.23)$$

Ejemplo 6.5.4.

°Las máquinas A, B y fabrican los productos 1 y 2.

Tabla 6.5.3 Datos para el ejemplo 6.5.4

Tipo de máquina Producto Horas disponibles

A	1	2	200
B	1	1	125
C	1	0	100
Beneficio	2	3	

338 Optimización

*Se trata de planear la producción de manera que se obtenga la máxima ganancia posible. Plantee el modelo matemático para este problema.

Sean x_1 y x_2 las cantidades del producto 1 y 2 fabricados.

*El objetivo será por lo tanto maximizar la ganancia, es decir

*Para producir x_1 unidades del producto 1 y x_2 del 2 se requieren las siguientes horas de la máquina A que están restringidas a 200:

*En forma similar las restricciones que provienen de la máquina B y C son:

Desde luego que no tiene significado físico producir unidades negativas, por lo tanto:

*El planteamiento matricial del problema es:

*Sujeto a las restricciones:

*A continuación se introduce el llamado problema dual o dual simplemente, del problema de programación lineal.

Si m es el número de restricciones y n el número de variables del problema, para definir el dual es necesario introducir un vector w de m componentes cuya interpretación se dará posteriormente. El dual del problema de programación lineal es otro problema cuya formulación matricial, comparada con la del último aparece en el siguiente cuadro:

*Planeación de la producción para maximización de la ganancia.

Solución

*Cantidad fabricada x_1 y x_2 .

*Maximizar la ganancia

$$\text{máx: } m = 2x_1 + 3x_2$$

*Carga de la máquina A

$$x_1 + 2x_2 \leq 200$$

*Carga de las máquinas B y C

$$x_1 + x_2 \leq 125$$

$$x_1 \leq 100$$

$$x_1, x_2 \geq 0$$

*Maximizar

$$\text{máx: } m = \begin{bmatrix} 2 \\ 3 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

*Restricciones

$$\begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 200 \\ 125 \\ 100 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq 0$$

*Problema dual.

* m restricciones
 n variables

Introduzca vector w

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}$$

Problema original
o primo
máx: $m = c^T x$
sujeto a las restricciones:

$$\begin{aligned} Ax &\leq b \\ x &\geq 0 \end{aligned}$$

Problema dual

$$\text{mín: } n = b^T w$$

$$\begin{aligned} A^T w &\geq c \\ w &\geq 0 \end{aligned}$$

A continuación se ilustra el planteamiento del problema dual.

Plantee el problema dual del ejemplo 6.5.4.

La solución aparece en el siguiente cuadro:

Problema original
o primo

$$\text{máx: } m = \begin{bmatrix} 2 \\ 3 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Problema dual

$$\text{mín: } n = \begin{bmatrix} 200 \\ 125 \\ 100 \end{bmatrix}^T \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

Sujeto a las restricciones:

$$\begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 125 \\ 200 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq 0$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \geq \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \geq 0$$

El teorema más importante de la programación lineal establece la siguiente relación entre el problema original o primo y el problema dual:

Teorema: La función objetivo m de un problema de maximización de programación lineal asume su valor máximo si y solamente si la función objetivo n del problema dual correspondiente alcanza un mínimo y en este caso.

$$\text{máx: } m = \text{mín } n.$$

Además, si P y Q son soluciones factibles tales que en $m(P) = n(Q)$, entonces las soluciones P y Q son los óptimos del problema primo y del problema dual respectivamente.

La demostración de este teorema aparece en la mayoría de los textos de programación lineal (ref. 4).

*Una primer aplicación de este teorema se encuentra en la solución de problemas de minimización.

°Aplicación del teorema dual a problemas de minimización.

340 Optimización

Antes de una interpretación económica al problema dual se resolverá el problema de producción del ejemplo 6.5.4. Este ejemplo no sólo sirve como repaso del método simplex sino muestra cómo la tabla terminal de este problema permite resolver tanto el problema original como el dual.

Empleando el método simplex resuelva el problema de producción del ejemplo 6.5.4.

A continuación aparecen las diferentes tablas que se establecen hasta encontrar la solución con el pivote en cada ocasión encerrado en un círculo y las columnas de las variables base marcadas con un asterisco (*).

Ejemplo 6.5.6

Solución:

Tabla 6.5.3 Solución del ejemplo 6.5.4

niveles de actividad		variables de holgura							
x_1	x_2	x_3	x_4	x_5	b				
1	2	1	0	0	200	100 ←			
1	1	0	1	0	125	125			
1	0	0	0	1	100	∞			
-2	-3	0	0	0	0	Punto O			
*	*								
	↑								
$\frac{1}{2}$	1	$\frac{1}{2}$	0	0	100	200			
$\frac{1}{2}$	0	$\frac{1}{2}$	1	0	25	50 ←			
1	0	0	0	1	100	100			
$-\frac{1}{2}$	0	$\frac{3}{2}$	0	0	300				
*		*							
	↑								
Variables con valor dado por la última columna	{	x_2	0	1	1	-1	0	75	
	{	x_1	1	0	-1	2	0	50	
	{	x_5	0	0	1	-2	1	50	
	{	indicadores	0	0	1	1	0	325	Punto E

Siguiendo las reglas expuestas, previamente se puede obtener de inmediato la solución del problema de la última tabla. A saber:

*El valor máximo de la función objetivo es precisamente 325] el valor de las variables es:

*Máximo 325.

$x_1 = 50, x_2 = 75, x_3 = x_4 = 0$ y $x_5 = 50$

Con objeto de aclarar el método también se incluye la solución gráfica en la figura 6.5.16.

A continuación se señala *cómo se obtiene la solución del problema dual de la tabla final del método simplex del problema original.

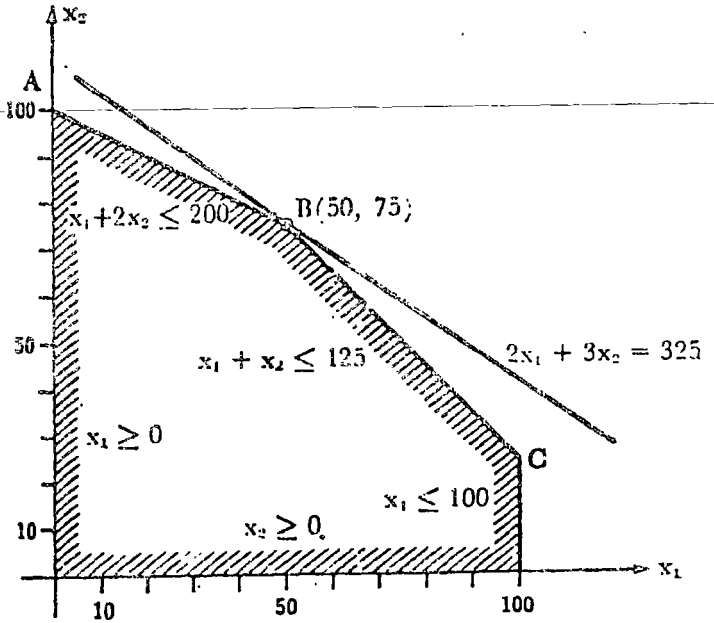


Fig. 6.5.16 Solución gráfica del ejemplo 6.5.7.

La tabla 6.5.4 muestra la tabla final del método simplex del problema original y cómo se obtienen de ella resultados del problema original y del dual.

Tabla 6.5.4 Tabla final del método simplex del problemas.

	niveles de actividades				variables de holgura			
	x_1	$x_2 \dots$	x_n	x_{n+1}	$x_{n+2} \dots$	x_{n+m}		
Renglón con la x_1 i'sima columnaritaria							Valor de i'simo nivel de actividad	
							} renglones correspondientes a las m restricciones	
				q_1	$q_2 \dots$	q_m	Valor del máximo en el problema original o del mínimo en el dual	
				valores del vector w en el problema dual				
	indicadores (todos positivos)							

El siguiente ejemplo ilustra el empleo de la tabla 6.5.4 para resolver el problema original y su dual.

Ejemplo 6.5.7

342 Optimización

Obtenga de la tabla final del método simplex la solución del problema original y del dual del ejemplo 6.5.4.

Solución:

A continuación aparece el planteamiento original del problema y el del dual.

Problema original	Problema dual
$\text{máx: } m = \begin{bmatrix} 2 \\ 3 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$	$\text{mín: } n = \begin{bmatrix} 200 \\ 125 \\ 100 \end{bmatrix}^T \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$

Sujeto a las restricciones

$\begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 200 \\ 125 \\ 100 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \geq \begin{bmatrix} 2 \\ 3 \end{bmatrix}$
$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq 0$	$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \geq 0$

De la tabla 6.5.3 se tiene:

$$\begin{array}{l} x_2 \\ x_1 \\ x_3 \\ 0 \end{array} \left[\begin{array}{cccc|c} 0 & 1 & 1 & -1 & 0 & 75 \\ 1 & 0 & -1 & 2 & 0 & 50 \\ 0 & 0 & 1 & -2 & 1 & 50 \\ 0 & 0 & 1 & 1 & 0 & 325 \end{array} \right]$$

$$m \Big|_{\text{máx}} = 325 \qquad n \Big|_{\text{mín}} = 325$$

$$\begin{array}{l} \underline{x} \\ \text{máx} \end{array} = \begin{bmatrix} 50 \\ 75 \end{bmatrix} \qquad \begin{array}{l} \underline{w} \\ \text{mín} \end{array} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

*En efecto el valor mínimo de la función objetivo en el problema dual es:

*Valor mínimo de la función objetivo en el dual:

$$n = \begin{bmatrix} 200 \\ 125 \\ 100 \end{bmatrix}^T \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = 325$$

*A continuación se dará una interpretación económica a la solución del problema dual. Recuerdese el planteamiento del problema original y del problema dual:

*Interpretación económica del problema dual.

Problema original

$$\text{máx: } m = c^T x$$

sujeto a las restricciones

$$Ax \leq b$$

$$x \geq 0$$

Problema dual

$$\text{mín: } n = b^T w$$

$$A^T w \geq c$$

$$w \geq 0$$

Para el problema de asignación de trabajo a las máquinas de una fábrica (ejemplo 6.5.4), el vector x representa la cantidad de artículos producidos y el vector b representa la disponibilidad de horas-máquina, y c el costo de los artículos producidos.

*Del teorema de la dualidad se tiene:

donde $c^T x$ es la cantidad a maximizar en el problema original y representa el beneficio total que se obtiene al producir x artículos.

*La cantidad $b^T w$ es el producto de la disponibilidad de horas-máquina por un vector w . La forma de esta ecuación hace pensar que w representa el costo de operar las máquinas por unidad de tiempo. Para el ejemplo que se discute:

Si w_1, w_2, w_3 son los costos de operación por hora de las máquinas A, B, y C respectivamente, la suma anterior en efecto representa el costo total de operación. *Estos costos de operación, o sea las componentes del vector w reciben el nombre de *precios sombra*.

Es necesario ahora interpretar la otra ecuación del problema dual:

*Recuérdese que la componente a_{ij} de la matriz A representa el número de horas de la máquina i que se requiere para producir una unidad del producto j . *En el producto $A^T w$ el primer renglón es:

El término $a_{11} w_1$ representa el costo en la máquina 1 para producir una unidad del producto 1, $a_{21} w_2$ el costo de la máquina 2, para producir una unidad del producto 1. *Por lo tanto el producto representa el costo total de producción de una unidad de cada uno de los productos. Como * c es el costo de venta de una unidad de cada uno de los productos, la desigualdad $A^T w \geq c$ puede interpretarse de la siguiente manera: El costo de producción unitario $A^T w$ es por lo menos tan grande como el beneficio c . Es posible extender esta interpretación del problema dual, para lo cual es necesario introducir teoremas que están fuera del alcance de esta obra**

* x \equiv cantidad de artículos producidos.

* b \equiv disponibilidad de máquinas.

* c \equiv costo de los artículos.

*Teorema de la dualidad.

$$c^T x = b^T w$$

* $c^T x$ \equiv beneficio total.

* w \equiv costo de operación de las máquinas por unidad de tiempo

$$b^T w = \begin{bmatrix} 200, & 125, & 100 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \\ = 200w_1 + 125w_2 + 100w_3$$

* w \equiv precios sombra.

$$A^T w \geq c$$

* $A = [a_{ij}]$ a_{ij} \equiv horas de máquina i para producir una unidad del producto j .

*1er. renglón del $A^T w$

$$A^T w_1 = a_{11} w_1 + a_{21} w_2 + \dots$$

* $A^T w$ \equiv costo total de producción por unidad de cada artículo.

* c \equiv costo de los artículos.

* $A^T w \geq c \rightarrow$ Costo de producción unitario no debe exceder beneficio.

344 Optimización

Como con frecuencia es más fácil resolver el problema dual que el primo o viceversa, resulta conveniente conocer ambos métodos.

Con el siguiente comentario finalizará esta sección sobre programación lineal.

Si en el ejemplo 6.5.2 los datos fuesen diferentes, de manera que el sistema de ecuaciones del problema de programación lineal hubiese sido:

$$\begin{aligned} m &= x_1 + 2x_2 \\ x_1 + 4x_2 &\leq 12 \\ x_1 + x_2 &\leq 4.5 \\ 3x_1 + x_2 &\leq 10.5 \end{aligned}$$

*La solución óptima hubiese sido $x_1 = 2$, y $x_2 = 2.5$ como el lector podrá checar fácilmente (ver problema 6.8.11). La solución de este problema para ser relevante debe ser entera. Cuando como en este caso, la solución debe ser entera, puede recurrirse si las variables son suficientemente grandes y el resultado no es sensible a errores de aproximación a redondear el resultado a la cifra más próxima, ó puede recurrirse a la programación entera. (ref. 3)

*Solución óptima:
 $x_1 = 2, x_2 = 2.5$

En la sección 6.8 el lector puede encontrar diversos problemas de programación lineal (problemas 6.8. 10-6.8.15) y en el apéndice A-17, encuentra un programa de computadora para resolver este tipo de modelos.

El problema del ejemplo 6.5.2 ha sido resuelto empleando el programa A.17.

Los datos de este problema aparecen en la tabla 6.5.5 y los resultados en la 6.5.6. En esta tabla las variables que no aparecen tienen un valor nulo.

Tabla 6.5.5 Datos para el programa A.17

EL ORDEN DE LA MATRIZ DE COEFICIENTES ES 3 X 5 LA MATRIZ DE COEFICIENTES DEL SISTEMA ES:

1.00000E+00	4.00000E+00	1.00000E+00	0.	0.
1.00000E+00	1.00000E+00	0.	1.00000E+00	0.
3.00000E+00	1.00000E+00	0.	0.	1.00000E+00

EL VECTOR DE CONSTANTES INDEPENDIENTES ES

$$\begin{aligned} &2.40000E+01 \\ &9.00000E+00 \\ &2.10000E+01 \end{aligned}$$

**Ver capítulo 6 (ref. 10) y capítulos 3 y 4 (ref. 9)

LOS COEFICIENTES DE LA FUNCION A OPTIMIZAR SON

2.00000E+00 4.00000E+00 0. 0. 0.

Tabla 6.5.6 Resultado del programa A.17.

***LOS VALORES QUE OPTIMIZAN LA FUNCION SON

INCOGNITA	VALOR
X2	5.00000000E+00
X1	4.00000000E+00
X5	4.00000000E+00

EL VALOR OPTIMO DE LA FUNCION ES 2.80000000E+01

6.6. PROGRAMACION DINAMICA

6.6.1 Características

En la sección 6.1.1 se señaló que los métodos de optimización pueden clasificarse en *métodos de gradiente* y *métodos de búsqueda*. *En la sección 6.5 se estudió el método de programación lineal que constituye un método de gradiente. En la siguiente sección se establecen las bases de la *programación dinámica*, un método de optimización de búsqueda. Este último método, todavía más que el de programación lineal requiere del uso de la computadora digital. *Como se trata de una técnica enumerativa, los tiempos de cómputo para este método son en general grandes, así como los requerimientos de memoria. Debido a ello el empleo de esta técnica es un cuanto limitado, a pesar de su extensivo número de aplicaciones potenciales.

*La programación dinámica es una técnica de optimización enumerativa aplicable a problemas con restricciones y funciones objetivo que pueden ser *no lineales* y regiones factibles *no convexas*.

*Se aplica en forma natural a problemas que pueden descomponerse en etapas a lo largo del tiempo, pero también puede emplearse en problemas *no secuenciales* o con estructura en serie.

En el análisis de sistemas, la programación dinámica se usa en general en problemas de toma de decisiones, frecuentemente relacionados con la asignación de recursos.

*Para resolver este tipo de problemas, se establece un modelo matemático cuyas principales componentes son:

*Métodos de optimización de gradiente y búsqueda.

* La programación dinámica es un método de búsqueda.

*Requiere mucho tiempo de cómputo y memoria.

*Puede aplicarse a problemas no lineales y regiones no convexas.

*El problema debe poder expresarse en forma secuencial.

*Modelo matemático.

- 1). Un estado inicial x que da toda la información relevante sobre el sistema antes de la toma de una decisión.

Como el problema de *decisiones* se presenta en aquellas situaciones, donde un problema tiene varias soluciones factibles o alternativas, con objeto de poder seleccionar entre éstas, es necesario asociar a todas las posibles soluciones una función de beneficio o ganancia, que mida la utilidad que se asocia a cada una de las posibles soluciones.

Esta función o relación de transformación puede ser una relación matemática o puede estar dada en forma tabular.

Para representar estas componentes del modelo de toma de decisiones resulta útil introducir un diagrama de bloque (figura 6.6.1).

Como la función de transformación T es univaluada puede sustituirse (6.6.2) en (6.6.1) para obtener:

*Es decir, la función de beneficio r sólo depende de los estados iniciales y las variables de decisión.

Recordando que la función de transformación es univaluada puede obtenerse la transformación inversa T' , a saber

Sustituyendo este valor en (6.6.1) se llega a:

o bien

Un problema de toma de decisiones consiste en maximizar o minimizar la función de beneficio r , si las variables independien-

- 2). Un estado final, \tilde{x} que da toda la información relevante sobre el sistema después de haberse tomado la decisión.
- 3). La variable de decisión $D = (d_1, d_2, \dots, d_n)$ que puede manipularse para obtener determinado cambio del sistema de su estado inicial x , a su estado final \tilde{x} .

- 4). El beneficio r que es una función escalar que depende del valor de los estados iniciales, de las decisiones tomadas, y de los estados finales, es decir

$$r = r(x, D, \tilde{x})$$

- 5). Una transformación T , univaluada que relaciona los estados finales, con los estados iniciales, y las variables de decisión.

$$x = T(x, D) \tag{6.6.2}$$

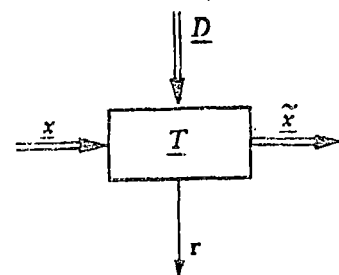


Fig. 6.6.1 Modelo de un problema de toma de decisión.

$$r = r(x, D, T(x, D))$$

*Función de beneficio

$$r = r'(x, D) \tag{6.6.3}$$

$$x = T'(x, D)$$

$$r = r(T'(x, D), D, x)$$

$$r = r''(x, D) \tag{6.6.4}$$

*Maximizar o minimizar el beneficio.

tes o de decisión toman todos los posibles valores, dentro de las restricciones que fija el problema.

Estos problemas de toma de decisiones son, por lo tanto, problemas de optimización entre los que podemos distinguir dos tipos:

El problema de optimización de estado inicial x consiste en encontrar el máximo (o mínimo) del beneficio como función del estado inicial, es decir:

En el problema de estado final \bar{x} , debe determinarse el máximo (o mínimo) del beneficio como función del estado final, es decir:

Con objeto de facilitar la presentación del material subsecuente e ilustrar la naturaleza de estos problemas, conviene introducir algunos símbolos:

°Optimización de estado inicial x

$$f(x) = \max_D r(x, D) \quad (6.6.5)$$

(6.6.5)

°Optimización de estado final \bar{x}

$$f(\bar{x}) = \max_D r(\bar{x}, D) \quad (6.6.6)$$

°Símbolos empleados en programación dinámica.

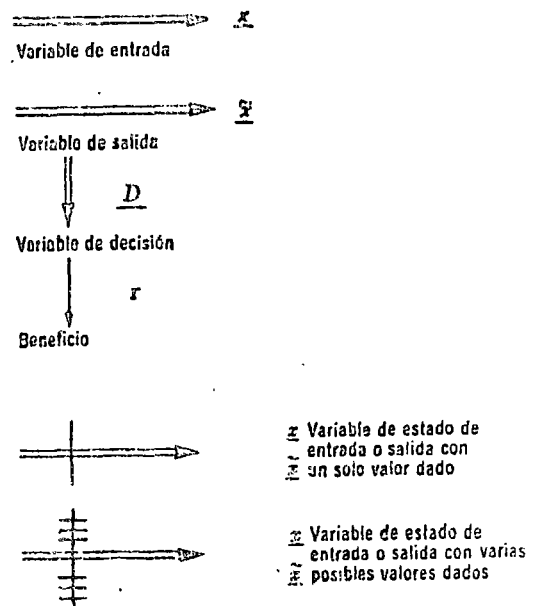


Fig. 6.6.2 Símbolos en problemas de programación dinámica.

Usando estos símbolos el problema de valor inicial puede simbolizarse:

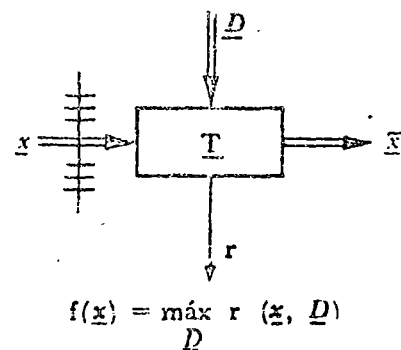


Fig. 6.6.3 Problema de valor inicial.

y el de valor final

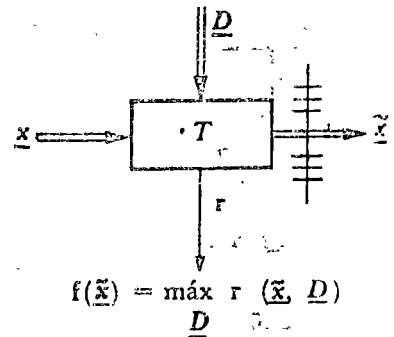


Fig. 6.6.4 Problema de valor final.

*Problemas de optimización como los planteados en las figuras (6.6.3) y (6.6.4) contienen muchas variables. La programación dinámica transforma un problema de esta naturaleza en una serie de problemas más sencillos, que contienen pocas variables.

Esta transformación es invariante en el número de soluciones factibles del problema y se conserva el valor de la función beneficio asociada a cada una de las posibles soluciones.

*La programación dinámica se basa en el principio de optimalidad expuesto por R.D. Bellman: (ref. 2).

Un ejemplo adaptado de la ref. 8 servirá para aclarar este concepto, en que se basa la programación dinámica.

*Supóngase que se desea asignar recursos a tres proyectos industriales, A, B y C con el objeto de maximizar las ganancias, *sean R_A , R_B y R_C las cantidades que se asignan a los proyectos A, B y C respectivamente y sean R_T los recursos totales disponibles que son limitados. Debido a ello, la cantidad que se asigna a cada proyecto, depende de la cantidad asignada a los dos restantes. La asignación a C no debe exceder $R_T - R_A - R_B$ *Sin embargo, cualquiera que haya sido la asignación a los proyectos A y B, la asignación R_C al proyecto C, debe ser óptima con respecto a todas las posibles cantidades residuales que pueden quedar para el proyecto C, después de asignar fondos a los proyectos A y B. *La asignación de fondos a los proyectos B y C debe ser óptima con respecto a la cantidad residual que queda después de asignar recursos a A, cualesquiera que haya sido esta asignación.

La asignación óptima al proyecto B, se encuentra maximizando el beneficio, que ocurre de la asignación al proyecto B, junto con el

*Programación dinámica:
 Un problema con muchas variables. \Rightarrow
 Muchos problemas de pocas variables.

*Principio de optimalidad de Bellman.

“Una serie de decisiones óptimas (políticas óptimas) tiene la propiedad, de que cualquiera que sea el estado inicial y la decisión inicial, las decisiones restantes deben ser óptimas con respecto al estado que resulte de la primera decisión”.

* Proyectos industriales A, B, C.

* R_A, B, C recursos para cada proyecto
 R_T recursos totales disponibles.

$$*R_A + R_B + R_C \leq R_T$$

*La asignación a C debe ser óptima con respecto a $R_T - R_A - R_B$.

*La asignación a B y C debe ser óptima con respecto a $R_T - R_A$.

beneficio óptimo del proyecto C, como función de los fondos que quedan de asignar recursos a B y A. La asignación óptima a A finalmente se encuentra para maximizar el beneficio de A más el beneficio óptimo de B y C, como función de los fondos que quedan después de asignar recursos a A.

Obsérvese que se ha descompuesto el problema, en una secuencia de toma de decisiones, asignando recursos a un solo proyecto a la vez.

En realidad la asignación de recursos es simultánea, pero la descomposición del problema, en una asignación secuencial o en serie de los recursos, permite tomar decisiones una a la vez.

El concepto de sistema secuencial o en serie es muy importante en este tipo de problemas y se discute con mayor detalle en la siguiente sección.

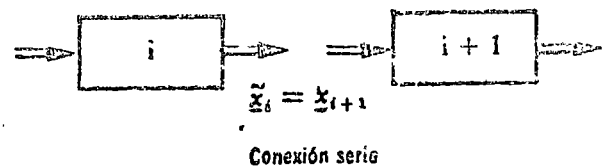
6.6.2 Estructuras serie

*En una estructura en serie, como se señaló en la sección 1.3.4, la salida de un elemento está conectada a la entrada del siguiente, sin haber realimentación, ésta, como se indicó en la sección 1.3.5, implica que la salida de un sistema influye sobre su entrada. La presencia de realimentación en un problema de programación dinámica puede resolverse sustituyendo la porción del sistema con realimentación por un subsistema equivalente no realimentado. Los ingenieros llaman a esta operación: sustituir el sistema realimentado por su función de transferencia.**

*En un problema con estructura serie en el tiempo, que son los más frecuentes en el análisis de sistemas, las decisiones que se toman en un determinado instante de tiempo, no alteran los eventos anteriores, sólo tienen influencia sobre los eventos posteriores.

En la construcción de una casa, el levantamiento de muros, es posterior a la construcción de los cimientos pero anterior a la colocación de ventanas y puertas. Si durante la construcción de los muros, se cambia la posición y tamaño de los huecos para las puertas y las ventanas, este cambio, resultado de una decisión, no afecta a la etapa anterior, o sea la construcción de los cimientos, pero sí influye sobre la etapa posterior, la de colocación de puertas y ventanas.

*Se asignan recursos a un proyecto a la vez.



*En una estructura serie las decisiones no afectan eventos anteriores.

**Gerez Greiser V. y Murray-Lasso, M. A. Teoría de Sistemas y Circuitos 1, Cap. 8. Servicios y Representaciones de Ingeniería, S. A. México, D. F., 1972.

Esquemáticamente un problema con estructura en serie, puede representarse usando los diagramas de bloque de la sección 1.3.4, de la forma mostrada en la figura 6.6.5.

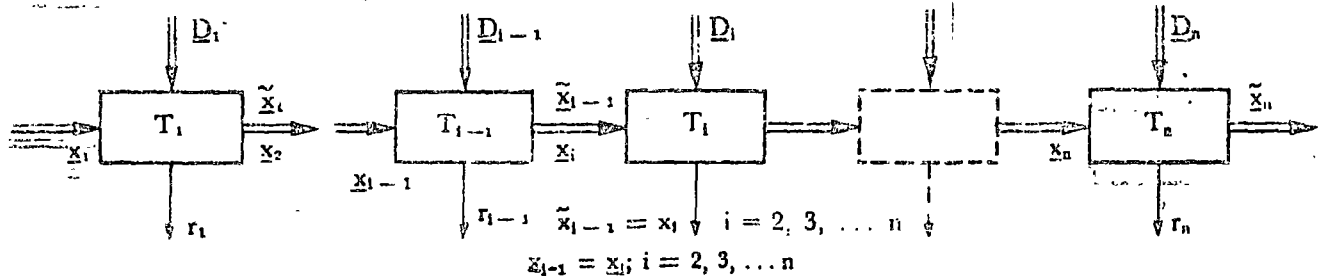


Fig. 6.6.5 Estructura en serie.

A continuación se hace una presentación formal del principio de optimalidad y se deduce la fórmula recursiva para resolver este tipo de problemas.

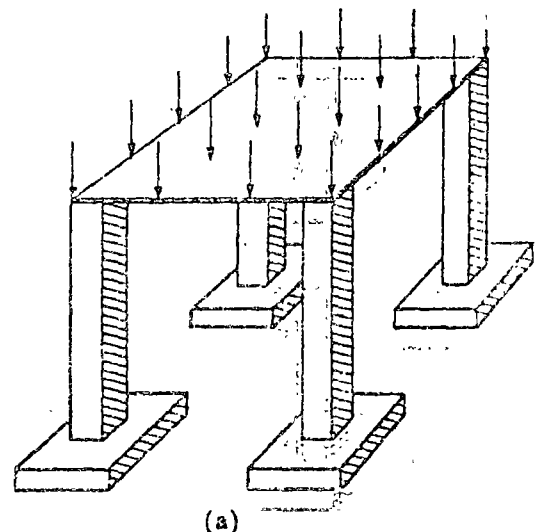
6.6.3 Principio de optimalidad

*Se señaló en la sección anterior que el objetivo de la descomposición del problema de optimización en una serie de problemas secuenciales, es reducir el número de variables que se manipulan en cada etapa, trabajando, de preferencia, con una variable de estado y una variable de decisión. Por esta razón en los desarrollos subsecuentes se emplean los símbolos que corresponden a cantidades escalares, como por ejemplo x , y no los correspondientes a vectores como \tilde{x} , tampoco se seguirá empleando el trazo doble para representar las variables en los diagramas de bloque.

*Trabajar de preferencia con una variable de estado y una de decisión.

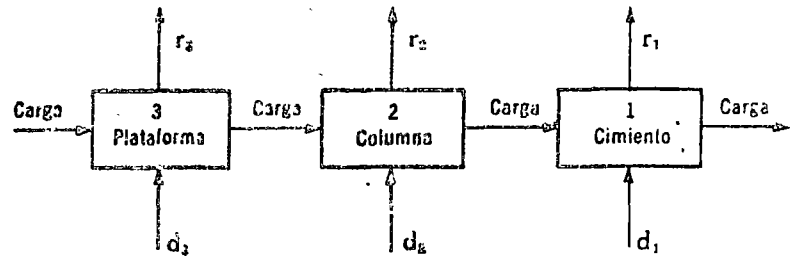
A continuación aplicaremos el principio de optimalidad a un problema de valor inicial adaptado de la ref. (1)

La figura 6.6.6a muestra una plataforma que debe soportar una carga dada de $\omega \text{ kg/m}^2$. El objetivo del problema es diseñar una plataforma, las columnas de soporte y los cimientos necesarios para soportar el peso minimizando el costo de la obra. Para aplicar la técnica de la programación dinámica a este problema, conviene descomponerlo en una serie de problemas más fáciles de optimizar.



(a) Plataforma para soportar $\omega \text{ Kg/m}^2$

La solución de este problema puede esquematizarse como muestra la fig. 6.6.6 b



(b)

Estructura secuencial para la solución del problema de diseño de una plataforma de carga

Fig. 6.6.6 Ejemplo de aplicación del método de programación dinámica.

Supóngase que se empieza analizando las columnas; si se encuentra que la solución más económica son las columnas de concreto, esta solución implica mayor peso sobre los cimientos que el producido por las columnas de hierro. Esta solución afecta el beneficio (costo) de todas las etapas subsecuentes (En este caso los cimientos). Por lo tanto no puede empezarse analizando las columnas.

*Resulta evidente que la estrategia adecuada de solución consiste en empezar analizando aquella parte del proyecto, que no influye sobre los restantes, en este caso los cimientos. Al igual que en la asignación de recursos a tres proyectos industriales en la sección 6.6.1, posteriormente pueden agruparse las dos últimas etapas, columnas y cimientos, para suboptimizarse posteriormente, sin afectar a ninguna otra etapa.

*Empiece por aquellas partes que no afectan otras etapas.

Como se ve, el proceso de optimización se realiza en orden inverso, primero se estudian los cimientos, después los cimientos en combinación con las columnas y finalmente todo el proyecto. Conviene por lo tanto numerar los pasos de solución en este orden, tal como aparece en la fig. 6.6.6 o en general como se muestra en la figura 6.6.7.

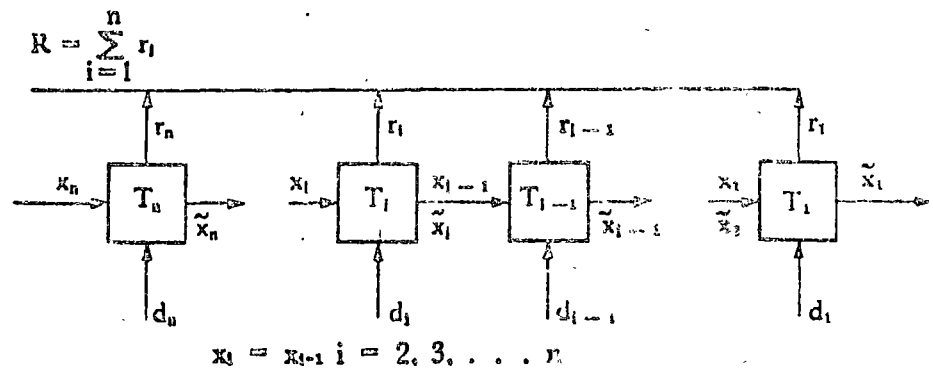


Fig. 6.6.7 Estructura secuencial de n pasos.

352 Optimización

*Recuérdese que el beneficio en un problema de valor inicial puede expresarse como función del estado inicial x_1 y de la variable de decisión d_1 (ecs. 6.6.4)

*Si la función beneficio R para todo el problema, es la suma de los beneficios de cada una de las etapas, se tiene:

*recordando la estructura serie del problema que implica

*y la relación entre la variable de entrada x_i , la de salida x_{i+1} y la de decisión d_i

*se obtiene para la primera etapa de la serie

Por ser la entrada al primero x_1 , igual a la salida del segundo \bar{x}_2 , se tiene:

pero

sustituyendo esta relación en la anterior

y como

y

se tiene al sustituir en (6.6.10)

Siguiendo con esta sustitución se obtiene:

$$r_1 = r_1 [T_2 (T_3 [T_4 \dots \{T_n (x_n, d_n), d_{n-1}\}, \dots] d_2) d_1] \quad (6.6.11)$$

*Obsérvese que esta relación indica que el beneficio r_1 asociado a la etapa 1 es función solamente de la variable de estado inicial y de todas las variables de decisión. Una conclusión idéntica se puede obtener para todas las etapas subsecuentes, por lo tanto el beneficio total del proyecto es función exclusiva del estado inicial y de todas las variables de decisión, es decir,

*El problema de optimización consiste en encontrar los valores de las variables de decisión d_1, d_2, \dots, d_n que para un valor dado x_n del estado inicial maximicen o minimicen la función de beneficio R de todo el proyecto.

*Beneficio de la etapa i 'sima:

$$r_i = r_i (x_i, d_i) \quad (6.6.7)$$

*Para beneficios aditivos:

$$R = \sum_{i=1}^n r_i (x_i, d_i) \quad (6.6.8)$$

*Como la estructura es serie:

$$x_i = x_{i-1} \quad i = 2, 3, \dots, n \quad (6.6.9)$$

*relación entrada — salida

$$x_i = T_i (x_i, d_i) \quad (6.6.2)$$

*1ra. etapa

$$r_1 = r_1 (x_1, d_1)$$

$$x_1 = \bar{x}_2$$

$$r_1 = r_1 (\bar{x}_2, d_1)$$

$$\bar{x}_2 = T_2 (x_2, d_2)$$

$$r_1 = r_1 (T_2 (x_2, d_2), d_1) \quad (6.6.10)$$

$$x_2 = \bar{x}_3$$

$$\bar{x}_3 = T_3 (x_3, d_3)$$

$$r_1 = r_1 (T_2 (T_3 (x_3, d_3), d_2), d_1)$$

*El beneficio total depende del estado inicial y de las variables de decisión.

$$R = R(x_n, d_1, d_2, \dots, d_n) \quad (6.6.12)$$

*Encuentre d_1, d_2, \dots, d_n que optimice el beneficio total R , dado el estado inicial x_n .

Analícese ahora el problema empezando con la 1ra. etapa.

Para esta etapa, sea f_1 el máximo (o mínimo) de la función beneficio.

*Para cada valor posible de x_1 , la función beneficio tiene un valor óptimo, que se encuentra optimizando esta función con relación a la variable de decisión d_1 , es decir

°Beneficio óptimo $f_1(x_1)$ para cada valor de x_1

$$f_1(x_1) = \max_{d_1} r_1(x_1, d_1) \quad (6.6.13)$$

*Si se considera a continuación la segunda etapa su beneficio será:

°Para la 2da. etapa.

$$r_1(x_1, d_1) + r_2(x_2, d_2)$$

*y el óptimo será:

°Valor óptimo

$$\max_{d_1, d_2} \{r_1(x_1, d_1) + r_2(x_2, d_2)\}$$

*El beneficio óptimo de la primera etapa ya ha sido calculado en (6.6.13) y por lo tanto se tiene como beneficio óptimo de la primera y segunda etapas combinadas, por el principio de optimalidad.

°Beneficio para la 1ra. y 2da. etapas.

$$\max_{d_2} \{r_2(d_2, x_2) + f_1(x_1)\} \quad (6.6.14)$$

*Nótese que en esta segunda etapa ya solamente es necesario buscar el óptimo con respecto a d_2 .

°Sólo se busca el óptimo respecto a d_2 .

*Por la conexión serie entre etapas se tiene

°Conexión serie

$$x_1 = \bar{x}_2$$

y por la transformación que ejerce la segunda etapa

$$\bar{x}_2 = T_2(x_2, d_2)$$

Sustituyendo en (6.6.14)

$$\max_{d_2} \{r_2(d_2, x_2) + f_1(T_2(x_2, d_2))\}$$

*El beneficio óptimo de la primera y segunda etapas combinadas es por tanto:

°Beneficio óptimo de la 1ra. y 2da. etapas.

$$f_2(x_2) = \max_{d_2} \{r_2(x_2, d_2) + f_1(T_2(x_2, d_2))\}$$

*Procediendo con este razonamiento se llega a la n'sima y última etapa y se obtiene una relación similar para el beneficio óptimo.

°Para la última etapa.

$$f_n(x_n) = \max_{d_n} \{r_n(x_n, d_n) + f_{n-1}(T_n(x_n, d_n))\} \quad (6.6.15)$$

°Fórmula de recursión.

*Toda esta deducción puede por lo tanto resumirse en las siguientes ecuaciones de recursión para el problema de programación dinámica:

$$f_i(x_i) = \max_{d_i} Q_i(x_i, d_i) \quad i = 1, 2, \dots, n$$

$$Q_1(x_1, d_1) = r_1(x_1, d_1) \quad i = 1 \quad (6.6.16)$$

$$Q_i(x_i, d_i) = r_i(x_i, d_i) + f_{i-1}(T_i(x_i, d_i))$$

$$i = 2, 3, \dots, n$$

El problema siguiente ilustra el empleo de la programación dinámica.

Ejemplo 6.6.1.

*Supóngase que se desea maximizar el beneficio que se obtiene de un programa de desarrollo industrial.

*Maximización del beneficio.

*El proyecto prevé la instalación de un máximo de tres industrias diferentes. El beneficio que se obtiene de cada industria depende del nivel de inversión en las mismas. *Sea x_i el nivel de inversión en la i 'sima industria, y $g_i(x_i)$ el beneficio que se obtiene de la misma, si el nivel de inversión en ella es de x_i . Además se cuenta con un capital máximo de 3 billones de pesos para el Programa. Debido a la naturaleza de cada proyecto de inversión, los niveles de inversión sólo pueden ser múltiplos enteros de 1 billón de pesos. La figura 6.6.8 y la tabla 6.6.1 muestran el beneficio que se obtiene de cada proyecto de acuerdo con el nivel de inversión.

*Tres unidades industriales.

* x_i nivel de inversión en industria i 'sima y $g_i(x_i)$ su beneficio.

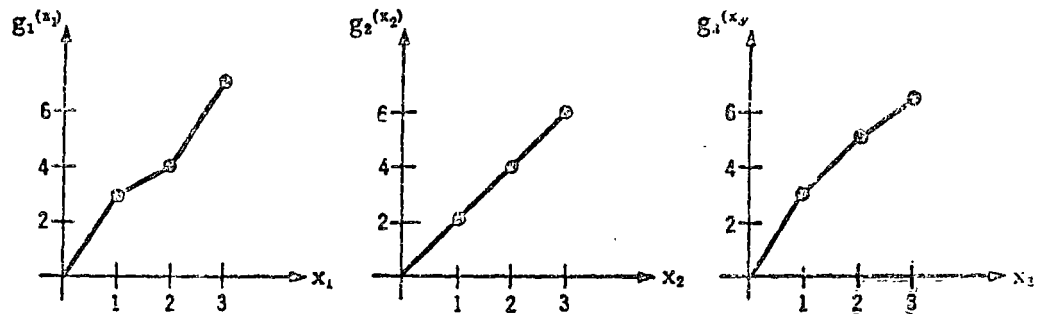


Fig. 6.6.8 Funciones de beneficio del ejemplo 6.6.1.

Tabla 6.6.1 Beneficio de los proyectos del ejemplo 6.6.1.

Función de beneficio	Industria i		
	1	2	3
$g_1(0)$	0	0	0
$g_1(1)$	3	2	3
$g_1(2)$	4	4	5
$g_1(3)$	7	6	6

Solución.

*Debido a la naturaleza del proyecto, la función objetivo o beneficio total que se obtiene de este proyecto es de carácter aditivo, es decir:

*Además, se tiene la restricción en los fondos de:

*Como el orden de asignación de recursos en este caso es irrelevante puede establecerse cualquier secuencia en la serie. Si empleamos la del enunciado se tiene el diagrama de bloque de la figura 6.6.9

*Función de beneficio total aditiva.

$$R = \sum_{i=1}^3 g_i(d_i)$$

*Restricción de fondos

$$3 \geq x_1 + x_2 + x_3$$

*La secuencia de asignación es irrelevante.

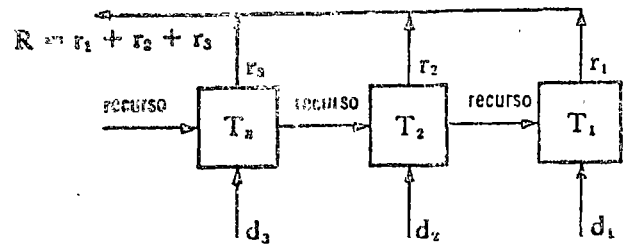


Fig. 6.6.9 Diagrama de bloque del ejemplo 6.6.1.

Como variable de entrada a cada proyecto puede considerarse el recurso que queda por asignarse, después de asignados recursos a los anteriores, y como salida lo que queda por asignar, una vez asignados fondos al mismo. La entrada, al tercero es fijo e igual a 3. Si se toma la decisión de asignar dos billones de pesos a este proyecto, es decir, $d_3 = 2$, la salida del tercer bloque x_3 será 1, y el beneficio r_3 de acuerdo con la tabla 6.6.1 serie de 4 tal como lo ilustra la figura 6.6.10

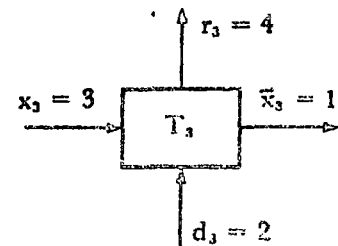
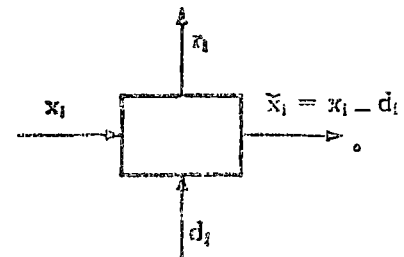


Fig. 6.6.10 Ejemplo de asignación de recursos al proyecto 3.

En este ejemplo, la transformación tiene esta forma simple $\bar{x}_1 = x_1 - d_1$ y las asignaciones de recursos están sometidas a la limitación



$$+ d_1 + d_2 + d_3 \leq 3$$

$$x_1 - d_1 \geq 0 \quad \& \quad r_1 \geq d_1$$

Como la variable que entra a cada bloque es el recurso disponible, se debe tener además que es decir, no se puede gastar en un proyecto más de los recursos disponibles.

356 Optimización

*La función de beneficio $r_1(x_1, d_1)$ en este caso solamente depende de la decisión que se tome, es decir:

*La fórmula de recursión para la solución del problema es

En este caso la transformación es:

Sustituyendo en la relación (6.6.17) se obtiene

*Recordando que para $i = 1$ la función óptima de beneficio es:

Con la importante restricción señalada de que

Puede establecerse por lo tanto la tabla 6.6.2 para el cálculo de la función de beneficio óptima del 1er. proyecto.

*Función de beneficio

$$r_1(x_1, d_1) = g_1(d_1)$$

*Fórmula de recursión

$$Q_i(x_i, d_i) = r_i(x_i, d_i) + f_{i-1}(T_{i-1}, d_i) \quad (6.6.17)$$

$$T_i(x_i, d_i) = x_i - d_i$$

$$Q_1(x_1, d_1) = g_1(d_1) + f_{1-1}(x_1 - d_1) \quad (6.6.18)$$

*Para el 1er. proyecto.

$$f_1(x_1) = \max_{d_1} g_1(d_1)$$

$$x_1 \geq d_1$$

Tabla 6.6.2 Asignación de recursos a la etapa 1.

Valor de x_1	Posibles valores de d_1 $d_1 \leq x_1$	Beneficio $g_1(d_1)$	Beneficio óptimo $f_1(x_1)$	Valor de d_1^* que produce el óp.
0	0	0	0	0
1	0 1	0 3	3	1
2	0 1 2	0 3 4	4	2
3	0 1 2 3	0 3 4 7	7	3

*Para la segunda etapa la fórmula de recursión establece:

Este máximo también tiene que encontrarse para todos los valores posibles de x_2 . La tabla 6.6.3 ilustra cómo se obtiene esta serie de máximos para los diversos valores de x_2 . *Nótese además que tanto en la tabla anterior como en ésta, se anotan los valores de las variables de decisión que llevan al beneficio óptimo.

°para la 2da. etapa

$$f_2(x_2) = \max_{d_2} \{g_2(d_2) + f_1(x_2 - d_2)\}$$

°Anote el valor de las variables de decisión "óptimas".

Finalmente para la etapa 3 se tiene

$$f_3(x_3) = \max_{d_3} \{g_3(d_3) + f_2(x_3 - d_3)\}$$

En la tabla 6.6.4 se resumen los valores de esta etapa.

Tabla 6.6.3 Asignación de recursos a la etapa 2.

Valor de x_2	Posibles Vals. de d_2 $d_2 \leq x_2$	Beneficio de la etapa $g_2(d_2)$	Diferencia $x_2 - d_2$	Beneficio ópt. de la etps. ants. $f_2(x_2 - d_2)$ (Tabla 6.6.2)	Valor d_1^* que prod. $f_1(x_2 - d_2)$	Beneficio acumulado $Q_2(x_2, d_2)$	Beneficio óptimo $f_2(x_2)$	Val. de var. de decs. que prod. el ópt.	
								d_1^*	d_2^*
0	0	0	0	0	0	0	0	0	0
1	0	0	1	3	1	3	3	1	0
	1	2	0	0	0	2			
2	0	0	2	4	2	4	5	1	1
	1	2	1	3	1	5			
	2	4	0	0	0	4			
3	0	0	3	7	3	7	7	3	0
	1	2	2	4	2	6			
	2	4	1	3	1	7			
	3	6	0	0	0	6			

Tabla 6.6.4 Asignación de recursos a la etapa 3.

Valor de x_3	Posibles valores de d_3 $d_3 \leq x_3$	Beneficio de la etapa $g_3(d_3)$	Diferencia $x_3 - d_3$	Beneficio opt. de las etps. ants. $f_2(x_3 - d_3)$ (Tabla 6.6.3)	Valores d_1^* y d_2^* que prod. $f_2(x_3 - d_3)$		Beneficio acumulado $Q_3(x_3, d_3)$	Beneficio óptimo $f_3(x_3)$	Valores variables d_1^* , d_2^* y d_3^* que prod. el beneficio óptimo		
					d_1^*	d_2^*			d_1^*	d_2^*	d_3^*
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	3	1	0	3	3	1	0	0
	1	3	0	0	0	0	3		0	0	1
2	0	0	2	5	1	1	5	6	1	0	1
	1	3	1	3	1	0	6				
	2	5	0	0	0	0	5				
3	0	0	3	7	1	0	7	8	1	1	1
	1	3	2	5	1	1	8				
	2	5	1	3	1	0	8				
	3	6	0	0	0	0	6				

*Esta última tabla 6.6.4 permite concluir que el beneficio óptimo que se obtiene dentro de los límites de los recursos disponibles $x_3 \leq 3$ es de 8. El beneficio de 8 se obtiene asignando recursos de las dos maneras que muestra la figura 6.6.11.

*Beneficio óptimo.

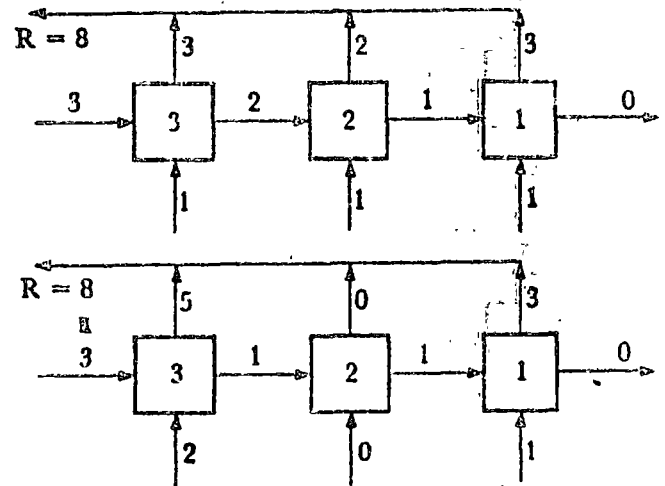


Fig. 6.6.11 Asignación óptima de recursos al proyecto del ejemplo 6.6.1

Obsérvese que en este caso existen dos estrategias de asignación de recursos que llevan al mismo beneficio de 8, dentro de la limitación $x_3 \leq 3$ ó $d_1 + d_2 + d_3 \leq 3$. La tabla 6.6.5 resume los resultados de este problema.

Tabla 6.6.5 Estrategias óptimas de inversión en el proyecto del ejemplo 6.6.1.

Proyecto	Asignación de recursos	Beneficio	
1	1	3	
	1		3
2	1	2	
	0		0
3	1	3	
	2		5
Beneficio total			

*Para aclarar la razón por la cual la programación dinámica es una técnica enumerativa y por la cual el principio de optimalidad reduce el número de alternativas entre las que hay que buscar el máximo, se procede a continuación a ilustrar la solución de este problema empleando árboles de decisiones, como los empleados en la sección 1.3.9.

*El principio de optimalidad reduce el número de alternativas a explorar.

Empezando asignando recursos al proyecto 1, se tienen las alter-

nativas mostradas en la figura 6.6.12. La cantidad dentro de los nodos indica el beneficio que se ha obtenido siguiendo las asignaciones de recursos asociadas a los segmentos de recta del nodo en cuestión hasta el origen del diagrama. El símbolo $g_1(d_1)$ representa el beneficio que se obtiene al asignar d_1 recursos al proyecto i

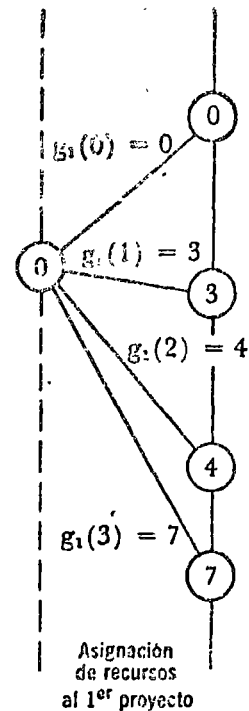


Fig. 6.6.12 Árbol de combinaciones para la asignación de 3 unidades al 1er. proyecto del ejemplo 6.6.1.

La asignación de recursos al segundo proyecto, depende de la que ya se asignó al primero. Si por ejemplo al 1er. proyecto se le asigna 1 unidad y se obtiene un beneficio de 3, al segundo proyecto solamente pueden asignársele 0, 1 ó 2 unidades sin excederse de los recursos totales de 3. Los beneficios totales que se obtienen después de estas posibles asignaciones al segundo proyecto aparecen en la figura 6.6.13

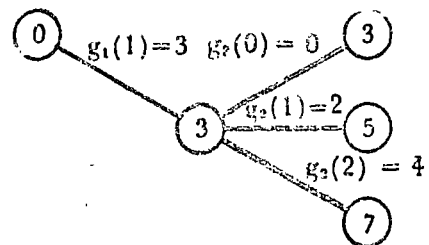


Fig. 6.6.13 Árbol con algunas posibles asignaciones de recursos al 2do. proyecto.

Siguiendo con el método expuesto, se puede construir el árbol de asignación de recursos para todo el proyecto. Este árbol se muestra en la figura 6.6.14.

Este árbol muestra de inmediato las dos estrategias óptimas que aparecen en la figura 6.6.15

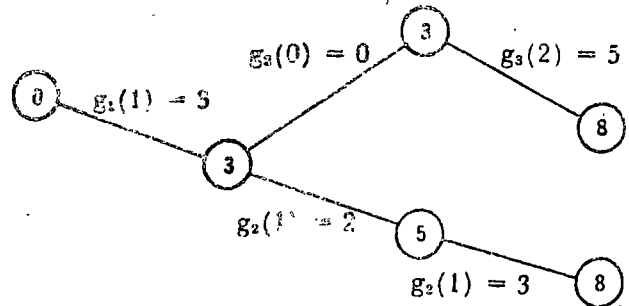


Fig. 6.6.15 Asignación óptima de recursos al proyecto del ejemplo 6.6.1.

El árbol de decisiones de la figura 6.6.14 enumera todas las posibles alternativas del proyecto, y constituye un método de fuerza bruta. *A continuación se señala cómo la programación dinámica refina este método reduciendo el número de alternativas entre las que se tiene que buscar el máximo.

*La programación dinámica reduce las alternativas entre las que se busca el óptimo.

*Recuérdese que el proceso empieza en la primera etapa señalando que la función de beneficio es:

*Función de beneficio para la 1ra. etapa:

$$f_1(x_1) = \max_{d_1} g_1(d_1)$$

*y para la segunda etapa se tiene:

*Para la 2da. etapa

$$f_2(x_2) = \max_{d_2} \{g_2(d_2) + f_1(x_2 - d_2)\}$$

Esta fórmula indica que no es necesario buscar el óptimo beneficio que se obtiene al asignar recursos a los proyectos 1 y 2 buscando entre todos los posibles valores de los beneficios de las etapas 1 y 2, sino solamente entre las posibles combinaciones de beneficios de dos con beneficios óptimos de la primera etapa.

Finalmente para la última etapa se tiene:

$$f_3(x_3) = \max_{d_3} \{g_3(d_3) + f_2(x_3 - d_3)\}$$

*Igualmente el beneficio óptimo no se busca entre las posibles combinaciones de beneficios de la primera, segunda y tercera etapas, sino simplemente entre las combinaciones de beneficios de la última etapa y del óptimo de las dos anteriores. Esta estrategia de búsqueda, resultado del principio de optimalidad, reduce el número de alternativas entre las que hay que buscar el óptimo. Las figuras 6.6.16 a, b, c, ilustran cómo se eliminan alternativas de acuerdo con la descripción anterior.

*Se busca entre los beneficios de una etapa y el óptimo de la combinación de las anteriores.

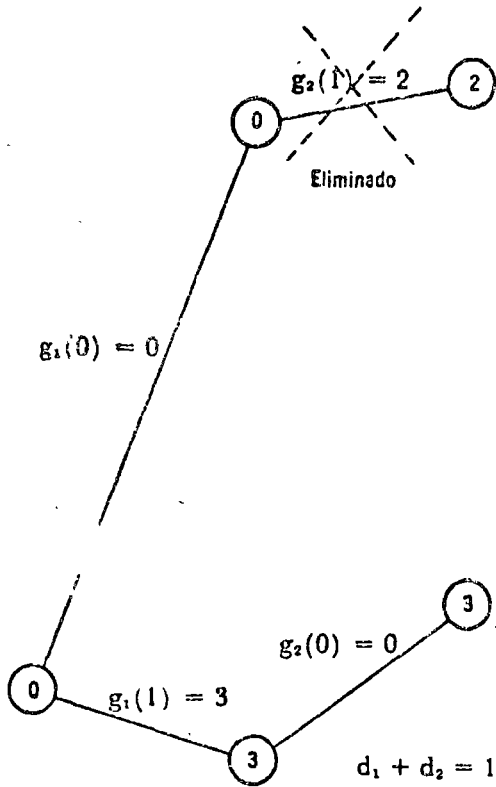


Fig. 6.6.16 a Asignación de una unidad de recurso en 2 etapas.

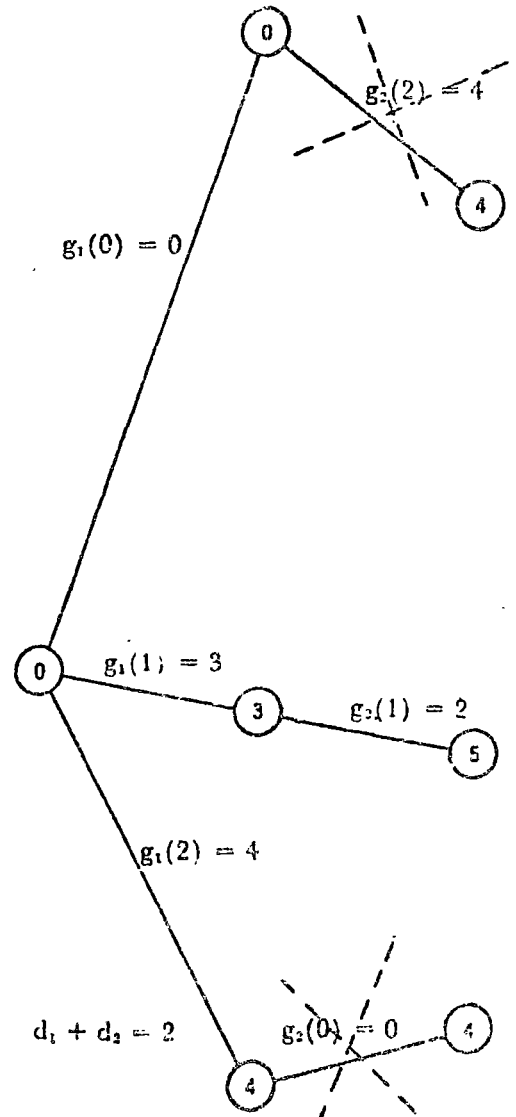
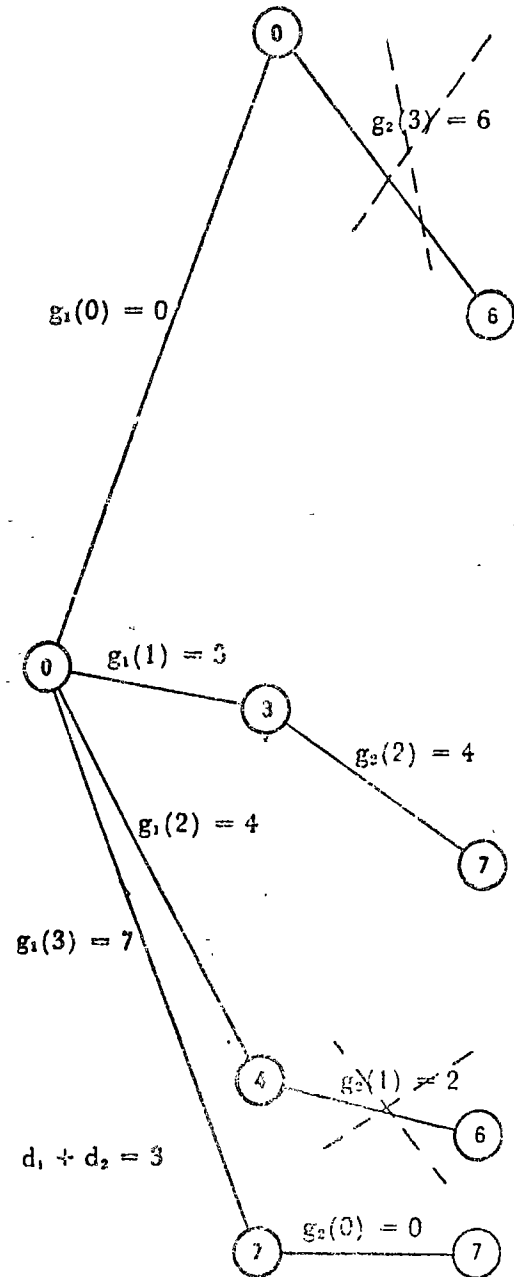


Fig. 6.6.16 b Asignación de 2 unidades de recurso en dos etapas.



La eliminación de estas alternativas reduce la búsqueda a los casos que muestra el árbol de la figura 6.6.17 con trazo grueso.

Fig. 6.6.16 a Asignación de 5 unidades de recurso en 3 etapas.

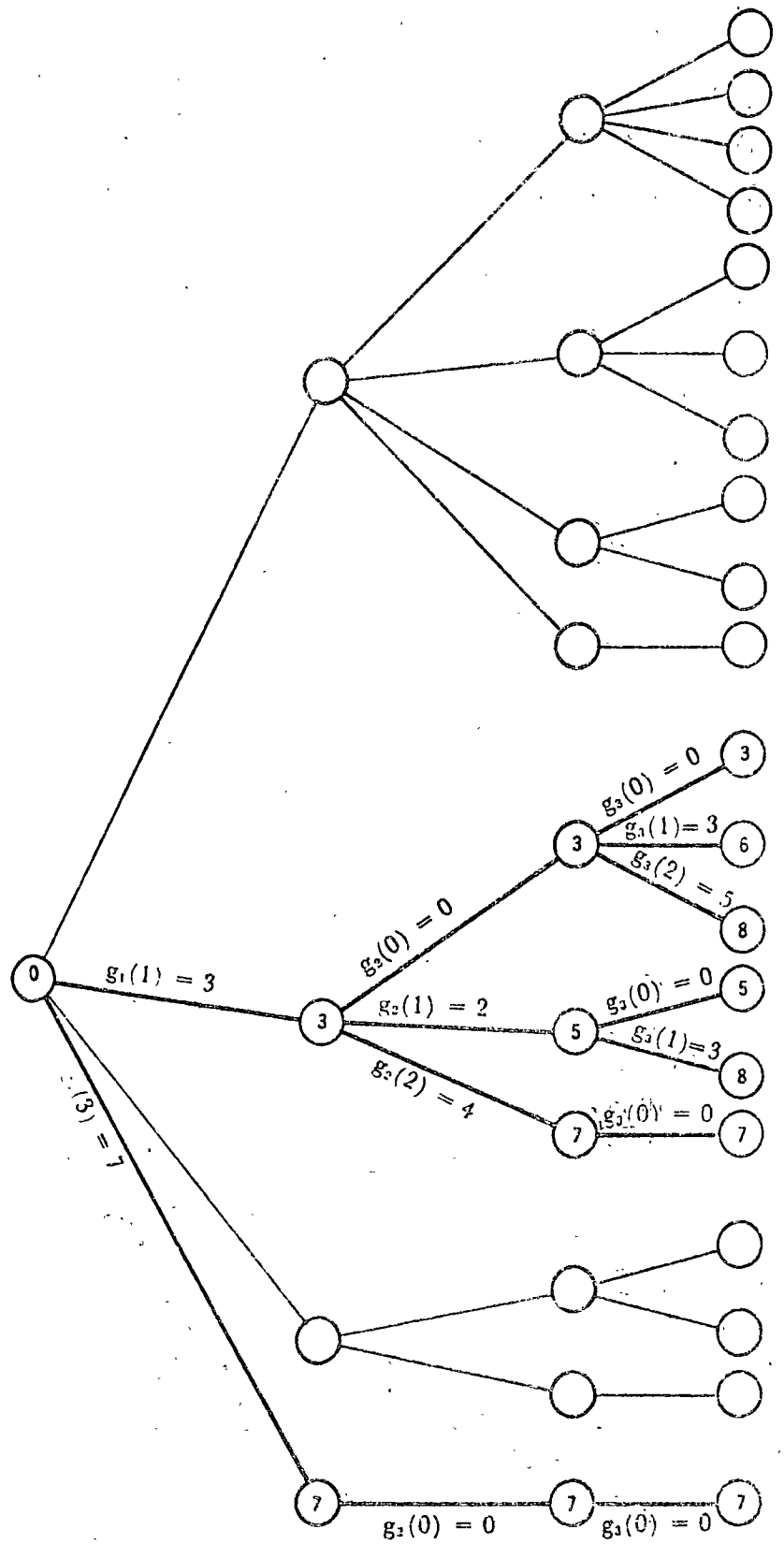


Fig. 6.6.17 Reducción de alternativas a explorar.

*La figura 6.6.14 muestra que este problema tiene 20 posibles alternativas. Si se emplea una búsqueda directa es necesario buscar entre estas posibles alternativas, para las cuales debe conocerse la combinación de decisiones que llevan a cada una de ellas, como ilustra la figura 6.6.18 para una de ellas.

*Búsqueda directa:
20 alternativas
Programación dinámica:
8 alternativas

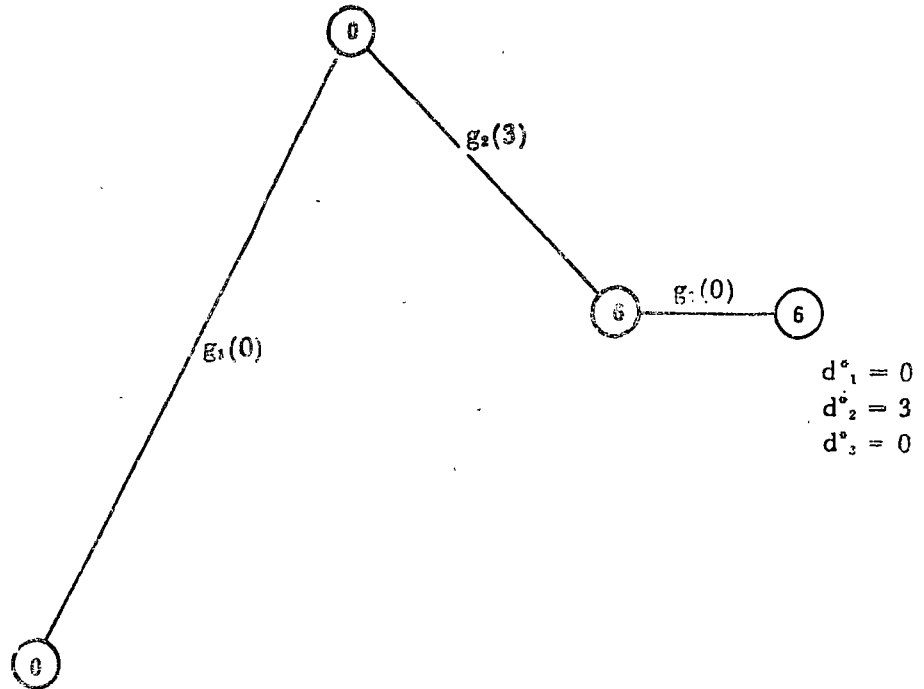


Fig. 6.6.18 Secuencia de decisiones que llevan a un beneficio determinado.

Como estos problemas tienen en general muchas más alternativas que las que se presentan en este ejemplo y más etapas de decisión, el método enumerativo directo requeriría de una gran cantidad de operaciones y de conservar en la memoria una gran cantidad de información: todas las posibles secuencias de la variable de decisión entre otros datos. La programación dinámica, al reducir el número de alternativas entre las que hay que buscar el óptimo, disminuye los tiempos de computación y los requerimientos de memoria. A pesar de ello, uno de los factores que ha limitado la aplicación de este método es precisamente el requerimiento de memoria que se necesita. En el capítulo 11 de la ref. 1 el lector puede encontrar una presentación formal sobre el problema de reducción del esfuerzo computacional entre la búsqueda directa y la programación dinámica.

La solución de un problema de asignación de recursos con un número mayor de etapas que el del ejemplo 6.6.1 puede encontrarse empleando el programa A18 del apéndice A. Este programa requiere de los siguientes datos:

- a) Número de industrias
- b) Monto de la inversión
- c) Funciones de beneficio de cada industria.

El resultado de este programa aparece en la tabla 6.6.6.

Tabla 6.6.6 Resultados del programa A18 para el ejemplo 6.6.1.

LOS RESULTADOS OBTENIDOS SON (LOS VALORES DE LA MATRIZ CORRESPONDEN A LAS INVERSIONES NECESARIAS A EFECTUAR EN CADA INDUSTRIA)

BENEFICIO	INDUSTRIA		
	1	2	3
0	0	0	0
3	1	0	0
3	0	0	1
6	1	0	1
8	1	1	1
8	1	0	2

Antes de continuar debe hacerse notar que en cada etapa de la solución es necesario encontrar un máximo (o mínimo). Para encontrarlo, de acuerdo con el tipo de problema se aplica alguna de las técnicas expuestas en las secciones anteriores de este capítulo, o bien una búsqueda del tipo introducido en las secciones 3.5.2 ó 3.5.3.

6.6.4 Redes de transporte

Una aplicación importante de la programación dinámica es la determinación de rutas más largas o más cortas en redes de transporte entre dos localidades. En esta sección se ilustra este problema.

*La figura 6.6.19 ilustra las posibles rutas entre una localidad V y dos puertos de un litoral. Supóngase que las poblaciones intermedias son de tres tipos, cercanas a la localidad, cercanas al litoral e intermedias, agrupadas como muestra la figura 6.6.19.

Los números asociados a las carreteras indican su longitud. Se trata de obtener la ruta más corta entre la población V y el litoral.

Ejemplo 6.6.2

*Posibles rutas del litoral al interior.

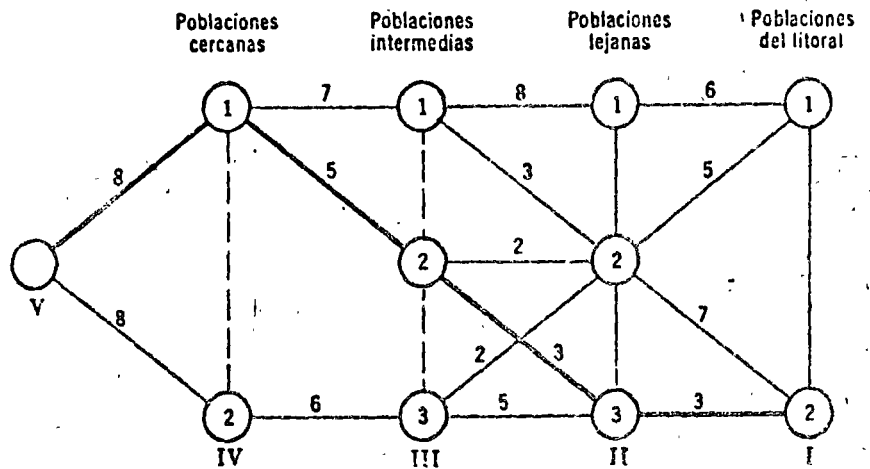
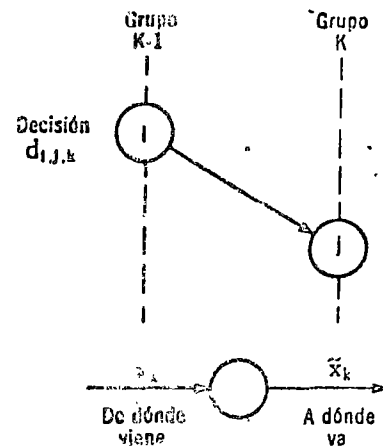


Fig. 6.6.19 Red de caminos entre la localidad V y puertos de un litoral.

*Para resolver todo problema conviene introducir una notación adecuada. Designemos con $d_{i,j,k}$ con la decisión de ir de la población i del grupo $k-1$, a la población j del grupo k . *Cada variable de estado de entrada x_k indica de qué población de la zona anterior viene la carretera, y la variable de salida \tilde{x}_k indica hacia qué población de la zona siguiente va la carretera.

Con esta nomenclatura se puede empezar a resolver el problema.



*Para la 1ra. etapa, o sea la ruta entre el litoral y las poblaciones lejanas se tiene como óptimo de la función objetivo:

*Del litoral a las poblaciones lejanas

$$f_1(x_1) = \min_{d_1} \{r_1(x_1, d_1)\}$$

La tabla 6.6.7 resume los resultados para encontrar el óptimo.

Tabla 6.6.7 Obtención del beneficio óptimo en la 1ra. etapa.

Población anterior x_1	Indices de la 1ra. decisión	Longitud del camino r_1	Población siguiente \tilde{x}_1	Óptimo $f_1(x_1)$	Decisión óptima d_1
II ₁	1 1 1	6	1	6	1 1 1
II ₂	2 1 1	5	1	5	2 1 1
	2 2 1	7	2		
II ₃	3 2 1	3	2	3	3 2 1

Para la comunicación entre las poblaciones lejanas y las intermedias, etapa 2, se tiene:

*Entre poblaciones lejanas e intermedias.

$$f_2(x_2) = \min_{d_2} \{r_2(x_2, d_2) + f_1(x_2, d_2)\}$$

Estos valores se resumen en la tabla 6.6.8

Tabla 6.6.8 Obtención del beneficio óptimo en 2 etapas.

Población anterior x_2	Indices de la 2da. decisión	Longitud r_2	$x_1 = \tilde{x}_2$	$f_1(x_1)$	$r_2 + f_1$	Óptimo $f_2(x_2)$	Decisiones óptimas	
							d_{I^o}	d_{II^o}
III ₁	1 1 2	8	1	1	9	8	2 1 1	1 2 2
	1 2 2	3	2	5	8			
III ₂	2 2 2	2	2	5	7	6	3 2 1	2 3 2
	2 3 2	3	3	3	6			
III ₃	3 2 2	2	2	5	7	7	2 1 1	3 2 2
	3 3 2	5	3	3	8			

*Para la etapa 3 la fórmula para determinar el beneficio es: *Entre poblaciones intermedias y cercanas

$$f_3(x_3) = \min_{d_3} \{r_3(x_3, d_3) + f_2(x_3, d_3)\}$$

La búsqueda en este óptimo se resume en la tabla 6.6.9

Tabla 6.6.9 Obtención del beneficio óptimo en 3 etapas.

Población anterior x_4	Indices de la 3ra. decisión	Longitud r_3	$x_2 = \bar{x}_3$	$f_2(x_2)$	$r_3 + f_2$	Óptimo valor $f_3(x_3)$	Decisiones óptimas		
							d_I^*	d_{II}^*	d_{III}^*
IV ₁	1 1 3	7	1	8	15	11	321	232	123
	1 2 3	5	2	6	11				
IV ₂	2 3 3	6	3	7	13	13	211	322	233

*Finalmente para elegir las rutas entre la 1ra. localidad y las poblaciones cercanas se tiene: * Tramo final

$$f_4(x_4) = \min_{d_4} \{r_4(x_4, d_4) + f_3(x_3, d_3)\}$$

Para encontrar este mínimo se realizan los cálculos que aparecen en la tabla 6.6.10

Tabla 6.6.10 Obtención del beneficio óptimo en 4 etapas.

Población anterior x_4	Indices de la 4a. decisión	Longitud r_4	$x_3 = \bar{x}_4$	$f_3(x_3)$	$r_4 + f_3$	Valor óptimo $f_4(x_4)$	Decisiones óptimas			
							d_I^*	d_{II}^*	d_{III}^*	d_{IV}^*
x_5	1 1 3	8	1	11	19	17	321	232	123	114
	1 2 3	8	2	13	21	21	211	322	233	123

De esta última tabla se concluye que el camino de mínima longitud entre los puestos del litoral y la población V tiene una longitud de 17 a lo largo de la ruta 114, 123, 232 y 321, marcada con trazo grueso en la figura 6.6.18.

El lector interesado en profundizar sobre este tema puede consultar las refs. 1, 2, 5, 8 y 9. Los problemas 16 a 19 de la sección 6.8 ilustran diferentes aplicaciones de este método.

6.7. RUTA CRITICA

6.7.1-Introducción

En las diferentes fases de un proyecto, desde la planeación del programa hasta el retiro es necesario ejecutar con una secuencia lógica y a través del tiempo una serie de *actividades que pueden algunas ejecutarse en paralelo, o sea simultáneamente, mientras que otras tienen que realizarse en serie, es decir, no se puede iniciar una actividad antes de haber terminado la anterior. En la fase de construcción de un edificio, no puede iniciarse el montaje de la estructura si ésta es de acero, o su colado, si éste es de

*Actividades simultáneas ó en paralelo y actividades secuenciales ó serie.

concreto, si no se han terminado los cimientos. En la fabricación de un coche, no se puede proceder a armarlo, si no se cuenta con la carrocería, el motor, etc. Estas actividades se tienen que realizar secuencialmente. Por otra parte la fabricación del motor y el troquelado de las carrocerías puede realizarse simultáneamente. Esta orden de ejecución de actividades en un proyecto puede representarse mediante redes. Estas redes permiten determinar fundamentalmente:

- a) La secuencia temporal de las actividades.
- b) El tiempo de terminación de cada actividad y de todo el proyecto.
- c) Las actividades críticas, que si no se ejecutan dentro del tiempo previsto, pueden retrasar todo el proyecto.
- d) La asignación óptima de recursos.

Existen dos métodos para controlar la ejecución de proyectos:

*Como los dos métodos son muy similares, por eso solamente se estudia el de la ruta crítica (CPM). Si se conoce uno de ellos, puede comprenderse fácilmente el otro.

A continuación se describe cómo se establece la red de actividades de un proyecto, que sirve como base a estos métodos.

6.7.2 Red de actividades

*Esta red es una gráfica con nodos, representados mediante círculos, y unidos mediante segmentos dirigidos. Los nodos representan actividades y eventos, y los segmentos dirigidos la relación entre los eventos y las actividades.

La relación entre eventos y actividades es la siguiente:

*Con objeto de tener redes con un solo nodo inicial y terminal se incluyen estas gráficas, dos nodos ficticios, que representan actividades con cero tiempo de duración, que son el nodo inicial y el nodo terminal. Estos dos nodos son los únicos de la gráfica que solamente, o preceden a toda otra actividad del proyecto

- a) Método de la ruta crítica (CPM).
- b) Evaluación de programa y técnica de revisión (PERT).

*Ruta crítica (CPM).

Red:

nodos (actividades) unidos con segmentos dirigidos. (Secuencia temporal)

1. Una actividad o evento puede realizarse tanto en forma paralela con otra actividad como en forma secuencial.
2. Toda actividad o evento, exceptuando el primero, está precedido por una o varias actividades.
3. Toda actividad o evento, exceptuando el último, precede a una o varias actividades.

*Nodos ficticios: Nodo inicial y nodo final.

o están precedidos por todas las actividades restantes de la gráfica, tal como muestra la figura 6.7.1

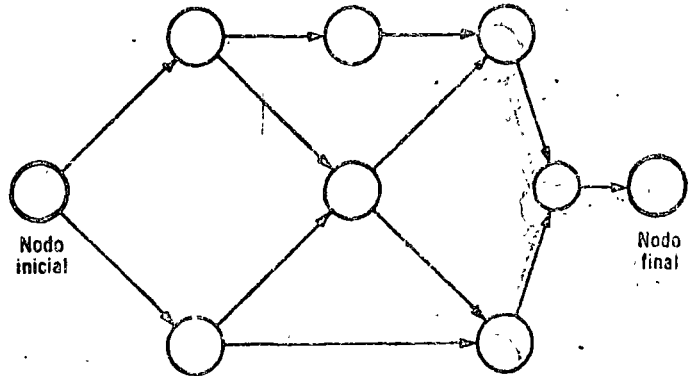


Fig. 6.7.1 Red de actividades.

Para construir la gráfica de actividades es necesario listar éstas, indicando su relación con otras actividades y el tiempo que toma ejecutarlas tal como aparece en el ejemplo 6.7.1.

Con este ejemplo se ilustra la construcción de la gráfica de actividades de un proyecto.

Con objeto de familiarizar a los lectores en el método de la ruta crítica se ha escogido un ejemplo, que no requiere para su comprensión de conocimientos en una especialidad. Tanto las actividades importantes del proyecto como su secuenciación se han seleccionado fundamentalmente para ilustrar aspectos importantes del método, tratando, sin embargo, de ser lo más realista posible.

Ejemplo 6.7.1.

El constructor señala que para la construcción de una pequeña casa es necesario realizar después de obtenidos los permisos y licencias de construcción necesarias, las siguientes actividades que se enumeran en la tabla 6.7.1. Estas actividades han sido designadas con letras. Se incluyen además en la tabla la duración en semanas de cada actividad, tomando en cuenta las limitaciones naturales y del personal, su relación con otras actividades y cómo el dueño desea conocer los pagos que debe hacer cada semana al constructor, el costo de cada una de ellas. Si una actividad dura varias semanas, su monto se supone que se cubre semanalmente por partes no necesariamente iguales.

Se desea establecer un diagrama de actividades de esta obra.

Solución:

A continuación se señalan con detalle los pasos que se siguen para trazar el diagrama de actividades.

Tabla 6.7.1 Lista de actividades del ejemplo 6.7.1

ACTIVIDAD	NOMBRE	DURACION	COSTO	OBSERVACIONES
A	Nivelar y rellenar el terreno	3	\$ 40,000.00	Ira. actividad.
B	Bardear el terreno	2	\$ 20,000.00	Se ejecuta después de A.
C	Construir cimientos	4	\$ 20,000.00	Puede ejecutarse simultáneamente con A.
D	Levantamiento de muros y colocado del techo	6	\$ 60,000.00	Se ejecuta después de C.
E	Colocación de tuberías y alambrado de la instalación eléct.	2	\$ 15,000.00	Se ejecuta después de D y B.
F	Colocación de ventanería.	1	\$ 15,000.00	Se ejecuta después de D y B y puede ejecutarse simultáneamente con E.
G	Aplanado, enyesado y colocación de mosaicos y muebles sanitarios.	2	\$ 20,000.00	Se ejecuta después de F y E
H	Pintura y detalles en los acabados	4	\$ 25,000.00	Se ejecuta después de G y de I.
I	Colocación de tierra en el jardín	2	\$ 20,000.00	Se ejecuta después de D.
J	Colocación de plantas	2	\$ 10,000.00	Se ejecuta después de H e I.

*Se empieza trazando tantos nodos como actividades tiene el proyecto, además de dos nodos adicionales, el inicial y el final.

I
Dibuje nodos

*El primero conviene trazarlo a la izquierda de la hoja y el segundo a la derecha. El resto conviene distribuirlo de acuerdo aproximadamente con su orden de ejecución. Por ejemplo la actividad A por ser la primera debe aparecer después del nodo inicial, y la B después de la A, o sea a su derecha y como la actividad C puede realizarse simultáneamente con la A, conviene que los nodos que representan a la actividad A y C estén sobre una misma vertical imaginaria, tal como lo muestra la figura 6.7.2.

o Nodo inicial a la izquierda y nodo terminal a la derecha.

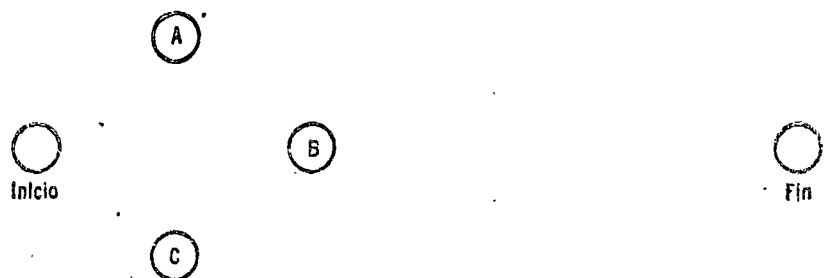


Fig. 6.7.2 Primeros nodos en el trazo de una red de actividades.

Si siguiendo el criterio anterior se termina trazando todos los nodos, tal como aparece en la figura 6.7.3, desde luego que las in-

dicasiones que se han dado sobre la colocación de los nodos en la gráfica, sólo son recomendaciones que permiten trazar una gráfica más clara. La relación temporal entre las actividades, se indica con segmentos de flecha dirigidos, que a continuación se anexarán a la gráfica.

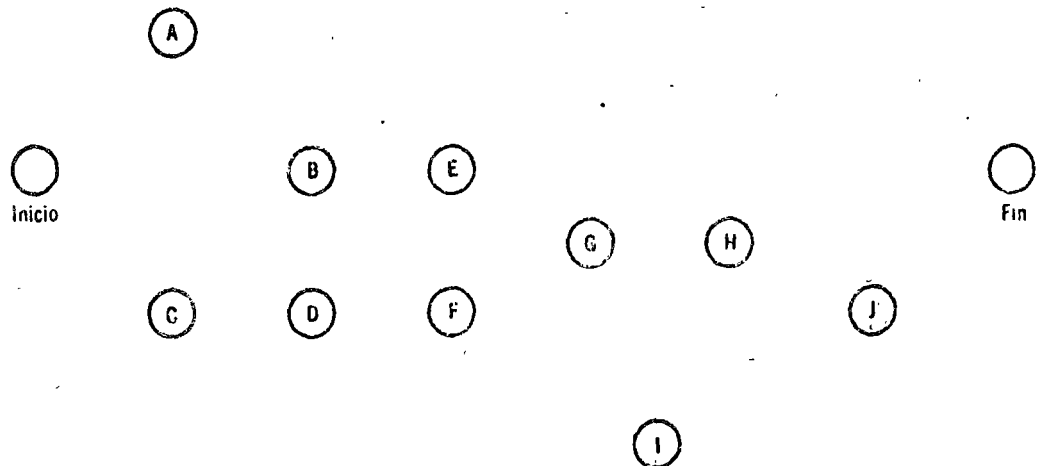


Fig. 6.7.3 Diagrama de actividades con todos los nodos.

*Las actividades A y C pueden realizarse simultáneamente y son las primeras del proyecto. Esta característica se indica mediante las flechas de la figura 6.7.4 que unen a estas dos actividades con el nodo inicial.

*Indique las relaciones temporales con segmentos dirigidos.

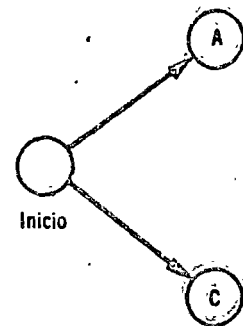


Fig. 6.7.4 Actividades iniciales.

De acuerdo con la tabla, la actividad B requiere para iniciarse que se haya terminado la actividad A. Esta secuencia se indica en el diagrama, como muestra la figura 6.7.5.

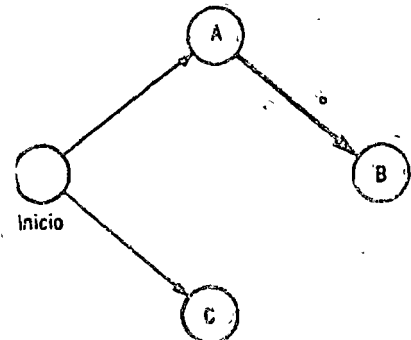


Fig. 6.7.5 La actividad B empieza después de terminada la actividad A.

En forma similar se traza el resto de los siguientes dirigidos que aparecen en la figura 6.7.6. Nótese en esta figura cómo se indica en un diagrama de actividades que varias de éstas (E y F) requieren para iniciarse que se hayan terminado otras actividades (B y D). Además, siendo J la última actividad del proyecto queda unida mediante un segmento dirigido al nodo terminal.

*En la gráfica 6.7.6 se muestra exclusivamente la relación temporal entre las actividades. El siguiente paso es el método de la ruta crítica, consiste en asignar valores a la red. La siguiente sección trata de este tema.

III
Asignación de valores a la red.

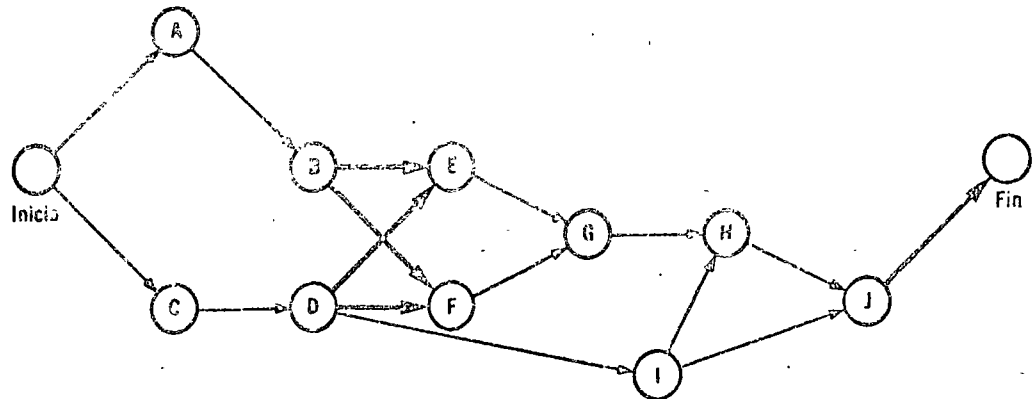


Fig. 6.7.6 Las actividades E y F requieren de la terminación de B y D para iniciarse.

En la tabla 6.7.1 aparece el tiempo de duración de cada actividad. *En esta sección se estudia cómo se relaciona el tiempo de duración de cada actividad con la duración mínima de todo el proyecto, desde su inicio hasta su terminación. Además, se encuentran aquellas actividades que determinan la duración mínima de todo el proyecto y cuya iniciación no puede posponerse o cuyo tiempo de ejecución no puede atrasarse sin alargar la duración de toda la obra. *Este tipo de actividades determinan la llamada *ruta crítica* del proyecto. Otras actividades no críticas pueden posponerse o alargarse sin afectar la duración del proyecto. El método que se expone a continuación permite determinar la holgura que se tiene en la iniciación o duración de estas actividades que se tiene en la iniciación o duración de estas actividades no críticas.

La duración de cada actividad aparece en el interior de cada nodo, tal como lo muestra la figura 6.7.7 para algunos nodos de la red del ejemplo 6.7.1. En este caso la duración se da en semanas.

Determinación de la ruta crítica.

*Determinación del tiempo mínimo de duración del proyecto.

*La ruta crítica está formada por actividades cuya iniciación no puede posponerse ni su duración alargarse sin atrasar el proyecto.

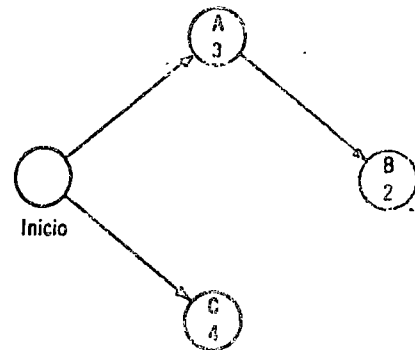


Fig. 6.7.7 Duración de las actividades.

Una vez indicada la duración de todas las actividades en la red, se procede a recorrer hacia adelante del nodo inicial al final. Durante esta fase se determinan los siguientes tiempos asociados al proyecto:

Estos tiempos se colocan en unos casilleros con una flecha dirigida hacia adelante, como los mostrados en la figura 6.7.8 que indican que fueron calculados al recorrer el proyecto del nodo de iniciación al de finalización.

Como el nodo inicial representa una actividad ficticia de cero duración y representa el inicio del proyecto, en los dos casilleros debe aparecer un cero, tal como lo muestra la figura 6.7.9.

Como las actividades A y C son las primeras del proyecto y pueden ejecutarse simultáneamente, su tiempo más próximo de iniciación es 0 y su tiempo más próximo de terminación es 0 más la duración de la actividad respectiva, tal como aparece en la figura 6.7.9 en los casilleros de las flechas dirigidas hacia adelante.

IV
*Recorrido de la red hacia adelante.

- a) El tiempo más próximo de iniciación (EST) de una actividad es lo más pronto que puede iniciarse una actividad.
- b) El tiempo más próximo de terminación (EFT) de una actividad es lo más temprano que puede terminarse. Es igual al tiempo más próximo de iniciación (EST) más la duración (D) de la actividad. Es decir:

$$EFT = EST + D \quad (6.7.1)$$

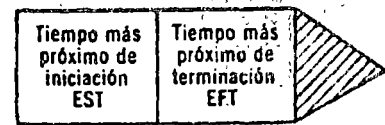


Fig. 6.7.8 Casilleros para indicar tiempos más próximos de iniciación y terminación.

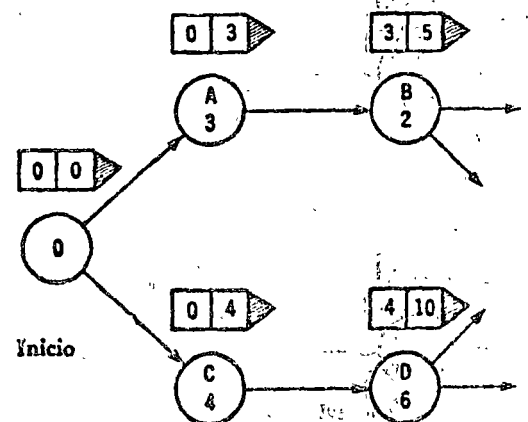


Fig. 6.7.9 Determinación de los tiempos más próximos de iniciación y terminación.

Como las actividades B y D no requieren más que de la terminación de A y C respectivamente, su tiempo más próximo de iniciación es igual al más próximo de terminación de la actividad que le precede, tal como muestra la figura 6.7.9.

El cálculo de tiempos más próximos para las actividades E y F requiere del siguiente razonamiento. Nótese que las actividades requieren para iniciarse, que se hayan terminado antes más de una actividad. Es decir, tienen más de una actividad que las precede inmediatamente. A la actividad E la preceden la B y la D. En estos casos el tiempo más próximo de iniciación es igual al tiempo más próximo de terminación máximo de las actividades que lo preceden. Es decir:

Tiempo más próximo de iniciación = Máximo tiempo más próximo de terminación de actividades precedentes.

Aplicando este criterio a la actividad E se tienen los resultados mostrados en la figura 6.7.10

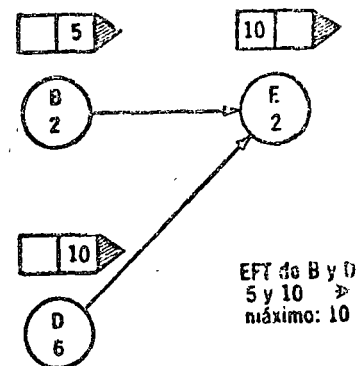


Fig. 6.7.10 Cálculo del tiempo más próximo de iniciación en actividades precedidas por varias otras.

Procediendo en forma similar para el resto de los nodos o actividades se obtienen todos los tiempos más próximos de iniciación y terminación que aparecen en la figura 6.7.11.

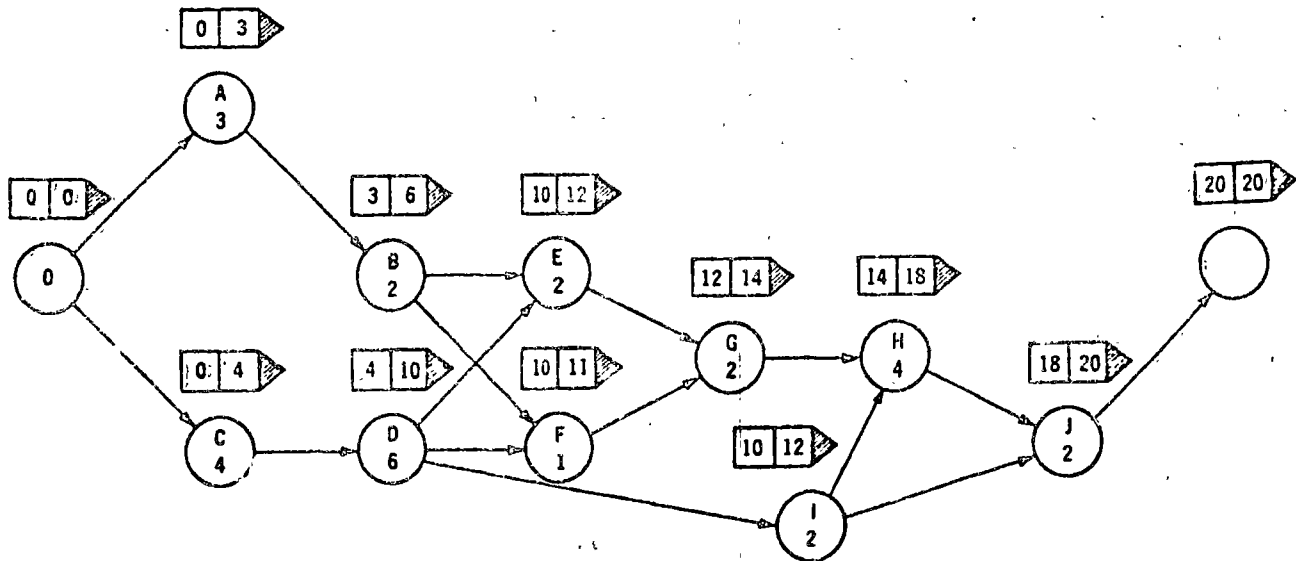


Fig. 6.7.11. Términos más próximos de iniciación y terminación del proyecto.

*Una vez recorrida la red en sentido directo del nodo inicial al nodo terminal y determinados los tiempos más próximos de iniciación y terminación, es necesario recorrer la red en sentido inverso, del nodo de terminación al nodo de iniciación. Durante este recorrido se determinan los siguientes tiempos.

*Recorrido de la red hacia atrás.

- a). El tiempo más lejano de terminación (LFT) es el tiempo lejano en el que puede terminarse una actividad sin retrasar ninguna otra actividad.
- b). El tiempo más lejano de iniciación (LST) es el tiempo más lejano en el que puede iniciarse una actividad sin atrasar ninguna otra. Es igual al tiempo más lejano de terminación de una actividad (LFT) menos su duración; es decir:

$$LST = LFT - D \quad (6.7.2)$$

Estos tiempos se colocan en unos casilleros adyacentes con una flecha dirigida hacia atrás, que indica que fueron calculados al recorrer el proyecto desde el nodo terminal al nodo inicial, como muestra la figura 6.7.12

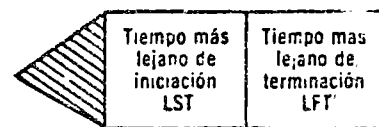


Fig. 6.7.12 Casilleros para indicar tiempo más lejano de iniciación y terminación.

Empezando con recorrer la red desde el nodo terminal, si el proyecto no debe sufrir retardo, entonces los tiempos más próximos de terminación y más lejano de terminación del nodo terminal deben ser iguales, tal como muestra la figura 6.7.13. Con la duración de la actividad asociada a este nodo terminal ficticio es nula, los tiempos más lejanos de iniciación y terminación son iguales.

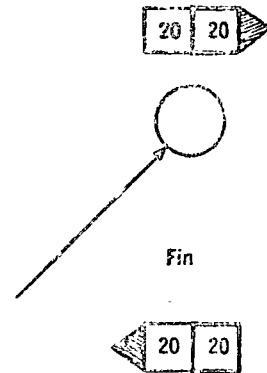


Fig. 6.7.13 Iniciación de los cálculos de tiempo más lejano de terminación e iniciación.

Procediendo en el sentido inverso de recorrido se llega al nodo J. El tiempo más lejano de terminación de este nodo debe ser 20 también, ya que de otra manera se atrasa el proyecto, y como la duración de la actividad es 2, el tiempo más lejano de iniciación es de $20 - 2 = 18$, tal como muestra la figura 6.7.14.

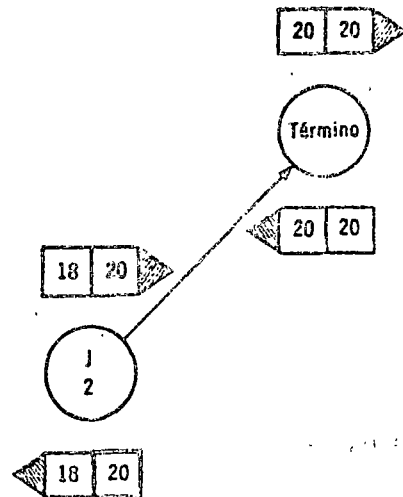


Fig. 6.7.14 Cálculo de tiempo más lejano de iniciación y terminación.

Continuando con el recorrido se nota como muestra la figura 6.7.15, que la actividad H precede sólo a la actividad J.

Para que las actividades posteriores a H, en este caso sólo J no sufra atraso, ninguna de las actividades precedentes debe termi-

378 Optimización

narse después del tiempo más lejano de iniciación de J, por lo tanto el diagrama se continúa como muestra la figura 6.7.15.

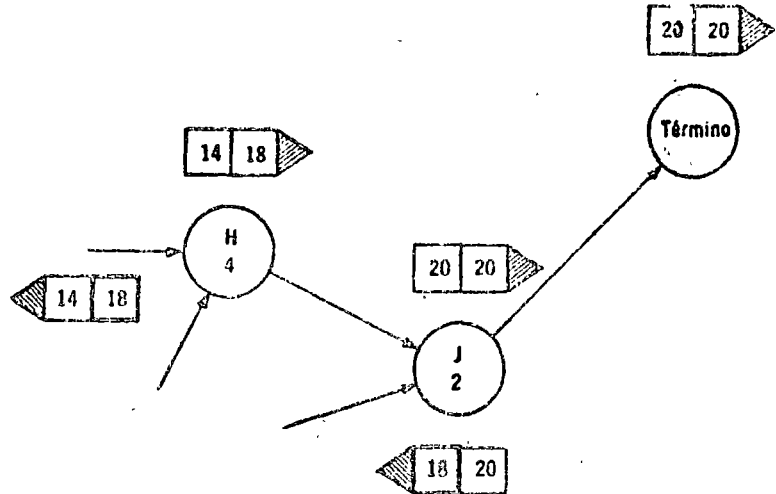


Fig. 6.7.15 Cálculos de tiempo más lejanos de iniciación y terminación.

Para la actividad I se observa que precede a la H y la J. Para que ninguna de estas últimas sufra atraso, el tiempo máximo de terminación de J, debe ser igual al mínimo tiempo más lejano de iniciación de las actividades inmediatas, para el caso de la actividad I, estos cálculos se muestran en la figura 6.7.16.

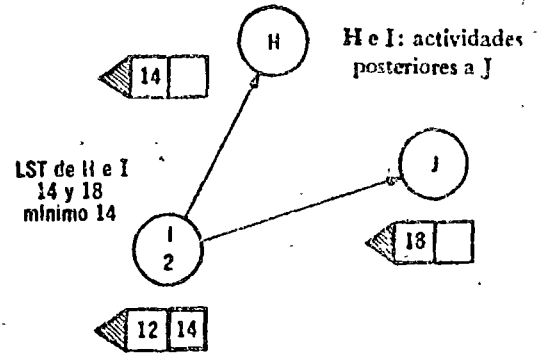


Fig. 6.7.16 Cálculo del tiempo más lejano de terminación.

En forma similar se continúa hasta obtener la gráfica completa de la red que aparece en la figura 6.7.17.

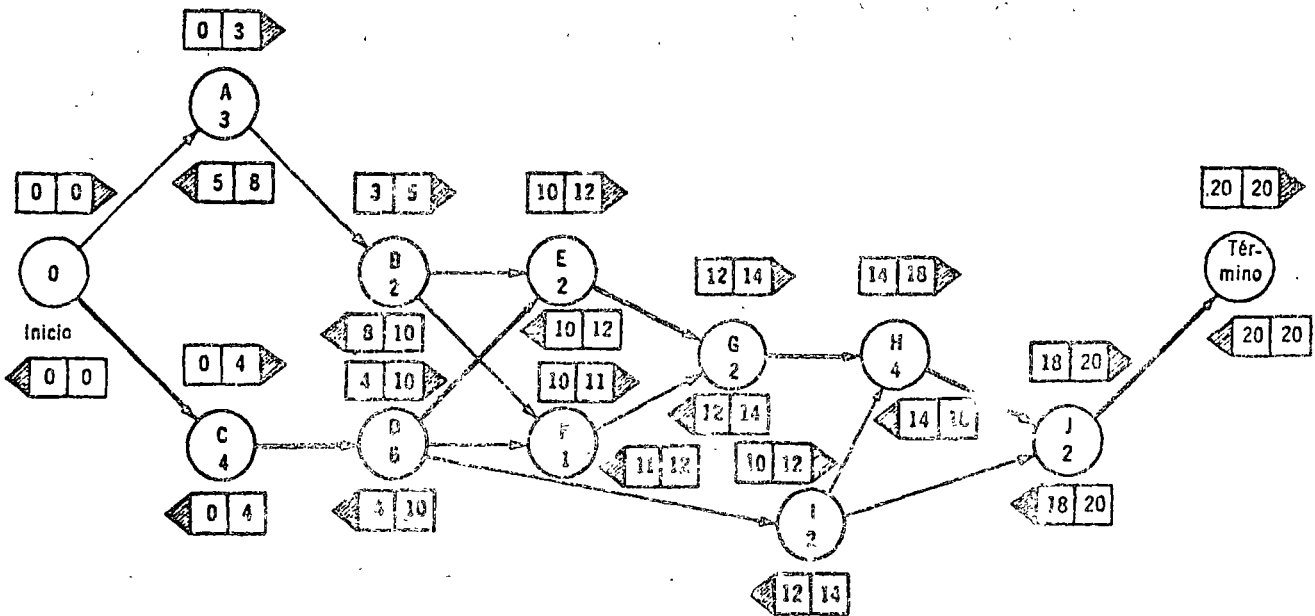


Fig. 6.7.17 Gráfica con todos los tiempos de iniciación y terminación del ejemplo 6.7.1.

VI

*Una vez terminada esta gráfica, puede determinarse la llamada *ruta crítica*, formada por aquellas actividades cuyo tiempo de iniciación o duración no puede prolongarse sin afectar al proyecto. Estas actividades deben tener su tiempo más próximo de iniciación igual al más lejano de iniciación. Es decir:

En la figura 6.7.17 se nota que las actividades con esta propiedad son la C, D, E, G, H, J que aparecen en la fig. 6.7.18

*Determinación de la ruta crítica.

EST = LST.

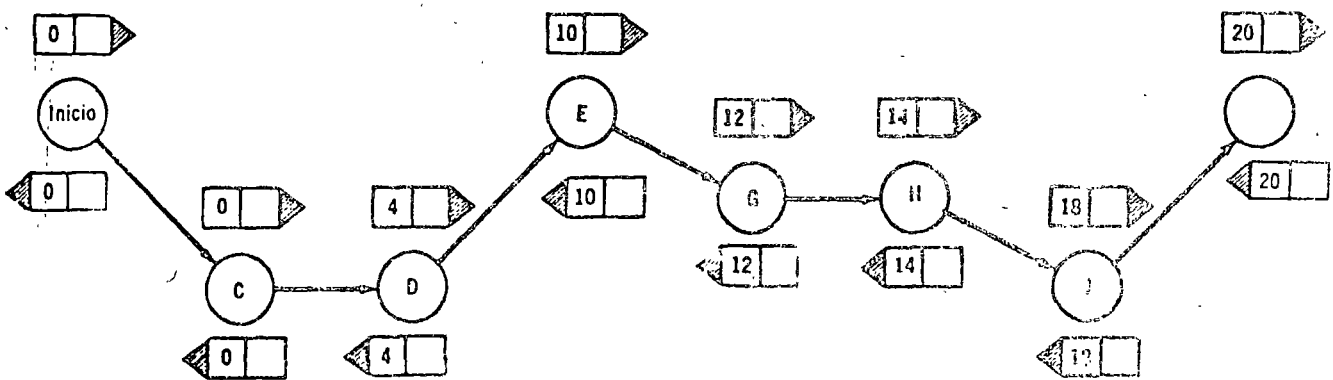


Fig. 6.7.18 Ruta crítica del ejemplo 6.7.1.

*Resulta de interés para el constructor determinar cuánto pueden atrasarse la iniciación de algunas actividades, desde luego las no situadas en la ruta crítica sin atrasar el proyecto. Este atraso recibe el nombre de *holgura total*. (TF)

*Holgura total:

máximo atraso posible en actividades sin retardar todo el proyecto.

380 Optimización

Este intervalo de tiempo es igual a la diferencia entre los tiempos más lejanos y próximos de iniciación de un proyecto, o entre los de terminación, es decir:

$$TF = LST - EST \quad (6.7.3)$$

$$TF = LFT - EFT$$

En la tabla 6.7.2 aparecen las holguras totales de todas las actividades, desde luego que las actividades de la ruta crítica tienen una holgura nula.

6.7.4 Asignación de recursos

Al plantear el problema del ejemplo 6.7.1 se mencionó que el dueño tiene interés en determinar cuáles son los pagos que debe realizar cada semana.

*Para determinar este calendario de pagos se procede a trazar un diagrama de barras como el mostrado en la figura 6.7.19. En el eje de las abscisas aparecen las semanas, desde la iniciación del proyecto. Las barras horizontales, representan las actividades del proyecto, y tienen una longitud igual a la duración de la actividad. Como el dueño no desea hacer erogaciones antes de lo necesario, no conviene empezar ninguna actividad antes de su tiempo más lejano de iniciación. Debido a esto se colocan las barras horizontalmente a partir del tiempo más lejano de iniciación, como muestra la figura 6.7.19 para la actividad A, cuyo tiempo más lejano de iniciación es 5 y su duración es de 3 semanas. Además debe incluirse en el diagrama información sobre los recursos que se necesitan para realizar la actividad. Suponiendo que el interés esté enfocado en los recursos económicos necesarios, en cada semana debe aparecer la erogación que tiene que realizarse. Para la actividad A, cuyo costo es de \$ 40,000 y tiene una duración de 3 semanas, se estima que semanalmente se requieren \$ 13,333. Esta información aparece en el diagrama de la fig. 6.7.19.

Para la actividad B, el tiempo más lejano de iniciación es de 8, su costo es de \$ 20,000 y su duración de 2 semanas. Se estima que la erogación semanal es de \$ 10,000.00, tal como aparece en la figura 6.7.20.

VII

*Trazo de un diagrama de barras:

Tabla 6.7.2 Holguras totales del proyecto del ejemplo 6.7.1.

Actividad	Holgura total
A	5
B	5
C	0
D	0
E	0
F	1
G	0
H	0
I	2
J	0

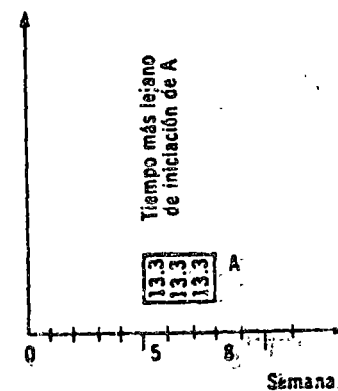


Fig. 6.7.19 Diagrama de barras con la actividad A.

La actividad C, tiene un tiempo más lejano de iniciación de 0, una duración 4, y un costo total de \$ 20,000.00. La erogación semanal para esta actividad se estima que será de: \$ 6,000, \$ 6,000, \$ 4,000 y \$ 4,000. Esta información también está contenida en la fig. 6.7.20.

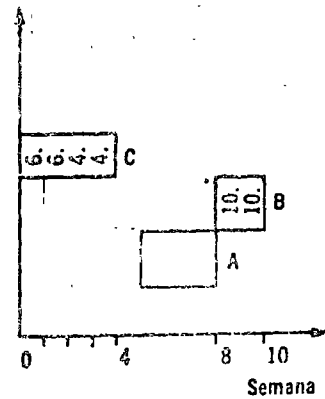


Fig. 6.7.20 Diagramas de barras de la actividad B.

*Debe hacerse notar que este tipo de diagrama puede contener información no sólo sobre recursos económicos, sino también de otra índole, como recursos humanos, maquinaria, etc. La cantidad de recursos que se emplea en cada unidad de tiempo depende del tipo de actividad. En este ejemplo se estima que las actividades A y B requieren de igual cantidad de dinero, durante cada semana de su duración, mientras que la actividad C, requiere de recursos no uniformemente distribuidos.

o Pueden indicarse diferentes necesidades de recursos en el diagrama de barras.

Siguiendo con la metodología expuesta, se termina de construir el diagrama. El diagrama completo de barras aparece en la fig. 6.7.21.

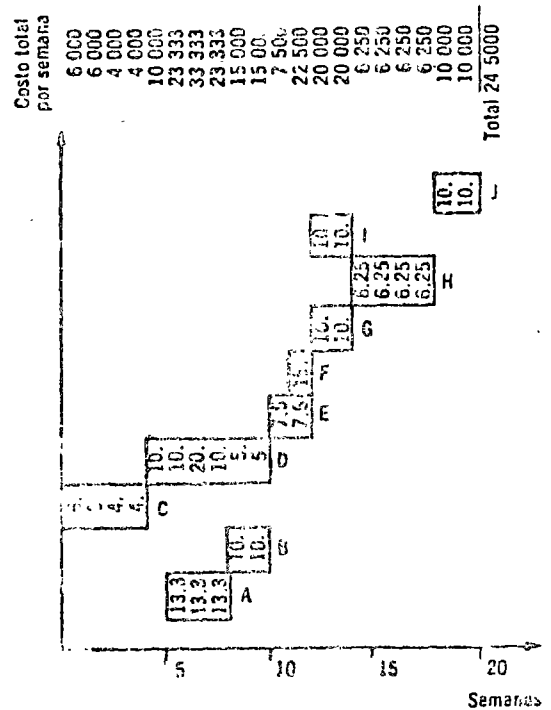


Fig. 6.7.21 Diagrama completo de barras del proyecto del ejemplo 6.7.1.

382 Optimización

Finalmente puede obtenerse de este diagrama el calendario de pagos semanales, simplemente sumando las cantidades que aparecen en cada columna. Este diagrama muestra además qué actividades deben ejecutarse cada semana. En la semana 10, se ejecutan las actividades B y D, que requieren de \$ 10,000 y 5,000 respectivamente. Por lo tanto esta semana se necesitan \$ 15,000.00 del recurso dinero, tal como muestra la fig. 6.7.22.

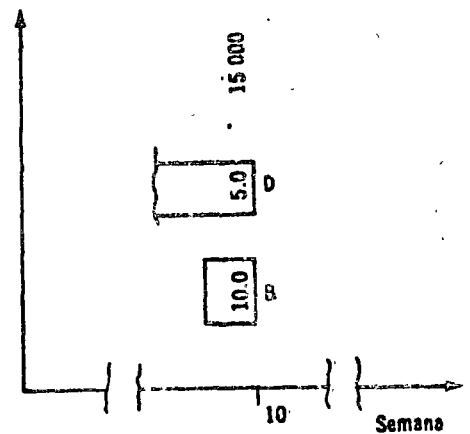


Fig. 6.7.22 Actividades y recursos de la semana 10.

Desde luego que en proyectos de mayor envergadura que la construcción de una casa, es necesario recurrir a la computadora digital para encontrar tiempos más próximos y lejanos, holguras, ruta crítica y distribución de recursos. En el apéndice A, el programa No. 19, permite calcular la ruta crítica de un proyecto. Los resultados de este programa para el ejemplo 6.7.1 aparecen en las páginas siguientes.

Como datos de este programa es necesario indicar el número de actividades, incluyendo las actividades ficticias de iniciación y terminación, la duración de cada actividad y la subordinación entre actividades, en forma de matriz tal como aparece en la tabla 6.7.3. El elemento a_{ij} de esta matriz tiene valor unitario si la actividad i requiere que se haya ejecutado la actividad j .

Los resultados del programa A.19 para el ejemplo 6.7.1 aparecen en la tabla 6.7.4.

Tabla 6.7.3 Matriz de subordinación del ejemplo 6.7.1 para el programa A.19.

MATRIZ DE SUBORDINACION DE ACTIVIDADES

1	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0	0	0
6	0	0	1	0	1	0	0	0	0	0	0
7	0	0	1	0	1	0	0	0	0	0	0
8	0	0	0	0	0	1	1	0	0	0	0
9	0	0	0	0	0	0	1	0	1	0	0
10	0	0	0	0	1	0	0	0	0	0	0
11	0	0	0	0	0	0	0	1	1	0	0
12	0	0	0	0	0	1	0	0	0	0	1

Tabla 6.7.4 Resultados del programa A.19 para el ejemplo 6.7.1

LOS RESULTADOS OBTENIDOS SON

ACTIVIDAD	DURACION	EST	EFT	LST	LFT	TFL	TFT	
1	0	0	0	0	0	0	0	R.C.
2	3	0	3	5	8	0	5	
3	2	3	5	8	10	5	5	
4	4	0	4	0	4	0	0	R.C.
5	6	4	10	4	10	0	0	R.C.
6	2	10	12	10	12	0	0	R.C.
7	1	10	11	11	12	1	1	
8	2	12	14	12	14	0	0	R.C.
9	4	14	18	14	18	0	0	R.C.
10	2	10	12	12	14	2	2	
11	2	18	20	18	20	0	0	R.C.
12	0	20	20	20	20	0	0	R.C.



centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



MÉTODOS NUMÉRICOS Y APLICACIONES
CON LA COMPUTADORA DIGITAL

PROGRAMACION LINEAL

M. EN C. VERONICA CZITROM

ABRIL, 1978

3.4 PROGRAMACION LINEAL

3.4.1 Un ejemplo

*Existen muchos problemas de maximización o minimización, cuyo modelo matemático se puede resolver con la técnica de optimización conocida con el nombre de *programación lineal*. Como en general se requieren muchos pasos algebraicos para resolver un problema por este método, se han desarrollado algoritmos y, basados en ellos, programas de computadora digital para encontrar la solución óptima.

*La estructura de los problemas que pueden resolverse con la técnica de programación lineal es siempre la misma; con un buen programa pueden resolverse todos los problemas de este tipo, sin necesidad de tener que escribir programas especiales para la solución de problemas particulares. *Problemas de optimización que no pueden resolverse con el modelo de programación lineal, como los de programación dinámica, con frecuencia no tienen esta característica y es necesario desarrollar programas particulares para obtener la solución de problemas específicos.

El método simplex que se estudiará en la subsección 3.4.6 constituye la metodología matemática más conocida y empleada, para resolver problemas de programación lineal. Existen otros métodos, como el método gráfico (subsección 3.4.4), el método de prueba y error, el método vectorial el método de multiplicadores de Lagrange y programación dinámica, que se pueden consultar en la bibliografía.

*Muchos problemas de optimización ya tienen un modelo lineal, y puede aplicarse directamente la técnica de programación lineal para encontrar el máximo o el mínimo del problema.

*Otros modelos son no lineales. Si esta no linealidad es pequeña, puede aproximarse por modelos lineales. Existen varias técnicas para ello, siendo la linealización y el cambio de variables de ellas. La mayoría de los textos de optimización o programación matemática que se listan en la bibliografía, tratan este tópico.

*La programación lineal es una técnica de optimización.

*Todos los problemas de programación lineal tienen el mismo modelo matemático.

*No existen modelos generales para problemas de programación dinámica.

*Muchos modelos ya son lineales.

*Existen técnicas para aplicar métodos de optimización lineales a modelos no lineales.

*El primer ejemplo ilustra un problema de producción, al cual es aplicable la técnica de programación lineal. Este ejemplo sirve para introducir varios conceptos de programación lineal, como:

*Conceptos de programación lineal:

función objetivo
restricciones
solución factible
solución óptima

Ejemplo 3.4.1

Un fabricante manufactura dos productos que tienen igual precio en el mercado. Cada producto requiere de una serie de operaciones de maquinado en 4 máquinas distintas.

La tabla 3.4.1 muestra los tiempos de maquinado. *Así por ejemplo, para producir una unidad del producto 1 se requieren:

*Para producir una unidad del producto 1:

0 horas de la máquina A
1 hora de la máquina B
2 horas de la máquina C
3 horas de la máquina D

	MAQUINA			
	A	B	C	D
PRODUCTO 1	0	1	2	3
PRODUCTO 2	2	2	1	0

Tabla 3.4.1 Horas de maquinado por pieza del ejemplo 3.4.1.

Como existe un número determinado de máquinas de cada tipo, el número de horas de máquina disponibles está limitado. La tabla 3.4.2 muestra esta limitación.

MAQUINA			
A	B	C	D
180	210	240	330

Tabla 3.4.2 Tiempos de máquina disponibles para el ejemplo 3.4.1.

El fabricante tiene que cubrir los gastos fijos de su empresa, por lo que tiene que vender al día por lo menos un total de 80 piezas, entre los productos 1 y 2. Establezca un modelo y determine cuántas piezas de cada artículo 1 y 2 debe vender al día para maximizar su ganancia.

•Para construir un modelo matemático que permita resolver este problema, hay que empezar definiendo variables. Sean

De acuerdo con la descripción del problema, es necesario maximizar el número total de piezas producidas diariamente, es decir:

•Para introducir el método de solución de los problemas de programación lineal, conviene emplear un método gráfico. Por esta razón se representará gráficamente el modelo formal de este problema.

Para trazar la recta que representa a la ecuación (3.4.1) se hace el siguiente razonamiento: Supongamos que $x_1 + x_2$ toma un valor z .

Cuando en esta recta

y si

Uniendo los puntos V y W de la recta $x_1 + x_2 = z$, se obtiene el trazo de la recta (3.4.1) para el valor z , como muestra la figura 3.4.1.

Solución:

•Se empieza definiendo variables:

x_1 = número de piezas del artículo 1 que se fabrican al día.

x_2 = número de piezas del artículo 2 que se fabrican al día.

$$x_1 + x_2 = z(\max) \quad (3.4.1)$$

•Se presenta la solución gráfica del problema.

$$x_1 + x_2 = z.$$

$$x_1 = 0, \text{ entonces } x_2 = z \text{ (punto V)}$$

$$x_2 = 0, \text{ entonces } x_1 = z \text{ (punto W)}$$

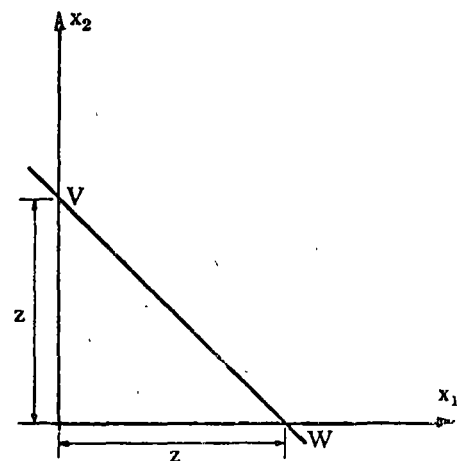


Fig. 3.4.1 Representación de $x_1 + x_2 = z$

*A continuación se establecerán las relaciones matemáticas que expresan las condiciones restantes del problema.

Consideremos el empleo de la máquina A. Como se producen x_2 piezas del tipo 2 y cada una requiere 2 horas de maquinado con el equipo A, en total esta máquina se emplea $2x_2$ horas. Para producir la pieza 1 no se emplea la máquina A, por la tabla 3.4.1. Por lo tanto el número de horas $2x_2$ que se emplea la máquina A, debe ser menor o igual que el tiempo disponible de la máquina A, que es de 180 horas, de acuerdo con la tabla 3.4.2. *Por lo tanto se tiene

Para encontrar la gráfica de esta relación se procede de la siguiente forma. Como

entonces

*Primero se traza la recta correspondiente a la condición de igualdad, es decir a:

Esta recta aparece en la figura 3.4.2.

Dicha recta divide el plano en dos regiones, R_1 y R_2 ; en la primera $x_2 \geq 90$, y en la segunda $x_2 \leq 90$. Por lo tanto la relación (3.4.2) representa a la región R_2 , que se muestra ashurada en la figura.

*Restricciones del problema:

*Empleo de la máquina A:

$$2x_2 \leq 180$$

$$2x_2 \leq 180$$

$$x_2 \leq 90$$

(3.4.2)

*Trace primero

$$x_2 = 90$$

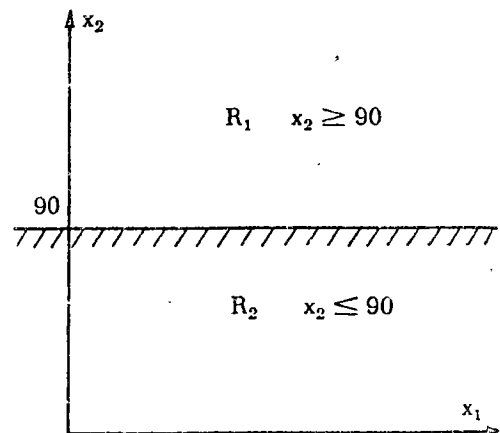


Fig. 3.4.2 Representación de $x_2 \leq 90$.

*La máquina B se emplea 1 hora para producir una pieza número 1, y 2 horas para una pieza 2, por lo que

representa el número de horas que se emplea la máquina B para producir los dos tipos de piezas. Esta suma tiene que ser menor que el número de horas disponibles para la máquina B, que es de 210, es decir

Al igual que la ecuación (3.4.2), esta ecuación también representa una región del plano. *Para determinarla, se empieza encontrando la recta que corresponde a la igualdad, a saber

Para dibujar esta recta considere que, si

y que cuando

Estos puntos aparecen en la figura 3.4.3, al igual que la recta que los une. La recta:

*divide al plano en las regiones R_1 y R_2 . Para determinar en qué región se cumple la desigualdad (3.4.3), se busca un punto cualquiera en alguna de las dos regiones, por ejemplo el origen (0,0) en R_2 . Se sustituyen los valores de este punto en la relación (3.4.3), y se determina el sentido de la desigualdad en la región R_2 . Como $0 + 2 \times 0 = 0 < 210$, en la región R_2 se cumple la desigualdad (3.4.3). En la figura 3.4.3 aparece ashrada la región representada por la desigualdad (3.4.3).

*Empleo de la máquina B:

$$x_1 + 2x_2$$

$$x_1 + 2x_2 \leq 210 \tag{3.4.3}$$

*Trace la recta que corresponde a la igualdad:

$$x_1 + 2x_2 = 210$$

$$x_1 = 0 \text{ entonces } x_2 = 105 \text{ (punto V)}$$

$$x_2 = 0 \text{ entonces } x_1 = 210 \text{ (punto W)}$$

$$x_1 + 2x_2 = 210$$

*divide el plano en zonas R_1 y R_2 .

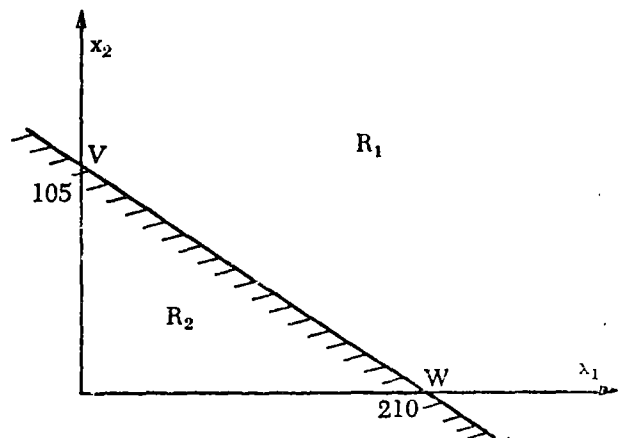


Fig. 3.4.3 Representación de $x_1 + 2x_2 \leq 210$.

Para encontrar las ecuaciones correspondientes a las máquinas C y D se procede de igual manera, obteniéndose:

$$\begin{aligned} \text{Máquina C:} \\ 2x_1 + x_2 &\leq 240 & (3.4.4) \\ \text{Máquina D:} \\ 3x_1 &\leq 330 \end{aligned}$$

o sea

$$x_1 \leq 110 \quad (3.4.5)$$

°Por otra parte, el enunciado del problema indica que es indispensable mantener un nivel mínimo de producción, a saber:

°Nivel mínimo de producción:

$$x_1 + x_2 \geq 80 \quad (3.4.6)$$

°Finalmente deben incluirse las condiciones de *no-negatividad* de las variables x_1 y x_2 :

°Condiciones de no-negatividad

$$x_1 \geq 0 \quad (3.4.7)$$

$$x_2 \geq 0 \quad (3.4.8)$$

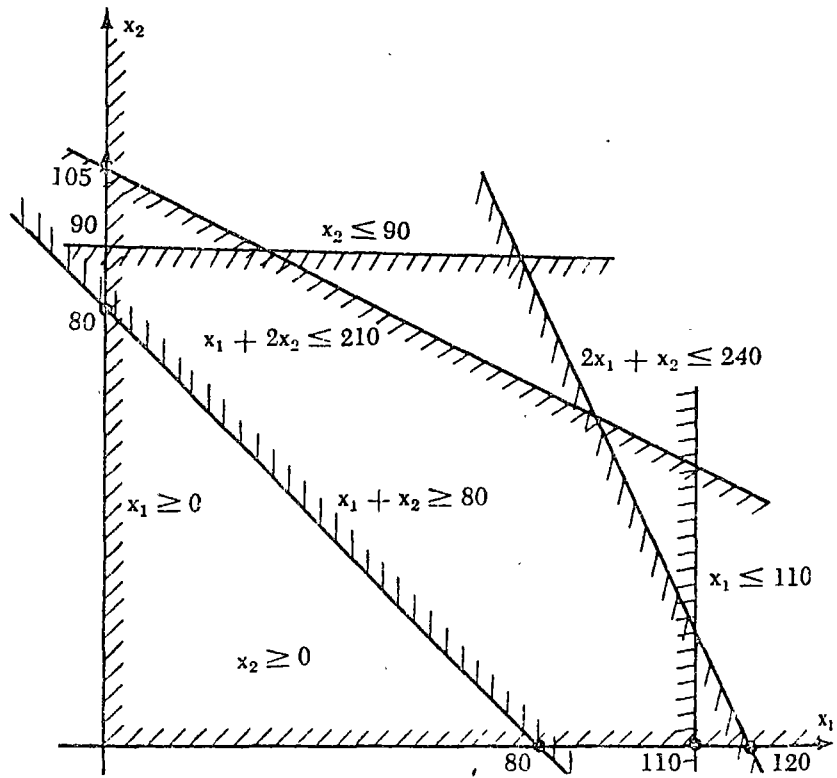


Fig. 3.4.4 Representación de las regiones definidas por las ecuaciones (3.4.2) a (3.4.8).

ya que no tiene sentido producir un número negativo de piezas.

La representación gráfica de las ecuaciones (3.4.2) a (3.4.8) aparece en la figura 3.4.4.

Las ecuaciones:

$x_2 \leq 90$	(3.4.2)
$x_1 + 2x_2 \leq 210$	(3.4.3)
$2x_1 + x_2 \leq 240$	(3.4.4)
$x_1 \leq 110$	(3.4.5)
$x_1 + x_2 \leq 80$	(3.4.6)
$x_1 \leq 0$	(3.4.7)
$x_2 \leq 0$	(3.4.8)

consideradas simultáneamente definen una región en el plano (x_1, x_2) . Para encontrar esta región deben sobreponerse las regiones que define cada una de las ecuaciones (3.4.2) a (3.4.8), mostradas en la figura 3.4.4, y encontrar la región que es común a todas ellas (su intersección). Esta región aparece en la figura 3.4.5.

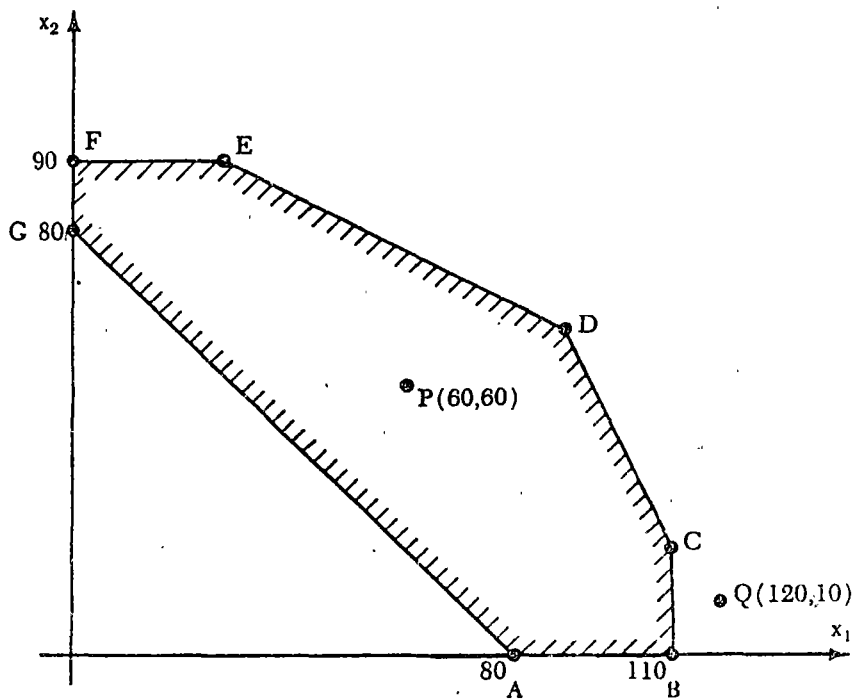


Fig. 3.4.5 Región definida por las restricciones del ejemplo 3.4.1.

Las ecuaciones (3.4.1) a (3.4.8) constituyen el modelo formal del ejemplo 3.4.1.

En la siguiente subsección se introducen algunas definiciones importantes.

3.4.2 Definiciones

*Las variables que intervienen en un problema de programación lineal se conocen con el nombre genérico de *actividades*, y su valor numérico como *nivel de actividad*.

*En el problema 3.4.1 se tenían dos actividades, la producción de artículos 1 representada con la variable x_1 , y la producción de artículos 2 con x_2 .

*El modelo matemático de un problema de programación lineal consta de dos partes:

*La *función objetivo* es la expresión matemática que se desea optimizar, es decir según el problema minimizar o maximizar. Con frecuencia, la función objetivo está dada por un costo o un beneficio, medido en unidades monetarias. En el ejemplo 3.4.1 la función objetivo es:

*Variables \equiv Actividades
Valor de las variables \equiv Nivel de actividad

*Actividades $x_1, x_2 \equiv$
producción de artículos 1 y 2

*Partes de un modelo de programación lineal:

- a) Función objetivo
- b) Restricciones.

*La función objetivo se maximiza o minimiza.

$$x_1 + x_2 = z \quad (3.4.1)$$

que se desea maximizar.

*En la mayoría de los problemas existen *restricciones*, representadas por un grupo de ecuaciones. Debido a restricciones en la capacidad de producción, el industrial no puede producir un número arbitrario de piezas. Estas restricciones limitan el valor que pueden tomar las variables.

Para el ejemplo 3.4.1, la figura 3.4.5 muestra la región definida por las restricciones. Cualquier punto dentro de esta región representa un posible nivel de producción que no viola ninguna restricción. Así por ejemplo en el punto P:

*Las restricciones determinan posibles valores de las variables.

P(60,60)

se cumplen las restricciones:

$$x_2 \leq 90 \quad (3.4.2)$$

$$60 \leq 90 \quad (3.4.3)$$

$$x_1 + 2x_2 \leq 210$$

$$60 + 2 \times 60 = 180 \leq 210$$

$$2x_1 + x_2 \leq 240 \quad (3.4.4)$$

$$2 \times 60 + 60 = 180 \leq 240 \quad (3.4.5)$$

$$x_1 \leq 110 \quad (3.4.6)$$

$$60 \leq 110$$

$$x_1 + x_2 \geq 80 \quad (3.4.7)$$

$$60 + 60 = 120 \geq 80$$

$$x_1 \geq 0 \quad (3.4.8)$$

$$60 \geq 0$$

$$x_2 \geq 0 \quad (3.4.8)$$

$$60 \geq 0$$

Por otra parte, un punto fuera de la región como el

viola una o varias de las restricciones. En este punto se violan las ecuaciones

$$Q(120,10)$$

$$2x_1 + x_2 \leq 240 \quad (3.4.4)$$

$$2 \times 120 + 10 = 250 > 240 \quad (3.4.5)$$

$$x_1 \leq 110$$

$$120 > 110$$

La ecuación (3.4.4) representa la restricción que impone la máquina C, y la (3.4.5) la que impone la máquina D. Al violarse se están tratando de emplear dichas máquinas más horas de las disponibles. No es posible mantener el nivel de producción representado por el punto Q.

*Cualquier punto dentro de la región mostrada en la figura 3.4.5 representa un nivel de actividad posible, es decir cumple con las restricciones:

*Las restricciones definen la zona de soluciones factibles.

$$x_2 \leq 90 \quad (3.4.2)$$

$$x_1 + 2x_2 \leq 210 \quad (3.4.3)$$

$$2x_1 + x_2 \leq 240 \quad (3.4.4)$$

$$x_1 \leq 110 \quad (3.4.5)$$

$$x_1 + x_2 \geq 80 \quad (3.4.6)$$

$$x_1 \geq 0 \quad (3.4.7)$$

$$x_2 \geq 0 \quad (3.4.8)$$

Cualquier punto que cumple con todas las restricciones se llama *solución factible*; el conjunto de soluciones factibles representa una *zona o región de soluciones factibles*. Esta región se ilustra en la figura 3.4.5 para el ejemplo 3.4.1.

*El problema de optimización puede por lo tanto replantearse en los siguientes términos:

*Replanteamiento del problema de programación lineal:

Buscar entre las soluciones factibles la solución óptima.

3.4.3 Funciones lineales

*Debe hacerse notar que en un problema de programación lineal,

*En programación lineal, la función objetivo

tanto la función objetivo como las restricciones deben ser lineales.

y las restricciones son funciones lineales,

Una *función lineal* satisface dos condiciones:

- a) Homogeneidad
- b) Aditividad

a) Homogeneidad:

Una función es homogénea si, al cambiar en cierta proporción todas las variables independientes, el valor de la función cambia en la misma proporción. En términos matemáticos, se tiene:

$$F(kx_1, kx_2, \dots, kx_n) = k F(x_1, x_2, \dots, x_n) \quad (3.4.9)$$

b) Aditividad

Una función es aditiva, si el valor de la función para la suma de dos conjuntos de variables independientes, es igual a la suma de las funciones para cada uno de los conjuntos de variables, es decir:

$$F(x_1 + x'_1, x_2 + x'_2, \dots, x_n + x'_n) = F(x_1, x_2, \dots, x_n) + F(x'_1, x'_2, \dots, x'_n) \quad (3.4.10)$$

En resumen, una función lineal satisface la siguiente relación:

$$F(kx_1 + k'x'_1, \dots, kx_n + k'x'_n) = k F(x_1, x_2, \dots, x_n) + k' F(x'_1, x'_2, \dots, x'_n) \quad (3.4.11)$$

Los siguientes ejemplos se emplean para ilustrar el concepto de linealidad.

Ejemplo 3.4.2

Establezca si la función

$$F(x_1, x_2) = x_1 + 2x_2$$

es lineal.

Solución:

Es necesario determinar si la función F cumple con la relación (3.4.11). Como

$$F(kx_1 + k'x'_1, kx_2 + k'x'_2) = (kx_1 + k'x'_1) + 2(kx_2 + k'x'_2)$$

es igual a

$$kF(x_1, x_2) + k'F(x'_1, x'_2) = k(x_1 + 2x_2) + k'(x'_1 + 2x'_2) = (kx_1 + k'x'_1) + 2(kx_2 + k'x'_2)$$

la función $F(x_1, x_2)$ es lineal.

Ejemplo 3.4.3

Determine si las funciones

$$F(x_1, x_2) = 3$$

$$G(x_1, x_2) = x_1^2$$

$$H(x_1, x_2) = x_1 x_2$$

son lineales.

Solución:

En este caso

$$F(kx_1 + k'x'_1, kx_2 + k'x'_2) = 3 \neq k^3 + k'^3 = kF(x_1, x_2) + k'F(x'_1, x'_2)$$

$$G(kx_1 + k'x'_1, kx_2 + k'x'_2) = (kx_1 + k'x'_1)^2 \neq kx_1^2 + k'x'^2_1 = kG(x_1, x_2) + k'G(x'_1, x'_2)$$

$$H(kx_1 + k'x'_1, kx_2 + k'x'_2) = (kx_1 + k'x'_1)(kx_2 + k'x'_2) \neq kx_1 x_2 + k'x'_1 x'_2 = kH(x_1, x_2) + k'H(x'_1, x'_2)$$

por lo que ninguna de las tres funciones es lineal.

De los ejemplos anteriores puede concluirse lo siguiente:

En una función lineal la relación entre las variables es aditiva, y no aparecen ni productos entre ellas, ni potencias diferentes de la primera.

El concepto de linealidad es muy importante en el análisis de sistemas, por lo que es indispensable dominar este concepto. *Muchas de las técnicas analíticas que se emplean para resolver problemas, como la programación lineal, sólo son aplicables a problemas lineales, donde las funciones son lineales. Debido a ello con frecuencia se recurre a procedimientos especiales, como linealización y cambios de variable, para lograr modelos lineales que se puedan resolver con técnicas de análisis lineal.

*Muchas técnicas de solución son aplicables únicamente a modelos lineales.

En el ejemplo 3.4.1 la función objetivo

$$x_1 + x_2 = z$$

es lineal, pues la función

$$F(x_1, x_2) = x_1 + x_2$$

es lineal:

$$F(kx_1 + k'x'_1, kx_2 + k'x'_2) = (kx_1 + k'x'_1) + (kx_2 + k'x'_2) = k(x_1 + x_2) + k'(x'_1 + x'_2) = kF(x_1, x_2) + k'F(x'_1, x'_2)$$

Además, los miembros izquierdos de las restricciones (3.4.2) a (3.4.8) son lineales. Por ejemplo en la restricción (3.4.3).

$$x_1 + 2x_2 \leq 210$$

la función

$$F(x_1, x_2) = x_1 + 2x_2$$

es lineal, como ya se observó en el ejemplo 3.4.2.

El ejemplo 3.4.1 es por lo tanto un problema lineal. A continuación se ilustra la solución gráfica de dicho problema lineal de maximización.

3.4.4 Solución gráfica

Al final de la subsección 3.4.2 se señaló que encontrar la solución de un problema de programación lineal, consiste en buscar entre las soluciones factibles, aquella que optimice la función objetivo.

La función objetivo por maximizar del ejemplo 3.4.1 es:

$$x_1 + x_2 = z \quad (3.4.1)$$

Cualquier recta paralela a las mostradas en la figura 3.4.6 puede ser la función objetivo; su posición exacta depende del valor de z . Cuanto mayor sea z , tanto más alejada del origen está la recta.

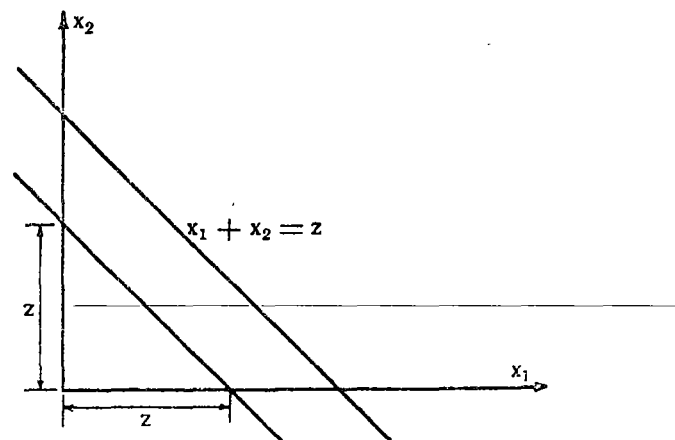


Fig. 3.4.6 Gráfica de la función objetivo.

Para determinar cual de las rectas anteriores está asociada al máximo valor de la función objetivo y todavía es una solución factible, es necesario combinar la gráfica de la figura 3.4.6 con la de la figura 3.4.5 de soluciones factibles. En la figura 3.4.7 aparece esta combinación.

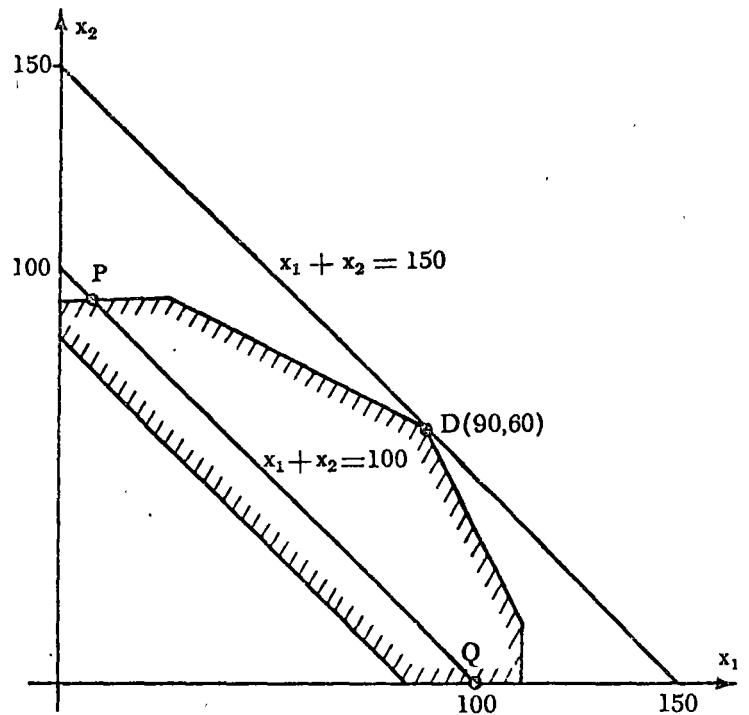


Fig. 3.4.7 Región de soluciones factibles y función objetivo.

Un nivel de producción correspondiente a un valor de la función objetivo z de 100, es factible, ya que cualquier punto (x_1, x_2) sobre el tramo PQ es factible. *Para encontrar el máximo, debe trasladarse la recta de la función objetivo hasta que su distancia del origen sea máxima y todavía tenga por lo menos un punto dentro de la zona factible. *Esta situación se presenta en el punto D de la figura 3.4.7, con coordenadas

*Para encontrar el máximo, aleje la función objetivo lo más posible del origen dentro de la zona de soluciones factibles.

*Óptimo: punto D

$$x_1 = 90$$

$$x_2 = 60$$

En este punto óptimo se tienen las condiciones de empleo de recursos que muestra la tabla 3.4.3. Esta tabla contiene además la información de las dos anteriores.

El renglón de empleo de recursos muestra el número de horas que se emplea cada máquina en el punto óptimo. *Por ejemplo, de acuerdo con la relación (3.4.3), la expresión

*Empleo de la máquina B

$$x_1 + 2x_2 = 90 + 2 \times 60 = 210 \text{ horas}$$

representa el tiempo que se emplea la máquina B. La tabla indica que se dispone de 210 horas de este recurso, por lo que esta máquina está plenamente aprovechada (holgura = 0). *Asimismo, la máquina C se emplea todo el tiempo disponible.

*Las máquinas B y C se aprovechan plenamente

En el punto óptimo, la máquina A se emplea 120 de las 180 horas disponibles, por lo que tiene una holgura de 60 horas no aprovechadas. *La máquina D tampoco se aprovecha durante 60 horas.

*Las máquinas A y D no se emplean durante 60 horas.

	M A Q U I N A				NIVEL DE PRODUCCION OPTIMA
	A	B	C	D	
PRODUCTO 1	0	1	2	3	90
PRODUCTO 2	2	2	1	0	60
EMPLEO DE RECURSOS	$2 \times 60 = 120$	$90 + 2 \times 60 = 210$	$2 \times 90 + 60 = 240$	$3 \times 90 = 270$	
DISPONIBILIDAD DEL RECURSO	180	210	240	330	
HOLGURA	$180 - 120 = 60$	$210 - 210 = 0$	$240 - 240 = 0$	$330 - 270 = 60$	

Tabla 3.4.3 Aprovechamiento de recursos al nivel óptimo de producción.

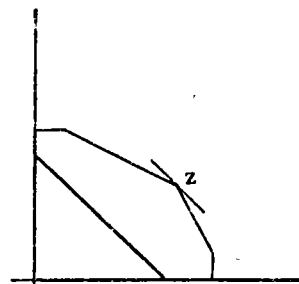
El número de piezas producidas al nivel óptimo es

$$\begin{aligned} z &= x_1 + x_2 \\ &= 90 + 60 \\ z &= 150 \end{aligned}$$

En las siguientes secciones se volverá a tratar el concepto de holgura que se acaba de introducir, al estudiar la solución analítica del problema de programación lineal.

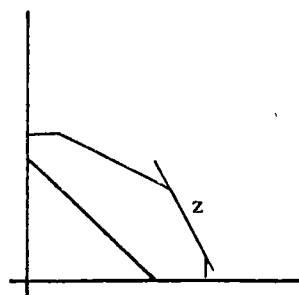
Cuando se traslada la recta correspondiente la función objetivo hasta el punto más alejado del origen que todavía tenga por lo menos un punto dentro de la región de soluciones factibles, pueden ocurrir dos casos. *Primeramente, como en la figura 3.4.7, la

*El punto óptimo puede ser único.



solución óptima se encuentra en uno de los vértices del polígono que define la región de soluciones factibles; en este caso la solución óptima es única. *En segundo lugar, puede ocurrir que la línea de la función objetivo coincida con uno de los lados del polígono; en este caso, todos los puntos de coincidencia constituyen soluciones factibles a la vez que óptimos, por lo que hay un número infinito de soluciones óptimas.

*Puede haber infinidad de puntos óptimos.



Con el objeto de ilustrar geoméricamente diversas interpretaciones económicas de un problema de programación lineal, y de estudiar posteriormente el problema dual, se introduce un ejemplo de dos variables y dos restricciones.

Ejemplo 3.4.4

Para producir dos posibles modelos de automóviles, el "Torbellino" y el "Ciclón", se requiere de K miles de pesos en materia prima y L miles de pesos en mano de obra, como muestra la tabla 3.4.4. Es decir, construir un Torbellino cuesta \$40 000 en materiales y \$75 000 en mano de obra. La empresa dispone men-

	K Materia Prima	L Mano de Obra	Beneficio
Torbellino	40	75	5
Ciclón	55	30	4
Disponibilidad	108 750	112 500	

Tabla 3.4.4 Datos del ejemplo 3.4.4 en miles de pesos.

sualmente de \$108 750 miles de pesos para pagar materia prima y de \$112 500 miles de pesos para pagar mano de obra. Si el beneficio de fabricar un automóvil Torbellino es de \$5 000 y para un Ciclón es de \$4 000, determine cuántos vehículos deben fabricarse al mes de cada tipo para maximizar la ganancia.

Solución:

x_1 = número de Torbellinos fabricados mensualmente
 x_2 = número de Ciclones fabricados mensualmente

Designando con:

Restricciones:

Materia prima:
 $40x_1 + 55x_2 \leq 108\,750$
 Mano de Obra:
 $75x_1 + 30x_2 \leq 112\,500$

Función objetivo:
 $5x_1 + 4x_2 = z$

*En la figura 3.4.8 aparece la solución geométrica del problema. Deben producirse por lo tanto:

*Solución: Deben producirse
 1 000 Torbellinos
 1 250 Ciclones

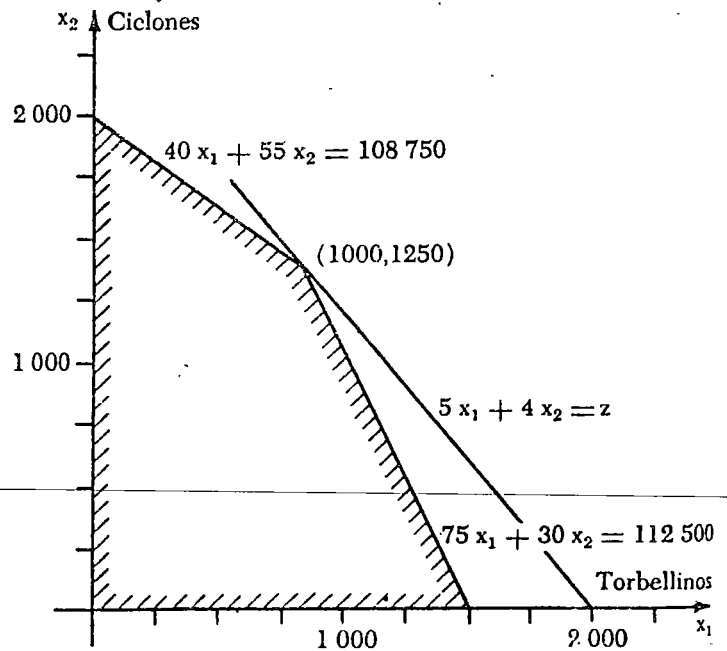


Fig. 3.4.8 Solución geométrica del ejemplo 3.4.4 en el espacio de actividades.

*La ganancia puede calcularse a partir de la función objetivo:

*El espacio mostrado en la figura 3.4.8 se conoce con el nombre de *espacio de actividades*, ya que sus dos ejes coordinados están asociados a las variables x_1 y x_2 , producción mensual (actividad) de uno y otro modelo de coche.

*El mismo problema puede resolverse empleando otro enfoque. Se establece el llamado *espacio de recursos* donde los ejes representan los recursos, en este caso capital para materia prima y mano de obra respectivamente, tal como muestra la figura 3.4.9.

*Ganancia:

$$1\ 000 \times \$5\ 000 + 1\ 250 \times \$4\ 000 = \$10 \text{ millones de pesos}$$

*Espacio de actividades: los ejes son las diversas actividades.

*Espacio de recursos: los ejes son los diversos recursos.

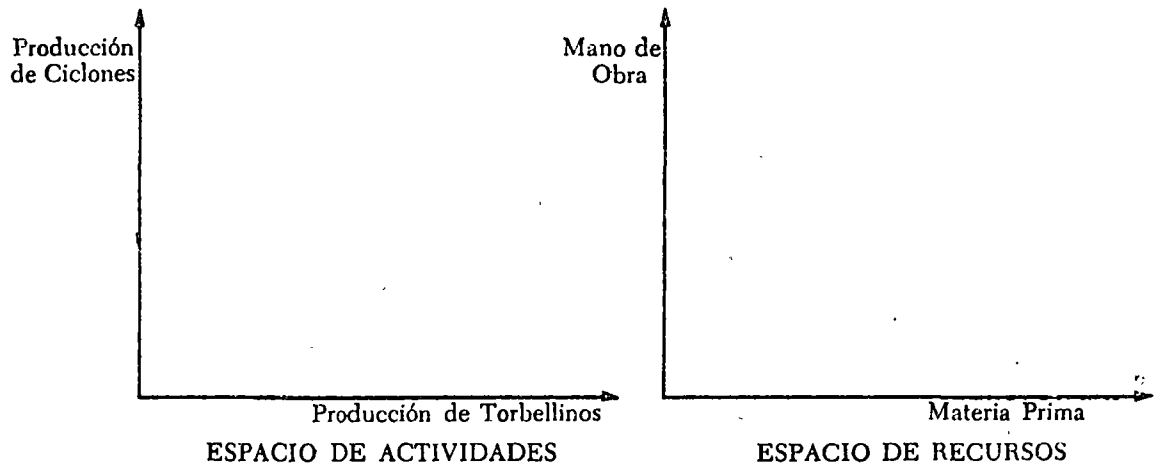


Fig. 3.4.9 Posibles espacios para representar el modelo de programación lineal.

Para analizar el problema empleando el espacio de recursos, debe hacerse el siguiente razonamiento. De acuerdo con la tabla 3.4.4, producir un Torbellino requiere de \$40 mil de materia prima y \$75 mil de mano de obra. *Para producir 800 Torbellinos se consumen:

de los recursos disponibles. *Ya que la producción de un Torbellino deja una ganancia de \$5 000, el beneficio de producir 800 Torbellinos es de

*El punto B de la figura 3.4.10 muestra el empleo de los recursos, materia prima y mano de obra para producir 800 Torbellinos

*Recursos empleados en producir 800 Torbellinos:

$$\begin{aligned} \$40 \times 800 &= \$32 \text{ millones de materia prima} \\ \$75 \times 800 &= \$60 \text{ millones de mano de obra} \end{aligned}$$

*Beneficio de producir 800 Torbellinos:

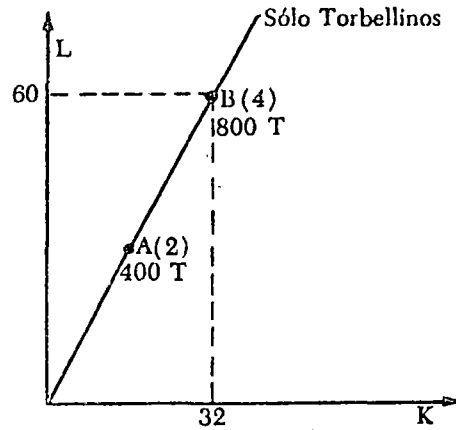
$$\$5 \times 800 = \$4 \text{ millones}$$

*Punto B:

Producción: 800 Torbellinos

Entre paréntesis, (4), se muestra el beneficio correspondiente en millones de pesos.

Empleo de recursos:
 \$32 millones en materia prima
 \$60 millones en mano de obra
 Beneficio: \$4 millones



Debido a la relación lineal entre producción y empleo de recursos, cualquier punto sobre la recta OB que va del origen al punto B, representa el empleo de recursos si se producen Torbellinos exclusivamente. El punto A, situado a media distancia sobre esta recta desde el origen, representa la producción de 400 Torbellinos.

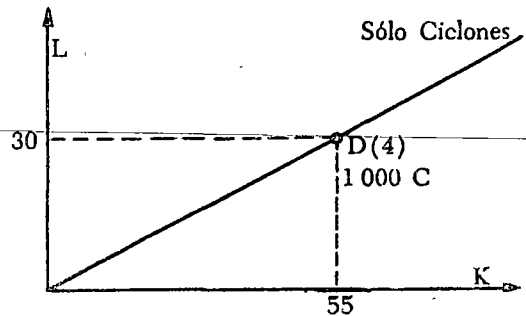
*Para continuar con el análisis debe buscarse ahora un punto D, que represente una producción exclusiva de Ciclones, y que deje el mismo beneficio que 800 Torbellinos. Debido a la relación de beneficios de \$5 000 a \$4 000 por unidad, se tiene que 1 000 "Ciclones" producen el mismo beneficio que 800 "Torbellinos", pues $1\,000 \times \$4 = \4 millones.

*Un nivel de producción de 1 000 Ciclones requiere del siguiente empleo de recursos:

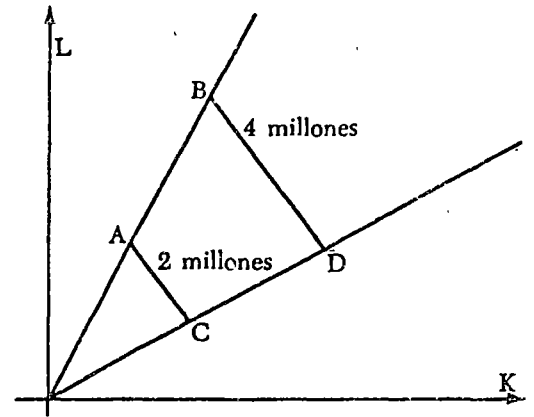
*Beneficio
 800 Torbellinos: \$4 millones
 1 000 Ciclones: \$4 millones

*Para producir 1 000 ciclones:
 $\$55 \times 1\,000 = \$55\,000$ miles en materia prima
 $\$30 \times 1\,000 = \$30\,000$ miles en mano de obra

El punto D se ilustra en la figura 3.4.10. La recta OD representa la producción exclusiva de Ciclones, con el empleo de recursos correspondiente.



Cualquier punto sobre la recta que une los puntos B y D, representa una producción combinada de Torbellinos y Ciclones que dejan un beneficio constante de \$4 millones. La recta AC, paralela a la BD y a la mitad de distancia del origen, representa la mitad del beneficio, o sea \$2 millones. En la figura 3.4.10 aparecen varias líneas de ganancia constante.

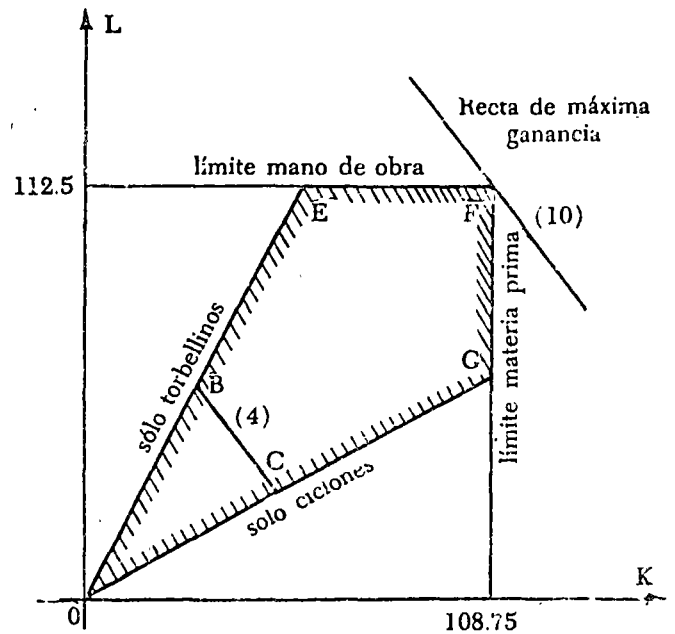


Finalmente, hay que contestar la siguiente pregunta:

¿Cómo puede determinarse la máxima ganancia empleando el espacio de recursos?

*Para ello, primero se localiza la zona de recursos posibles (soluciones factibles). Esta está limitada por las siguientes rectas:

*Se establece la zona de posibles usos de recursos y la línea de máximo beneficio.



OE (producción de Torbellinos exclusivamente), EF (límite de mano de obra disponible), FG (límite de materia prima disponible) y finalmente OG (producción de Ciclones exclusivamente). Para encontrar la línea de ganancia constante correspondiente a la solución óptima (máxima), es necesario trazar una recta de ganancia constante, paralela a la BC y lo más alejada posible del origen, que tenga por lo menos un punto dentro de la zona de soluciones factibles. Esta línea pasa por el punto F y corresponde a una ganancia de 10 millones, ya que esta recta está a una distancia del origen de 2.5 veces la recta BC, que corresponde a una ganancia de 4 millones.

Las coordenadas del punto F en términos de x_1 y x_2 se pueden determinar como sigue. Como sabemos que los recursos de materia prima y mano de obra se emplean plenamente en dicho punto, se puede encontrar el punto de intersección como la solución de las dos ecuaciones (igualdades)

$$\begin{aligned} 40x_1 + 55x_2 &= 108\,750 \\ 75x_1 + 30x_2 &= 112\,500 \end{aligned}$$

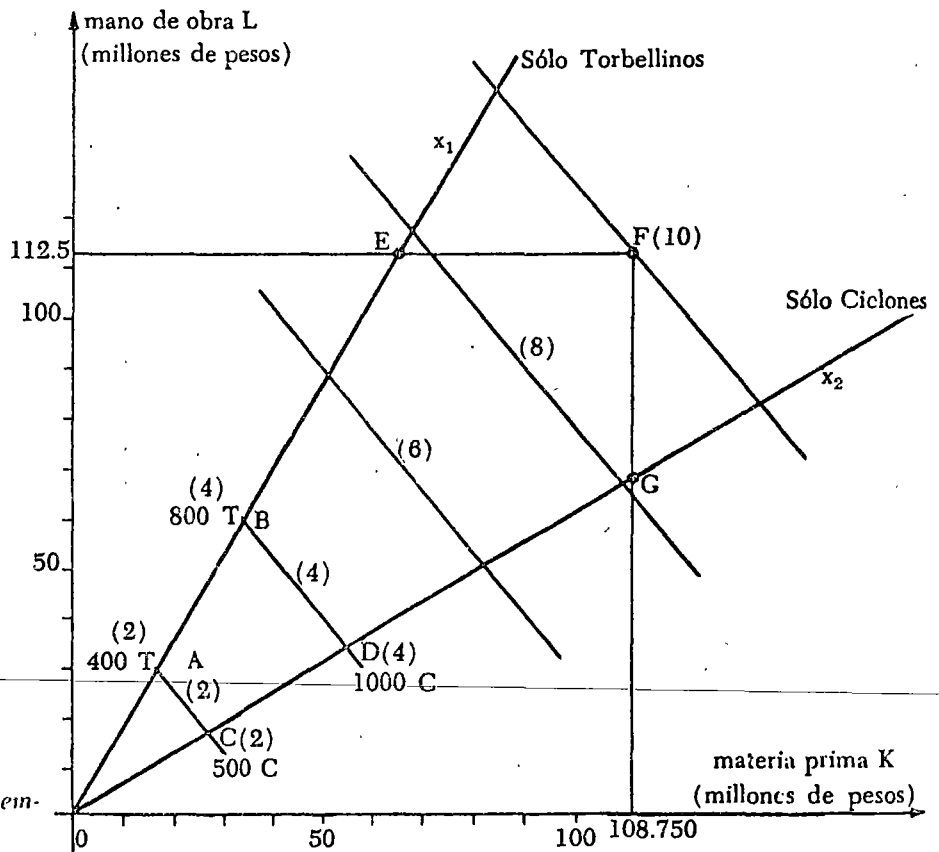


Fig. 3.4.10 Espacio de recursos del ejemplo 3.4.4 y solución.

La solución es

$$\begin{aligned} x_1 &= 1\ 000 \\ x_2 &= 1\ 250 \end{aligned}$$

que coincide con la obtenida anteriormente.

El análisis del espacio de recursos también permite contestar las siguientes preguntas:

¿Qué valor tiene disponer de una unidad más de mano de obra o una unidad más de materia prima?

Para contestar estas preguntas es necesario hacer el siguiente razonamiento:

Si el capital K asignado a materia prima aumenta de \$108.750 a \$118.750 millones, el óptimo cambia de F a H en la figura 3.4.11, y el beneficio de 10 a 10.5 millones. Es decir, un incremento en materia prima de $\Delta K = 10$ millones incrementa el beneficio en $\Delta Z = \$0.5$ millones.

El método analítico que se estudia en la sección 3.4.7 muestra que el valor exacto del incremento del beneficio es de \$0.513 millones.

*Si se va al banco a solicitar un financiamiento adicional para comprar más materia prima, podrá pagarse hasta:

*Máximo interés al capital para financiar materia prima:

$$\frac{\Delta Z}{\Delta K} = \frac{.513}{10} = 0.0513 = 5.13\% \quad (3.4.12)$$

de interés mensual. Esta cifra define el valor del recurso dado por el capital para materia prima.

*Un razonamiento similar, para el caso del capital para financiar el pago de mano de obra, se muestra en la figura 3.4.12. El valor de un financiamiento para pagar mano de obra adicional es de:

*Máximo interés al capital para financiar mano de obra:

$$\frac{\Delta Z}{\Delta L} = \frac{.393}{10} = 0.0393 = 3.93\% \quad (3.4.13)$$

que define el máximo interés mensual que se podrá pagar en un préstamo.

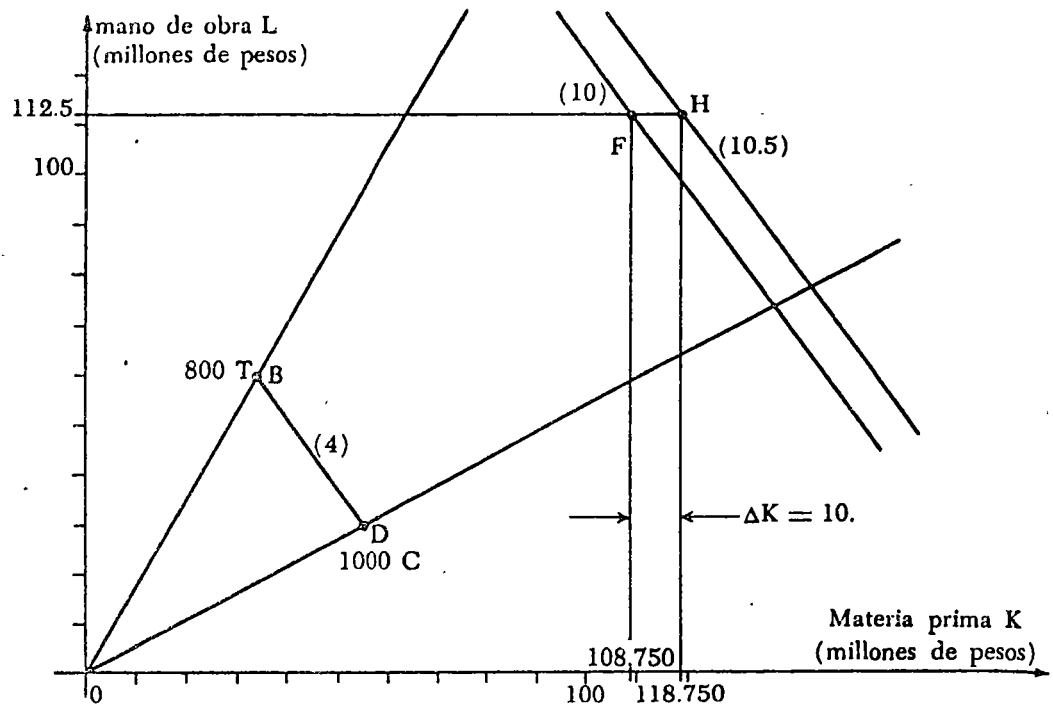


Fig. 3.4.11 Obtención gráfica del valor del recurso: capital para materia prima, del ejemplo 3.4.4.

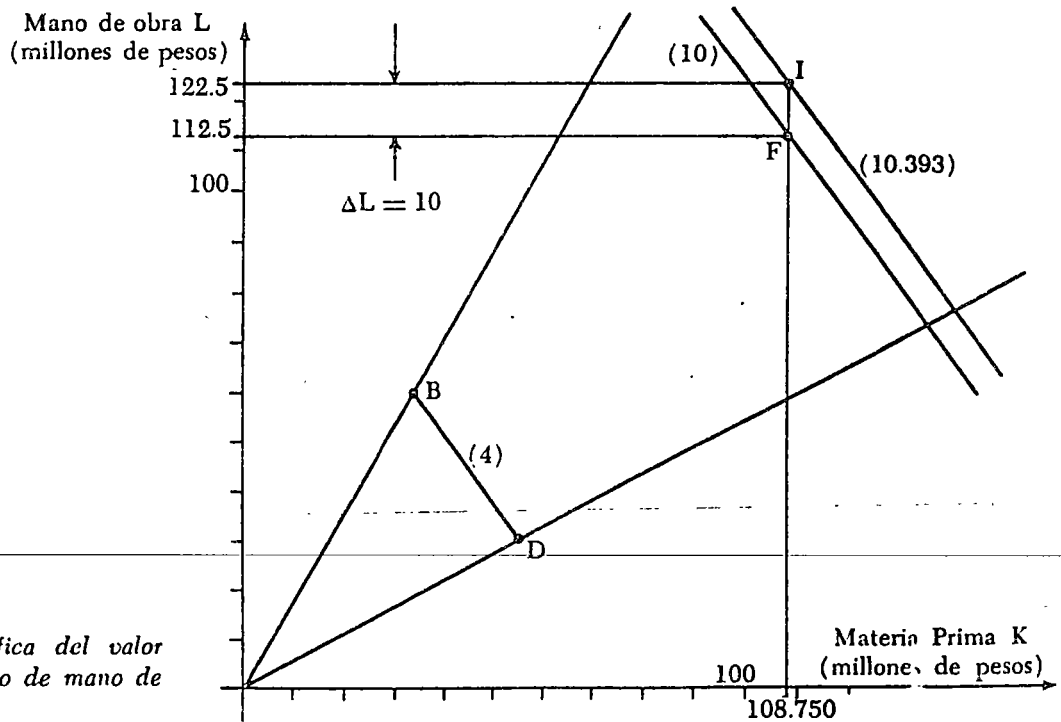


Fig. 3.4.12 Obtención gráfica del valor del recurso: capital para pago de mano de obra del ejemplo 3.4.4.

Como el rendimiento de una inversión de \$10 millones en materia prima incrementa el beneficio en \$0.513 millones, e igual incremento de \$10 millones en mano de obra produce sólo un incremento de \$0.393 millones en el beneficio, conviene invertir capital adicional para financiar la adquisición de materia prima.

En la sección 3.4.7, al estudiar la solución del problema dual, se resuelve este problema analíticamente. *Basta hacer notar que la programación lineal también permite resolver el problema del costo de los recursos, y no solo su aprovechamiento óptimo.

En la siguiente subsección se introduce el concepto de convexidad, importante propiedad que debe tener la zona de soluciones factibles en un problema de programación lineal.

3.4.5 Convexidad

*Las zonas de soluciones factibles de los ejemplos 3.4.1 y 3.4.4 de programación lineal, tienen la importante propiedad de ser convexas. Se define como *zona convexa* a una región que cumple con la siguiente característica:

°La programación lineal también permite determinar el costo de los recursos.

°Zona convexa:

Cualquier segmento de recta que una dos puntos sobre el perímetro de la región, debe encontrarse en el interior de la región o en su periferia.

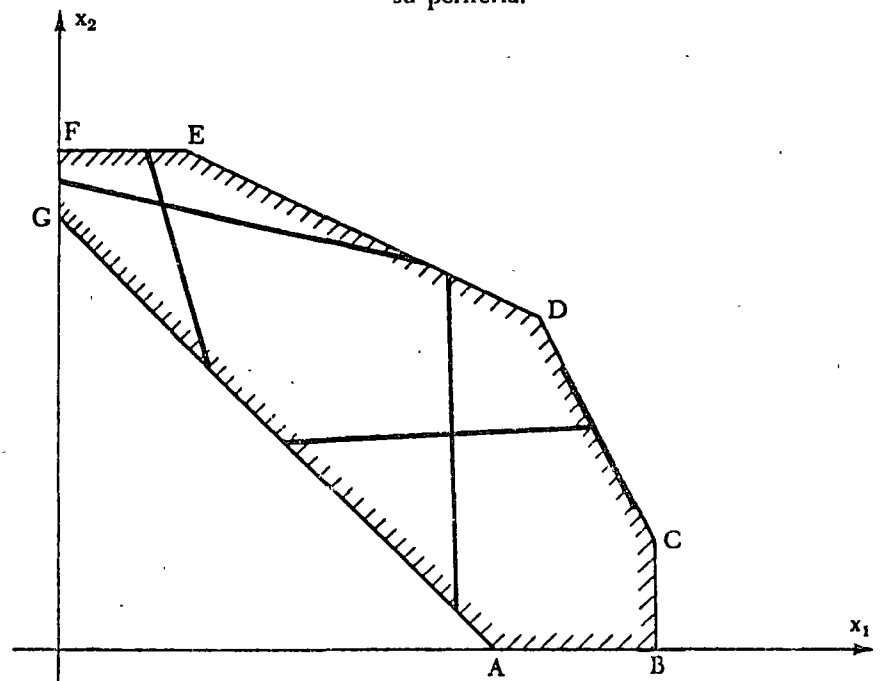


Fig. 3.4.13 Zona convexa.

La zona de soluciones factibles del ejemplo 3.4.1, mostrada en la figura 3.4.5, es efectivamente convexa. En la figura 3.4.13 se muestran varias rectas que unen puntos del perímetro de la zona. La totalidad del segmento de recta que los une se encuentra siempre dentro de la región o en su perímetro.

La figura 3.4.14 muestra diversas zonas que no son convexas, con posibles segmentos de recta fuera de la región.

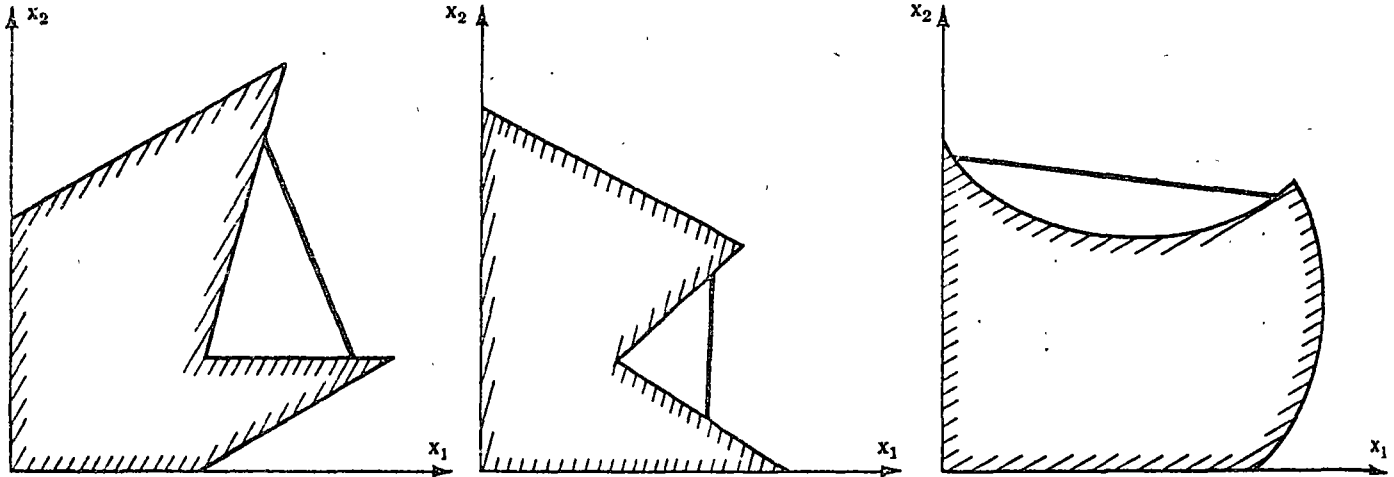


Fig. 3.4.14 Zonas no convexas.

Desde el punto de vista de la programación lineal, la convexidad de la zona definida por las restricciones del modelo del problema es importante, ya que bajo estas condiciones es válido el siguiente teorema:

En un problema de programación lineal con las restricciones definiendo una zona convexa, el punto óptimo se encuentra en la frontera de la región definida por las restricciones.

Este teorema permite buscar el óptimo de la función objetivo, mediante un método de búsqueda por gradiente, explorando el valor de la función objetivo únicamente en los vértices del polígono definido por las restricciones. A continuación se introducirá al lector a este método, conocido como método simplex.

3.4.6 El método simplex

En esta subsección se estudia el método simplex, que optimiza algebraicamente problemas de programación lineal. La persona que no desee estudiar este desarrollo matemático, puede pasar directamente a la subsección 3.4.7, que trata del problema dual.

El método simplex lo desarrollaron en 1947 el Dr. G. B. Dantzig y sus asociados para la Fuerza Aérea norteamericana.

Por el teorema expuesto al final de la subsección anterior, se sabe que la solución óptima de un problema de programación lineal se encuentra en algún punto de la periferia de la zona convexa de soluciones factibles. *El método simplex consiste en ir buscando el óptimo de vértice en vértice del polígono que determina la zona de soluciones factibles, hasta encontrarlo. Como el polígono tiene un número finito de vértices, se llegará a la solución óptima en un número finito de iteraciones. Se ilustrará este método resolviendo analíticamente el ejemplo 3.4.1.

Recuérdese el planteamiento de dicho problema. *La función objetivo por optimizar (maximizar) es

*Se determina el valor de la función en los vértices del polígono que limita la zona de soluciones factibles.

*Función objetivo

$$x_1 + x_2 = z \text{ (max)} \tag{3.4.1}$$

*sujeta a las restricciones:

*Restricciones:

$$x_2 \leq 90 \tag{3.4.2}$$

$$x_1 + 2x_2 \leq 210 \tag{3.4.3}$$

$$2x_1 + x_2 \leq 240 \tag{3.4.4}$$

$$x_1 \leq 110 \tag{3.4.5}$$

$$x_1 + x_2 \leq 80 \tag{3.4.6}$$

$$x_1 \geq 0 \tag{3.4.7}$$

$$x_2 \geq 0 \tag{3.4.8}$$

*Como las técnicas de programación matemática se emplean con sistemas de ecuaciones lineales simultáneas, es necesario transformar las desigualdades anteriores en igualdades. Esta transformación se logra añadiendo y sustrayendo variables auxiliares al sistema de desigualdades.

*En programación lineal, las desigualdades se transforman en igualdades.

*Si se tiene una desigualdad del tipo "menor o igual que", se puede convertir en una igualdad añadiendo al lado izquierdo una variable no negativa, llamada *variable de holgura*. Así las restricciones (3.4.2) a (3.4.5) se transforman en

*En las desigualdades del tipo "menor o igual que" se suman variables de holgura no negativas.

$$x_2 + x_3 = 90$$

$$x_1 + 2x_2 + x_4 = 210$$

$$2x_1 + x_2 + x_5 = 240$$

$$x_1 + x_6 = 110$$

al introducir una variable de holgura no negativa en cada ecuación.

*Para convertir una desigualdad del tipo "mayor o igual que" en

*En las desigualdades del tipo "mayor o

una igualdad, se resta del miembro izquierdo una variable de holgura no negativa. Así por ejemplo, la ecuación (3.4.6) se transforma en

*Las variables de holgura tienen la Interpretación económica de recursos no empleados. Por ejemplo para la máquina B, un valor mayor que cero de la variable de holgura de la ecuación $x_1 + 2x_2 + x_4 = 210$ correspondiente, nos indica que la máquina B no se emplea durante x_4 horas.

Observemos que en la igualdad anterior si

entonces

Como las variables de holgura deben ser no negativas, es necesario modificar la igualdad a

donde a_n es una variable no negativa (llamada *artificial*). Como esta variable se introduce simplemente con el propósito de tener una solución inicial

sencilla, debemos asegurar su ausencia (o sea $a_n = 0$) de la solución óptima. Esto se logra introduciendo a la variable con un gran coeficiente negativo (digamos -1000 para nuestro ejemplo) en la función objetivo:

*Resumiendo, en una restricción del tipo "igual a" o "mayor o igual que", debe sumarse una *variable artificial* no negativa.

*Para que las variables artificiales no aparezcan en la solución final, deben introducirse en la función objetivo con un gran coeficiente negativo en problemas de maximización, y con un gran coeficiente positivo en problemas de minimización.

Las ecuaciones de restricción de nuestro ejemplo convertidas

igual que" se restan variables de holgura no negativas.

$$x_1 + x_2 - x_7 = 80$$

*Las variables de holgura representan recursos no empleados.

$$x_1 = x_2 = 0$$

$$x_7 = -80$$

$$x_1 + x_2 - x_7 + a_8 = 80$$

$$x_1 = x_2 = 0$$

$$x_1 + x_2 - 1000 a_8 = Z$$

*En igualdades y en desigualdades del tipo "mayor o igual que" debe sumarse una variable artificial.

*Las variables artificiales deben aparecer en la función objetivo con un gran coeficiente negativo en problemas de maximización, y con un gran coeficiente positivo en problemas de minimización.

en igualdades con la inclusión de variables de holgura, y con la variable artificial, aparecen a continuación junto con la función objetivo.

$$\begin{array}{rcl}
 & x_2 + x_3 & = 90 \\
 x_1 + 2x_2 & + x_4 & = 210 \\
 2x_1 + x_2 & + x_5 & = 240 \\
 x_1 & + x_6 & = 110 \\
 x_1 + x_2 & - x_7 + a_8 & = 80 \\
 x_1 + x_2 & - 1000a_8 & = z
 \end{array}$$

Por facilidad, emplearemos notación matricial para escribir este sistema de ecuaciones de restricción como:

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ a_8 \end{matrix} = \begin{matrix} 90 \\ 210 \\ 240 \\ 110 \\ 80 \end{matrix} \quad (3.4.14)$$

y la función objetivo como:

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1000 \end{bmatrix} \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ a_8 \end{matrix} = z \quad (3.4.15)$$

*Un arreglo de números o variables en columnas y renglones se conoce con el nombre de *matriz*. Si una matriz tiene más de un renglón y una columna, se la designa con letra romana mayúscula. Así en la ecuación (3.4.14) se tiene la matriz A dada por

*Un arreglo de números o variables en columnas y renglones es una matriz.

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} = A$$

Cada número o variable de una matriz se identifica por el renglón y la columna a la que pertenece. Así por ejemplo, el elemento

$$\begin{bmatrix} 0 & 1 & \dots \\ 1 & 2 & \dots \\ 2 & \textcircled{1} & \dots \\ \cdot & \cdot & \\ \cdot & \cdot & \\ \cdot & \cdot & \end{bmatrix}$$

pertenece al tercer renglón y a la segunda columna, y se representa con

$$a_{32} = 1$$

primer subíndice = renglón
segundo subíndice = columna

Los dos subíndices indican el número del renglón y de la columna, respectivamente.

Una matriz de elementos a_{ij} se puede representar en forma compacta como

$$A = [a_{ij}]$$

donde el índice i varía desde 1 hasta el número de renglones, y el j desde 1 hasta el número de columnas.

*Existen matrices de una sola columna o de un solo renglón, también llamados *vectores*.

*Un vector es una matriz de una sola columna o de un solo renglón

Por ejemplo

$$\begin{bmatrix} 90 \\ 210 \\ 240 \\ 110 \\ 80 \end{bmatrix} = \text{matriz de una sola columna o vector}$$

es un vector columna. Suelen usarse letras minúsculas para representar vectores.

*Se dice que dos matrices A y B son iguales, si tienen igual número de columnas y renglones, y los elementos que ocupan posiciones correspondientes en ambas son iguales. Es decir,

$$^{\circ}A = [a_{ij}] \quad B = [b_{ij}] \quad A = B \text{ si:}$$

$$a_{ij} = b_{ij} \text{ para toda } i, \text{ para toda } j$$

Para entender las expresiones (3.4.14) y (3.4.15) es necesario definir las operaciones de adición o sustracción y multiplicación de matrices. No está definida la operación de división de matrices.

*Dos matrices se pueden sumar únicamente si tienen igual número de renglones y columnas. Sean

*Suma de matrices

y

dos matrices con el mismo número de renglones y de columnas. Su suma

se obtiene sumando los términos que ocupan la misma posición, es decir

donde

La notación

indica que se trata de una matriz con:

*Para multiplicar dos matrices, el número de columnas de la primera debe coincidir con el número de renglones de la segunda; si no se cumple este requisito, las matrices no se pueden multiplicar. *Matrices que cumplen con la condición anterior se conocen con el nombre de *conformables*. Usando la notación anterior se tiene que las matrices:

solo pueden multiplicarse si:

*El producto de dos matrices conformables A y B es una tercera matriz C, cuyos elementos están definidos por la fórmula

La matriz producto C tiene n renglones y q columnas.

*Se conoce con el nombre de *transpuesta* A^T de una matriz, A, a la matriz que se obtiene intercambiando los renglones por las columnas.

$$A = [a_{ij}]$$

$$B = [b_{ij}]$$

$$A + B = C$$

$$C = [c_{ij}]$$

$$c_{ij} = a_{ij} + b_{ij}$$

$$A_{n,m}$$

n renglones
m columnas

*Multiplicación de matrices

*Deben ser conformables: el número de columnas de la primera matriz debe ser igual al número de renglones de la segunda.

$$A_{n,m} B_{p,q}$$

$$m = p$$

$$C = A B$$

$$c_{ij} = \sum_{k=1}^m a_{ik} b_{kj} \quad (3.4.16)$$

*Los renglones de la matriz A son iguales a las columnas de la matriz transpuesta A^T .

Es decir, si

$$A = [a_{ij}]$$

$$A^T = [a_{ji}]$$

Antes de continuar, un ejemplo servirá para ilustrar los conceptos anteriores.

Ejemplo 3.4.5

Sean las matrices

$$A = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & -1 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 10 & 1 & -4 \end{bmatrix}$$

$$C = \begin{bmatrix} 4 & -2 \\ 6 & -1 \\ -3 & 4 \end{bmatrix}$$

Realice las siguientes operaciones:

A + B
A - B
A + C
A B
A C
C A
A^T

Solución:

$$A + B = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & -1 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ 10 & 1 & -4 \end{bmatrix} = \begin{bmatrix} 1+1 & 3+1 & 2+1 \\ 2+10 & 1+1 & -1-4 \end{bmatrix} = \begin{bmatrix} 2 & 4 & 3 \\ 12 & 2 & -5 \end{bmatrix}$$

$$A - B = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & -1 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 \\ 10 & 1 & -4 \end{bmatrix} = \begin{bmatrix} 1-1 & 3-1 & 2-1 \\ 2-10 & 1-1 & -1-(-4) \end{bmatrix} = \begin{bmatrix} 0 & 2 & 1 \\ -8 & 0 & 3 \end{bmatrix}$$

A + C La suma no puede realizarse porque no tienen igual número de columnas y renglones las matrices.

A B El producto no puede realizarse porque el número de columnas de A no coincide con el número de renglones de B.

$$AC = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & -1 \end{bmatrix} \begin{bmatrix} 4 & -2 \\ 6 & -1 \\ -3 & 4 \end{bmatrix} = \begin{bmatrix} 1 \times 4 + 3 \times 6 + 2 \times (-3) & 1 \times (-2) + 3 \times (-1) + 2 \times 4 \\ 2 \times 4 + 1 \times 6 + (-1) \times (-3) & 2 \times (-2) + 1 \times (-1) + (-1) \times 4 \end{bmatrix} = \begin{bmatrix} 16 & 3 \\ 17 & -9 \end{bmatrix}$$

$$CA = \begin{bmatrix} 4 & -2 \\ 6 & -1 \\ -3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 4 \times 1 + (-2) \times 2 & 4 \times 3 + (-2) \times 1 & 4 \times 2 + (-2) \times (-1) \\ 6 \times 1 + (-1) \times 2 & 6 \times 3 + (-1) \times 1 & 6 \times 2 + (-1) \times (-1) \\ (-3) \times 3 + 4 \times 1 & (-3) \times 1 + 4 \times 2 & (-3) \times 2 + 4 \times (-1) \end{bmatrix} = \begin{bmatrix} 0 & 10 & 10 \\ 4 & 17 & 13 \\ 5 & -5 & -10 \end{bmatrix}$$

$$A^T = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & -1 \end{bmatrix}$$

Debe observarse que, en un producto matricial, el orden de los factores *si* altera el producto. El producto AC es diferente del CA; ni siquiera tienen el mismo número de renglones y columnas.

Con este conocimiento introductorio sobre operaciones con matrices, es posible llegar a las relaciones (3.4.14) y (3.4.15) a partir del sistema de ecuaciones original.

Definiendo los vectores

$$b = \begin{bmatrix} 90 \\ 210 \\ 240 \\ 110 \\ 80 \end{bmatrix}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

$$c = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \ 000 \end{bmatrix}$$

y empleando la matriz A definida anteriormente, las expresiones (3.4.14) y (3.4.15) se pueden escribir compactamente como:

$$Ax = b \quad (3.4.17)$$

$$c^T x = z \quad (3.4.18)$$

*Además de las condiciones anteriores, debe incluirse la condición de no negatividad de las variables, que expresadas matricialmente toma la forma:

*Condición de no negatividad

$$x \geq 0$$

*El vector nulo o cero 0 de la ecuación anterior es un vector cuyas n componentes son nulas, es decir:

*Vector nulo 0

$$0 = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

La desigualdad vectorial

$$y \geq x$$

implica que

$$y_i \geq x_i \quad \text{para toda } i$$

Es decir, las componentes del primer vector son iguales o mayores que las correspondientes componentes del segundo.

*La ecuación matricial (3.4.17) representa un sistema de m ecuaciones en n incógnitas, donde m es el número de restricciones y n el número de variables, incluyendo de holgura y artificiales. En el ejemplo se tiene:

* m restricciones n variables (incluyendo de holgura y artificiales)

$$m = 5$$

$$n = 8$$

y el sistema de 5 ecuaciones en 8 incógnitas es:

$$\begin{array}{rcl} 0x_1 + 1x_2 + 1x_3 + 0x_4 + 0x_5 + 0x_6 + & 0x_7 + 0a_8 & = 90 \\ 1x_1 + 2x_2 + 0x_3 + 1x_4 + 0x_5 + 0x_6 + & 0x_7 + 0a_8 & = 210 \\ 2x_1 + 1x_2 + 0x_3 + 0x_4 + 1x_5 + 0x_6 + & 0x_7 + 0a_8 & = 240 \\ 1x_1 + 0x_2 + 0x_3 + 0x_4 + 0x_5 + 1x_6 + & 0x_7 + 0a_8 & = 110 \\ 1x_1 + 1x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_6 + (-1)x_7 + 1a_8 & = & 86 \end{array}$$

Si se introduce la siguiente notación:

$e_k =$ vector unitario, cuya k -ésima componente vale uno, y las demás valen cero

el sistema de ecuaciones (3.4.17) puede escribirse como sigue:

$$\begin{array}{r}
 \begin{array}{c} 0 \\ 1 \\ 2 \\ 1 \\ 1 \end{array} x_1 + \begin{array}{c} 1 \\ 2 \\ 0 \\ 1 \end{array} x_2 + e_1 x_3 + e_2 x_4 + e_3 x_5 + e_4 x_6 + \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{array} x_7 + e_5 a_8 = \begin{array}{c} 90 \\ 210 \\ 240 \\ 110 \\ 80 \end{array}
 \end{array}$$

Nótese que en el sistema hay cinco vectores unitarios, e_1, e_2, e_3, e_4 y e_5 , es decir tantos como ecuaciones. Reagrupando las variables puede escribirse el sistema anterior de la siguiente forma:

$$e_1 x_3 + e_2 x_4 + e_3 x_5 + e_4 x_6 + e_5 a_8 + \begin{array}{c} 0 \\ 1 \\ 2 \\ 1 \\ 1 \end{array} x_1 + \begin{array}{c} 1 \\ 2 \\ 0 \\ 1 \end{array} x_2 + \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{array} x_7 = \begin{array}{c} 90 \\ 210 \\ 240 \\ 110 \\ 80 \end{array}$$

De este último sistema se obtiene de inmediato:

$$\begin{aligned}
 x_3 &= -0x_1 - 1x_2 - 0x_7 + 90 \\
 x_4 &= -1x_1 - 2x_2 - 0x_7 + 210 \\
 x_5 &= -2x_1 - 1x_2 - 0x_7 + 240 \\
 x_6 &= -1x_1 - 0x_2 - 0x_7 + 110 \\
 a_8 &= -1x_1 - 1x_2 - (-1)x_7 + 80
 \end{aligned}$$

Estas ecuaciones indican que los valores de las variables x_3, x_4, x_5, x_6 y a_8 dependen de los valores de x_1, x_2 y x_7 .

Para emplear el método simplex para resolver el problema representado por las ecuaciones (3.4.17) y (3.4.18), debe empezarse por determinar una solución factible. *Si en el sistema de ecuaciones anteriores, que se equivalente al (3.4.17), igualamos x_1, x_2 y x_7 a cero, se obtiene:

*Si $x_1 = x_2 = x_7 = 0$

$$\begin{aligned}
 x_3 &= 90 \\
 x_4 &= 210 \\
 x_5 &= 240 \\
 x_6 &= 110 \\
 a_8 &= 80
 \end{aligned}$$

Estos valores y

$$\begin{aligned}
 x_1 &= 0 \\
 x_2 &= 0 \\
 x_7 &= 0
 \end{aligned}$$

constituyen una solución factible.

Debe observarse que esta solución factible con $x_1 = x_2 = 0$ no se encuentra dentro de la región de soluciones factibles original de la figura 3.4.5. Esto se debe a la introducción de la variable artificial a_8 .

A las variables x_1, x_2 y x_7 puede dárseles otro valor, pero al escoger un valor diferente del nulo para estas variables, puede resultar que alguna de las restantes sea negativa. Si, por ejemplo, se escoge

$$x_1 = x_2 = x_7 = 100$$

entonces se tiene

$$x_3 = -100 + 90 = -10$$

violándose ya una condición de no negatividad de las variables.

*De la exposición anterior se puede concluir que, para facilitar el cálculo de una solución factible, basta identificar las columnas no unitarias y dar a las variables asociadas a éstas el valor nulo, obteniéndose de inmediato una posible solución factible.

*Para obtener una solución factible, basta igualar a cero las variables no identificadas con columnas unitarias.

*En el método simplex de solución de problemas de programación lineal, conviene formar la llamada *tabla simplex* o *tableau*, que se muestra a continuación para el problema que se está trabajando. En ella se pueden realizar las operaciones anteriores.

*La tabla simplex o tableau se forma con los coeficientes de las ecuaciones de restricción y de la función objetivo.

		n columnas; n = número de variables								columna de restricciones
		x_1	x_2	x_3	x_4	x_5	x_6	x_7	a_8	b_j
		0	1	1	0	0	0	0	0	90
m renglones		1	2	0	1	0	0	0	0	210
m = número de restricciones		2	1	0	0	1	0	0	0	240
		1	0	0	0	0	1	0	0	110
		1	1	0	0	0	0	-1	1	80
	renglón de la función objetivo (coeficientes de costos)	1	1	0	0	0	0	0	-1 000	

(3.4.19)

En la tabla, los primeros m renglones constan de los coeficientes de las ecuaciones de restricción (3.4.14), con los valores del lado derecho de las ecuaciones de restricción en una columna de restricciones. En el último renglón se ponen los coeficientes correspondientes de la función objetivo, llamados *coeficientes de costos*.

*Para encontrar una solución factible, se analiza la parte de la tabla simplex correspondiente a los coeficientes de las ecuacio-

*Se identifican las columnas unitarias

nes de restricción, y se identifican las columnas unitarias. *El k-ésimo renglón se identifica con la variable correspondiente al vector unitario k-ésimo e_k .

*Al renglón k-ésimo se asocia la variable de e_k .

El primer renglón se identifica con x_3 porque el vector unitario e_1 está localizado en la columna de la variable x_3 . Las identificaciones se muestran en la tabla (3.4.20).

renglón	variable
1	x_3
2	x_4
3	x_5
4	x_6
5	a_8

Identificación de renglones	$e_1 \ e_2 \ e_3 \ e_4 \ e_5$								restricciones
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	a_8	b_j
x_3	0	1	1	0	0	0	0	0	90
x_4	1	2	0	1	0	0	0	0	210
x_5	2	1	0	0	1	0	0	0	240
x_6	1	0	0	0	0	1	0	0	110
a_8	1	1	0	0	0	0	-1	1	80
	1	1	0	0	0	0	0	-1	0

(3.4.20)

*De la tabla (3.4.20), se identifican las variables nulas de la solución factible como las variables correspondientes a las columnas *no* unitarias:

*Variables nulas de las columnas no unitarias.

$$x_1 = x_2 = x_7 = 0$$

*De la misma tabla se leen los valores de las variables no nulas. La variable asociada al j-ésimo renglón es igual a restricción b_j :

*Valor de las variables no nulas de la base:

$$\begin{aligned} x_3 &= 90 \\ x_4 &= 210 \\ x_5 &= 240 \\ x_6 &= 110 \\ a_8 &= 80 \end{aligned}$$

La exposición analítica realizada anteriormente justifica las reglas dadas.

*Las variables de valor no nulo reciben el nombre de *variables de la base*. Por razones de cómputo, conviene escoger aquellas asociadas a los vectores unitarios, como se hace en el método simplex. Los renglones de la tabla quedan identificados entonces con las variables de la base.

*Variables no nulas \equiv
Variables de la base \equiv
Variables asociadas a los vectores unitarios

*Obsérvese que la función objetivo (3.4.15) está expresada en función de variables nulas y posiblemente artificiales:

*La función objetivo está expresada en función de las variables nulas y artificiales:

*A la tabla (3.4.20) se le agrega una columna, al principio o al final, con los coeficientes con que aparecen en la función objetivo las variables de la base. *Como la función objetivo es $x_1 + x_2 - 1000 a_8$, y las variables de la base son x_3, x_4, x_5, x_6 y a_8 , los valores de la columna son:

*Los valores de esta columna reciben el nombre de *beneficios unitarios*.

*La primera tabla completa del método simplex es:

beneficio unitario	base	x_1	x_2	x_3	x_4	x_5	x_6	x_7	a_8	restricciones b_j
0	x_3	0	1	1	0	0	0	0	0	90
0	x_4	1	2	0	1	0	0	0	0	210
0	x_5	2	1	0	0	1	0	0	0	240
0	x_6	1	0	0	0	0	1	0	0	110
-1000	a_8	1	1	0	0	0	0	1	1	80 ←
Coeficientes de costos c_j		1	1	0	0	0	0	0	-1000	
Evaluadores netos		-1001	-1001						+1000	

(3.4.21)

*Para los valores de la primera solución factible, la función objetivo vale:

$$x_1 + x_2 - 1000 a_8 = z$$

*Primera iteración

Base: x_3, x_4, x_5, x_6, a_8

Función objetivo: $x_1 + x_2 - 1000 a_8 = z$

Coefficiente de la función objetivo en la base:

0, 0, 0, 0, -1000

*Estos coeficientes son beneficios unitarios.

*Tabla simplex primera iteración:

*Para la primera solución factible, la función objetivo vale:

$$\begin{aligned} z &= x_1 + x_2 - 1000 a_8 \\ &= 0 + 0 - 1000 \times 80 \\ z &= -80\,000 \end{aligned}$$

Obsérvese que este valor es igual a la suma de los productos de los términos de la columna de beneficios unitarios, con la columna de restricciones de la tabla (3.4.21), es decir

primera columna	última columna	producto
beneficio unitario	restricciones b_j	
0	90	$0 \times 90 = 0$
0	210	$0 \times 210 = 0$
0	240	$0 \times 240 = 0$
0	110	$0 \times 110 = 0$
-1000	80	$-1000 \times 80 = -80\,000$
Suma:		-80 000

Recordemos la interpretación de las variables. La función objetivo por maximizar z era la ganancia en la producción de los artículos 1 y 2. *El resultado $z = -80\,000$ indica que se pierden \$80 000 en dicha fabricación, por lo que no puede esperarse que este valor de z sea óptimo.

Como

y x_1 y x_2 representan el número de artículos 1 y 2 producidos, no se está fabricando nada. Además

* ¡Debe aumentar z !

$$x_1 = x_2 = x_7 = 0$$

$$\begin{aligned} x_3 &= 90 \\ x_4 &= 210 \\ x_5 &= 240 \\ x_6 &= 110 \\ a_8 &= 80 \end{aligned}$$

*Para interpretar estos valores, recordemos que la ecuación (3.4.2)

* Restricción que impone la máquina A

$$\begin{aligned} 2x_2 &\leq 180 \\ x_2 &\leq 90 \end{aligned} \tag{3.4.2}$$

representa la restricción que impone la disponibilidad de la máquina A en el proceso productivo. Es decir, el número de horas $2x_2$ que se emplea la máquina A no debe ser mayor de 180. Esta restricción se convirtió en una igualdad introduciendo la variable de holgura x_3 , es decir

$$x_2 + x_3 = 90$$

o sea

$$2x_2 + 2x_3 = 180$$

La máquina A se emplea durante

$$2x_2 = 0$$

horas, y tiene una holgura de

$$2x_3 = 180$$

horas disponibles. La máquina A no está plenamente aprovechada; lo que es más, ¡ni siquiera se está empleando!

Las variables de holgura restantes se pueden interpretar en forma análoga.

Para continuar con el método simplex, debe hacerse la siguiente pregunta:

*Antes de contestar esta pregunta, deben interpretarse los coeficientes de la tabla simplex, correspondientes a las variables que no están en la base, es decir a las variables no asociadas a las columnas unitarias. De la tabla (3.4.21) se tiene que

¿Si una de las variables nulas aumenta en una unidad, en cuánto se modifica la función objetivo?

*Interpretación de los coeficientes de la tabla simplex.

	[*] x ₁	[*] x ₂	x ₃	x ₄	x ₅	x ₆	[*] x ₇	[*] x ₈
base								
x ₃	0	1	1	0	0	0	0	0
x ₄	1	2	0	1	0	0	0	0
x ₅	2	1	0	0	1	0	0	0
x ₆	1	0	0	0	0	1	0	0
a ₈	1	1	0	0	0	0	-1	1

donde las columnas marcadas con un asterisco indican las variables que no están en la base:

base	x ₁	x ₂	x ₇	
x ₃	0	1	0	
x ₄	1	2	0	(3.4.22)
x ₅	2	1	0	
x ₆	1	0	0	
a ₈	1	1	-1	

Recuérdese ahora la formulación matemática de las restricciones en esta etapa.

$$\begin{aligned}
 & \boxed{1} x_2 + x_3 = 90 \\
 \triangle 2 x_1 + & 2 x_2 + x_4 = 210 \\
 \triangle 2 x_1 + & x_2 + x_5 = 240 \\
 & x_1 + x_6 = 110 \\
 & x_1 + x_2 - x_7 + a_8 = 80 \\
 & x_1 + x_2 - 1000a_8 = z
 \end{aligned}$$

y la solución:

$$\begin{aligned}
 x_3 &= 90 \\
 x_4 &= 210 \\
 x_5 &= 240 \\
 x_6 &= 110 \\
 a_8 &= 80 \\
 x_1 = x_2 = x_7 &= 0
 \end{aligned}$$

En esta fase

$$\begin{aligned}
 x_2 &= 0 \\
 x_3 &= 90
 \end{aligned}$$

De la primera ecuación de restricción

$$\boxed{1} x_2 + x_3 = 90$$

se observa que, si x_2 aumenta en una unidad, entonces x_3 disminuye en el valor del coeficiente de x_2 , o sea

$$\begin{aligned} \text{si } x_2 \nearrow \text{ en } 1 \\ x_3 \searrow \text{ en } \boxed{1} \times 1 = 1 \end{aligned}$$

Es decir, una unidad de x_2 sustituye una ($\boxed{1}$) unidad de x_3 .

El coeficiente

$$a_{12} = \boxed{1}$$

*representa la *razón de sustitución* entre la variable x_2 y la variable de la base asociada a ese renglón, es decir x_3 :

°Razón de sustitución:

	┌──────────┐		
base	x_1	x_2	x_7
x_3	0	$\boxed{1}$	0

En forma similar, el coeficiente

$$a_{31} = \triangle 2$$

representa la razón de sustitución entre la variable x_1 y la variable de la base asociada a ese renglón, es decir x_3 :

	┌──────────┐		
base	x_1	x_2	x_7
x_3	$\triangle 2$	1	0

*Si x_1 aumenta en una unidad, x_3 disminuye en dos unidades. En otras palabras, una unidad de x_1 sustituye a dos unidades de x_3 .

$$\begin{aligned} \text{°Si } x_1 \nearrow \text{ en } 1 \\ x_3 \searrow \text{ en } \triangle 2 \times 1 = 2 \end{aligned}$$

Los demás coeficientes asociados a las columnas no unitarias mencionadas tienen una interpretación semejante. El coeficiente negativo

$$a_{37} = -1$$

indica que, *si x_7 aumenta en una unidad, entonces a_8 "disminuye" en $(-1) \times 1 = -1$, es decir también aumenta. Ésto se puede observar claramente de la correspondiente ecuación de restricción:

$$\begin{aligned} \text{°Si } x_7 \nearrow \text{ en } 1 \\ a_8 \searrow \text{ en } (-1) \times 1 = -1, \text{ es decir} \\ a_8 \nearrow 1 \end{aligned}$$

$$x_1 + x_2 - x_7 + a_8 = 80$$

*Empleando la interpretación anterior, podemos ahora contestar a la pregunta planteada previamente: ¿en cuanto "mejora" la función objetivo, si alguna de las variables no base (de valor nulo), incrementa su valor en una unidad?

*Empecemos estudiando el efecto sobre la función objetivo, de un aumento en x_1 .

De los cálculos anteriores se sabe que el valor actual de la función objetivo es de $-80\ 000$.

*¿Cuánto cambia la función objetivo, si una variable nula aumenta su valor?

*¿Si $x_1 \nearrow$ en 1, en cuánto cambia la función objetivo?

base	primera columna	última columna	producto
x_3	0	90	0
x_4	0	210	0
x_5	0	240	0
x_6	0	110	0
a_8	-1000	80	-80 000
		Suma	-80 000

Ai aumentar x_1 , la función objetivo

$$Z = x_1 + x_2 - 1\ 000 a_8$$

*que en este momento vale $-80\ 000$, se modifica debido a dos factores:

*La función objetivo se modifica por

- a) el incremento en x_1
- b) el cambio en las variables de la base producido por el aumento en x_1

El primer factor modifica a la función objetivo directamente, y el segundo indirectamente.

Al aumentar x_1 en una unidad, la función objetivo

$$z = \textcircled{1} \times x_1 + x_2 - 1\ 000 a_8$$

aumenta en

$$\textcircled{1} \times x_1 = 1$$

además de posibles cambios indirectos debidos a modificaciones en las variables de la base. El $\textcircled{1}$ es precisamente el coeficiente que se encuentra en la tabla (3.4.21) en el renglón de coeficientes de costos, en la columna x_1 .

x_1	x_2	x_3	x_4	x_5	x_6	a_8	b_j
$\textcircled{1}$	1	0	0	0	0	0	-1000

Determinemos ahora los cambios en las variables de la base de-

bidos a un cambio en x_1 , y la forma en la que afectan a la función objetivo.

*Si x_1 aumenta en una unidad, las variables de la base cambian de acuerdo con el valor de las relaciones de sustitución de la tabla (3.4.21):

*Relaciones de sustitución entre x_1 y las variables de la base:

beneficio			
unitario	base	x_1	
0	x_3	0	
0	x_4	1	
0	x_5	2	(3.4.23)
0	x_6	1	
-1000	a_8	1	

Esta tabla indica que:

Si $x_1 \nearrow$ en 1: $x_3 \searrow$ en 0
 $x_4 \searrow$ en 1
 $x_5 \searrow$ en 2
 $x_6 \searrow$ en 1
 $a_8 \searrow$ en 1

El efecto indirecto sobre la función objetivo si x_1 cambia en una unidad, se obtiene multiplicando los elementos correspondientes de la columna de beneficios unitarios y de la columna x_1 , y sumando. A este efecto indirecto se resta el efecto directo, pues el primero disminuye mientras que el segundo se incrementa, al aumentar x_1 .

	beneficio			
	unitario	base	x_1	producto
	0	x_3	0	0×0
	0	x_4	1	0×1
	0	x_5	2	0×2
	0	x_6	1	0×1
	-1000	a_8	1	-1000×1
coeficientes de costos			①	
	$0 \times 0 + 0 \times 1 + 0 \times 2 + 0 \times 1 - 1000 \times 1$			$- ① \times 1 = -1001$
	beneficio indirecto			beneficio directo
				evaluador neto

Este efecto neto de -1001, o cambio en el beneficio o sea en la función objetivo, se muestra en el renglón de *evaluadores netos de la tabla (3.4.21).

*Los evaluadores netos son los beneficios.

El efecto sobre la función objetivo de incrementos en las otras

dos variables x_2 y x_7 que no estaban en la base, se estudia en forma análoga.

Si x_2 cambia de 0 a 1, el cambio en el beneficio o función objetivo es:

beneficio unitario	base	x_2
0	x_3	1
0	x_4	2
0	x_5	1
0	x_6	0
-1000	a_8	1

coeficientes de costos evaluador neto

$$0 \times 1 + 0 \times 2 + 0 \times 1 + 0 \times 0 - 1000 \times 1 - 1 \times 1 = -1001$$

beneficio indirecto beneficio directo evaluador neto

Es decir, el evaluador neto correspondiente a un cambio unitario en x_2 es de -1001 .

Finalmente, para la última variable que no está en la base se tiene:

beneficio unitario	base	x_7
0	x_3	0
0	x_4	0
0	x_5	0
0	x_6	0
-1000	a_8	-1

coeficiente de costos evaluador neto

$$0 \times 0 + 0 \times 0 + 0 \times 0 + 0 \times 0 + (-1000) \times (-1) - 1 \times 0 = 1000$$

beneficio indirecto beneficio directo evaluador neto

El evaluador neto correspondiente a la variable x_7 es $+1000$.

*Los tres evaluadores netos, correspondientes a las variables asociadas a columnas no unitarias (y por lo tanto de valores nulos), se muestran en el último renglón de la tabla simple (3.4.21).

*De la presentación anterior se concluye que los evaluadores netos se calculan de la siguiente manera:

*Solo hay evaluadores netos en las columnas no asociadas a vectores unitarios.

*Cálculo del renglón de evaluadores netos.

Los evaluadores netos representan el negativo del incremento que se produce en la función objetivo, al incrementar una variable nula en una unidad. Cuanto más negativo sea un evaluador neto, más se incrementa la función objetivo al aumentar de 0 a 1 la variable asociada. Por lo tanto:

Los evaluadores netos también nos indican si la solución que se tiene ya es la óptima.

*Para el problema que se está discutiendo, de la tabla (3.4.21) se observa que hay dos evaluadores netos negativos —1001 iguales, y correspondientes a las variables x_1 y x_2 . En este caso puede seleccionarse cualquiera de los dos. Emplearemos la primera, que se señala con una flecha en la tabla (3.4.21).

*Hemos decidido aumentar de cero a la variable no base x_1 , pero ¿hasta qué valor? Se verá que no se puede aumentar arbitrariamente, pues se podría violar alguna de las condiciones de no negatividad de las variables de la base, que disminuyen al aumentar x_1 .

Se multiplican las relaciones de sustitución de la columna en estudio por los beneficios unitarios correspondientes, se suman, y se le resta el valor del renglón de coeficientes de costos correspondiente a la columna analizada.

En problemas de maximización, para incrementar el valor de la función objetivo lo más rápidamente posible, debe aumentarse de cero el valor de aquella variable en cuya columna aparezca el evaluador neto más negativo.

En problemas de minimización, debe seleccionarse la variable con el valor más positivo en el renglón de evaluadores netos.

En problemas de maximización, si uno o más de los evaluadores netos es negativo, la solución *no* es óptima. Cuando todos los evaluadores netos son mayores o iguales a cero, la solución es óptima.

En problemas de minimización, si uno o más de los evaluadores netos es positivo, la solución *no* es óptima. Cuando todos los evaluadores netos son menores o iguales a cero, la solución es óptima.

°En problemas de maximización, marque con \uparrow la columna con el valor más negativo en el renglón de evaluadores netos.

En problemas de minimización, marque con \uparrow la columna con el valor más positivo en el renglón de evaluadores netos.

°¿Hasta cuánto puede aumentarse x_1 , sin violar la condición de no negatividad?

Las ecuaciones de restricción de la tabla (3.4.21) son

$$\begin{aligned} 0x_1 + x_2 + x_3 &= 90 \\ x_1 + 2x_2 + x_4 &= 210 \\ 2x_1 + x_2 + x_5 &= 240 \\ x_1 &+ x_6 = 110 \\ x_1 + x_2 - x_7 + a_8 &= 80 \end{aligned}$$

Como x_2 y x_7 seguirán siendo nulas, estas ecuaciones se transforman en

$$\begin{aligned} 0x_1 + x_3 &= 90 \\ x_1 + x_4 &= 210 \\ 2x_1 + x_5 &= 240 \\ x_1 + x_6 &= 110 \\ x_1 + a_8 &= 80 \end{aligned}$$

De estas ecuaciones se observa que, al aumentar x_1 , disminuyen las variables de la base x_3 , x_4 , x_5 , x_6 y a_8 .

El mínimo valor al que puede disminuir cada una de estas variables es cero, que nos da el máximo valor correspondiente al que puede aumentar x_1 :

$$\begin{aligned} 0x_1 &= 90, x_1 \rightarrow \infty \\ x_1 &= 210 \\ 2x_1 &= 240, x_1 = 120 \\ x_1 &= 110 \\ x_1 &= 80 \end{aligned} \quad (3.4.24)$$

*Como todas las variables deben permanecer no negativas simultáneamente, el mínimo de estos valores para x_1 nos da el máximo valor posible de x_1 :

*Valor máximo de x_1

$$x_{1\max} = 80$$

Si, por ejemplo, hubiéramos escogido

$$x_1 = 110$$

entonces la última ecuación

$$x_1 + a_8 = 80$$

nos hubiera dado

$$a_8 = 80 - 110 = -30$$

violándose la condición de no negatividad de una de las variables.

Obsérvese que cuando

$$x_1 = 80$$

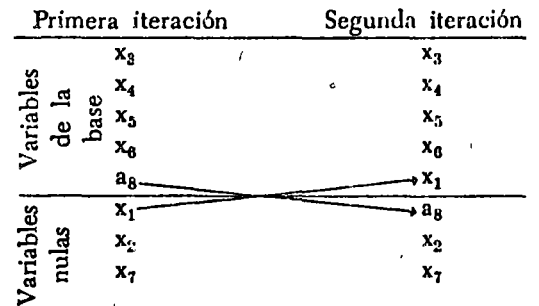
entonces

*Esto indica que la variable a_8 no nula en la primera iteración deja la base, al mismo tiempo que entra x_1 .

$$a_8 = 0$$

$$x_1 : 0 \rightarrow 80$$

$$a_8 : 80 \rightarrow 0$$



Las operaciones de las ecuaciones (3.4.24), se pueden realizar directamente en la tabla simplex (3.4.21). Para ello, basta colocar en una columna auxiliar los cocientes correspondientes de la columna de restricciones b_j , entre la columna correspondiente a la variable que entra a la base, en este caso x_1 .

	e_1	e_2	e_3	e_4	e_5	restricciones	Columna auxiliar
	x_2	x_3	x_4	x_5	x_6	b_j	
x_1	1	1	0	0	0	90	$90/0 \rightarrow \infty$
	2	0	1	0	0	210	$210/1 = 210$
	1	0	0	1	0	240	$240/2 = 120$
	1	0	0	0	1	110	$110/1 = 110$
	1	1	0	0	0	80	$80/1 = 80$

Observe que los valores de esta columna auxiliar coinciden con los de las ecuaciones (3.4.24).

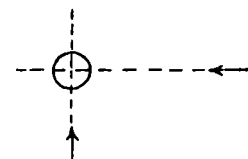
*El renglón correspondiente al menor valor positivo de la columna auxiliar se identifica y marca con una flecha:

*Identifique el renglón del menor valor positivo en la columna auxiliar con



*En un problema de maximización, la intersección de la columna con el valor más negativo (más positivo para minimización) en el renglón de evaluadores netos (marcado con \uparrow), con el renglón de menor valor positivo en la columna auxiliar (marcado con \leftarrow) define el llamado *pivote*.

*Pivote:



La variable x_1 sustituye a a_8 como variable base en la segunda

iteración, y en la primera iteración a_8 estaba asociada al vector unitario e_5 . Por facilidad computacional, debe ahora manipularse el sistema (3.4.21) para que la primera columna, asociada a x_1 , se convierta en e_5 . En resumen:

Primera iteración		Segunda iteración	
Variable base	a_8	x_1 (sustituye a a_8 en la base)	
Vector unitario asociado	e_5		e_5

En la tabla simplex debe cambiar

x_1		x_1
0...		0...
1...		0...
2...	a	0...
1...		0...
1...		1...

Desde luego que al hacer estos cambios, no debe cambiar la solución del sistema de ecuaciones que definen las restricciones del problema.

Recuérdese que si se suma a una ecuación de un sistema de ecuaciones lineales, otra ecuación del sistema multiplicada por una constante, no cambia la solución.

Por ejemplo, sea el sistema

$$\begin{aligned}x_1 + x_2 &= 10 \\ 2x_1 + x_2 &= 15\end{aligned}$$

cuya solución es:

$$\begin{aligned}x_1 &= 5 \\ x_2 &= 5\end{aligned}$$

Si ahora a la primera ecuación se le suma la segunda multiplicada por 3, se tiene el siguiente sistema:

$$\begin{aligned}x_1 + x_2 + 3(2x_1 + x_2) &= 10 + 3 \times 15 \\ 2x_1 + x_2 &= 15\end{aligned}$$

Este sistema

$$\begin{aligned}7x_1 + 4x_2 &= 55 \\ 2x_1 + x_2 &= 15\end{aligned}$$

es equivalente al anterior, ya que su solución también es:

También se obtienen sistemas de ecuaciones equivalentes si una o varias ecuaciones se multiplican o dividen por una constante.

En efecto, si se divide la primera ecuación entre 2 se tiene el sistema:

cuya solución también es

Las operaciones anteriores se conocen con el nombre de:

Realizando operaciones elementales de renglón, puede transformarse el sistema (3.4.21) en uno equivalente cuya primera columna sea precisamente e_5 .

*Debe empezarse por dividir el renglón del pivote entre el valor del pivote, para obtener un 1 en el pivote. En este ejemplo esta operación no tiene que realizarse, ya que el pivote vale 1.

*El renglón del pivote, con un coeficiente unitario en la posición del pivote, se designará con el nombre de *renglón de pivote normalizado* (r.p.n.) En nuestro ejemplo es

*La primera columna (correspondiente al pivote) se quiere cambiar a e_5 .

Para ello debe restarse el renglón del pivote normalizado multi-

$$\begin{aligned} x_1 &= 5 \\ x_2 &= 5 \end{aligned}$$

$$\begin{aligned} \frac{1}{2}x_1 + \frac{1}{2}x_2 &= 5 \\ 2x_1 + x_2 &= 15 \end{aligned}$$

$$\begin{aligned} x_1 &= 5 \\ x_2 &= 5 \end{aligned}$$

Operaciones elementales de renglón:

- 1) dividir o multiplicar una ecuación por una constante
- 2) sumar o restar a una ecuación otra ecuación multiplicada o dividida por una constante.

*Divida el renglón del pivote entre el pivote.

*Si el coeficiente del pivote es uno, se dice que el renglón está normalizado (r.p.n.)

$$\textcircled{1} \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad -1 \quad 1 \quad 80 \quad \text{r.p.n.}$$

x_1		e_5
$\begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \\ 1 \\ \textcircled{1} \end{bmatrix}$	→	$\begin{bmatrix} x_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$
Primera iteración		Segunda iteración

plicado por 0 del primer renglón, multiplicado por 1 del segundo renglón, por 2 del tercero y por 1 del cuarto:

Multiplique el	y reste del renglón
r.p.n. por	número
0	1
1	2
2	3
1	4

Realicemos estas operaciones.

El primer renglón no se modifica, pues el renglón de pivote normalizado multiplicado por 0 da cero, que restado al primer renglón no lo altera. Efectivamente, ya se tiene el 0 de e_5 en el primer elemento de la primera columna:

0	← no se necesita →	0
1	cambiar	0
2		0
1	→	0
1		1

De la tabla simplex (3.4.21) se obtienen los datos para modificar el segundo renglón:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	b_j
segundo renglón	1	2	0	1	0	0	0	0	210
(r.p.n.) $\times 1$	1	1	0	0	0	0	-1	1	80
Resta: nuevo segundo renglón:	0	1	0	1	0	0	1	-1	130

Para los dos renglones de restricciones restantes se tiene:

tercer renglón	2	1	0	0	1	0	0	0	240
(r.p.n.) $\times 2$	2	2	0	0	0	0	-2	2	160
Resta: nuevo tercer renglón	0	-1	0	0	1	0	2	-2	80
cuarto renglón	1	0	0	0	0	1	0	0	110
(r.p.n.) $\times 1$	1	1	0	0	0	0	-1	1	80
Resta: nuevo cuarto renglón:	0	-1	0	0	0	1	1	-1	30

*Con estos resultados se empieza a formar la tabla simplex (3.4.25) para la segunda iteración.

*Tabla simplex segunda iteración:

beneficios unitarios	base	e_5	e_1	e_2	e_3	e_4	restricciones	columna auxiliar
	x_1	x_2	x_3	x_4	x_5	x_6	b_j	
0	x_3	0	1	1	0	0	90	∞
0	x_4	0	1	0	1	0	130	130
0	x_5	0	-1	0	0	1	80	40
0	x_6	0	-1	0	0	1	30	30 ←
1	x_1	1	1	0	0	0	80	-80
coeficientes de costos c_j		1	1	0	0	0	0	-1 000
evaluadores netos			0				-1	1 001

↑

(3.4.25)

*En la segunda columna de la tabla anterior se identifican las variables de la nueva base

*Base segunda iteración:
 x_3, x_4, x_5, x_6, x_1

correspondientes a los vectores unitarios. La columna de restricciones b_j nos da los valores de estas variables:

$$\begin{aligned} x_3 &= 90 & x_6 &= 30 \\ x_4 &= 130 & x_1 &= 80 \\ x_5 &= 80 \end{aligned}$$

Las variables que no están en la base, correspondientes a las columnas no unitarias, valen cero:

$$\begin{aligned} x_2 &= 0 \\ x_7 &= 0 \\ a_8 &= 0 \end{aligned}$$

Los coeficientes de la función objetivo

$$x_1 + x_2 - 1\ 000 a_8 = z$$

en la nueva base son:

$$0, 0, 0, 0, 1$$

que corresponden a la nueva columna de beneficios unitarios.

*El valor de la función objetivo en esta iteración se obtiene de la suma de los productos correspondientes de la columna de beneficios unitarios con la columna de restricciones:

*Valor de la función objetivo en la segunda iteración:

beneficios unitarios	restricciones b_j	producto
0	90	0
0	130	0
0	80	0
0	30	0
1	80	80

suma: 80

*Es decir, en la segunda iteración la función objetivo, o sea el beneficio, vale

*Función objetivo segunda iteración:

$$z = 80$$

Este valor de la función objetivo es mucho mayor que el valor de

$$z = -80\,000$$

*en la primera iteración, y por lo tanto es mejor.

*Mejóro la función objetivo.

Los evaluadores netos correspondientes a las variables x_2 , x_7 y a_8 que no están en la base son:

Para x_2

beneficio unitario	x_2
0	1
0	1
0	-1
0	-1
1	1

coeficiente de costos
evaluador neto

$$\underbrace{0 \times 1 + 0 \times 1 + 0 \times (-1) + 0 \times (-1) + 1 \times 1}_{\text{beneficio indirecto}} - \underbrace{-1}_{\text{beneficio directo}} = \underbrace{0}_{\text{evaluador neto}}$$

y similarmente para x_7

$$\underbrace{0 \times 0 + 0 \times 1 + 0 \times 2 + 0 \times 1 + 1 \times (-1)}_{\text{beneficio indirecto}} - \underbrace{-0}_{\text{beneficio directo}} = \underbrace{-1}_{\text{evaluador neto}}$$

y a_8

$$\underbrace{0 \times 0 + 0 \times (-1) + 0 \times (-2) + 0 \times (-1) + 1 \times 1}_{\text{beneficio indirecto}} - \underbrace{(-1\,000)}_{\text{beneficio directo}} = \underbrace{1\,001}_{\text{evaluador neto}}$$

Estos evaluadores netos se muestran en el último renglón de la tabla (3.4.25).

El evaluador neto más negativo de -1 corresponde a x_7 , por lo que ésta será la siguiente variable que pasará a la base. *La co-

*Columna x_7 : ↑

columna correspondiente a x_7 se señala con \uparrow en la tabla (3.4.25).

La columna auxiliar se forma dividiendo los valores correspondientes de la columna de restricciones b_j entre la columna del pivote x_7 .

$$\begin{aligned} 90/0 &\rightarrow \infty \\ 130/1 &= 130 \\ 80/2 &= 40 \\ 30/1 &= 30 \\ 80/(-1) &= -80 \end{aligned}$$

*El menor de estos valores positivos, 30, corresponde al cuarto renglón, que se marca con una flecha \leftarrow . El nuevo pivote es el elemento.

*Cuarto renglón: \leftarrow

$$a_{47} = \textcircled{1} \leftarrow$$

\uparrow

que se muestra en la tabla (3.4.25).

En la tercera iteración, la variable x_7 entra a la base, y la variable de la base correspondiente al cuarto renglón x_6 , sale. *Como x_6 estaba asociada a e_4 , la columna x_7 se tendrá que cambiar a e_4 por operaciones elementales de renglón.

*Tercera iteración:

$$\begin{array}{c|c} x_7 & e_4 \\ \hline 0 & 0 \\ 1 & 0 \\ 2 & 0 \\ \textcircled{1} & 1 \\ -1 & 0 \end{array} \rightarrow \begin{array}{c|c} e_4 & x_7 \\ \hline 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 0 \end{array}$$

Segunda iteración Tercera iteración

El renglón del pivote normalizado es:

cuarto renglón: 0 -1 0 0 0 1 $\textcircled{1}$ -1 30 r.p.n.

Para mantener la columna de restricciones b_j positiva, las operaciones elementales de renglón que se deben realizar son:

Al renglón	número	reste el r.p.n.	multiplicado por
	2		1
	3		2
	5		-1

*La tabla simplex correspondiente a la tercera iteración es la siguiente:

*Tabla simplex tercera iteración:

beneficios unitarios	base	$e_5 \quad e_1 \quad e_2 \quad e_3 \quad e_4$								b_j	Columna auxiliar	
		x_1	x_2	x_3	x_4	x_5	x_6	x_7	a_8			
0	x_3	0	1	1	0	0	0	0	0	90	90	
0	x_4	0	2	0	1	0	-1	0	0	100	50	
0	x_5	0	①	0	0	1	-2	0	0	20	20	← (3.4.26)
0	x_7	0	-1	0	0	0	1	1	-1	30	-30	
1	x_1	1	0	0	0	0	1	0	0	110	∞	
coeficientes de costos c_j		1	1	0	0	0	0	0	-1	000		
Evaluadores netos			-1				1		1	000		

La función objetivo para la tercera iteración vale:

Función objetivo Tercera iteración:

$$z = x_1 + x_2 - 1\,000 a_8$$

$$= 110 + 0 - 0$$

$$z = 110$$

El pivote es

$$a_{32} = \textcircled{1} \leftarrow$$

$$\uparrow$$

por lo que x_2 entra a la base y x_5 sale. La columna x_2 para la cuarta iteración debe cambiar a e_3 :

x_2	e_3
1	0
2	0
①	1
-1	0
0	0

Tercera iteración

Cuarta iteración

Realizando las operaciones elementales de renglón necesarias se llega a la tabla simplex para la cuarta iteración:

Tabla simplex cuarta iteración:

beneficio unitario	base	$e_5 \quad e_3 \quad e_1 \quad e_2 \quad e_4$								b_j	columna auxiliar	
		x_1	x_2	x_3	x_4	x_5	x_6	x_7	a_8			
0	x_3	0	0	1	0	-1	2	0	0	70	35	
0	x_4	0	0	0	1	-2	③	0	0	60	20	←
1	x_2	0	1	0	0	1	-2	0	0	20	-10	
0	x_7	0	0	0	0	1	-1	1	-1	50	-50	
1	x_1	1	0	0	0	0	1	0	0	110	110	
Coeficientes de costos c_j		1	1	0	0	0	0	0	-1	000		
Evaluadores netos						1	-1		1	000		

(3.4.27)

*Analizando la tabla (3.4.27) se observa que todavía hay valores negativos en el renglón de evaluadores netos.

*La función objetivo, que en esta cuarta iteración vale

todavía puede mejorar.

El nuevo pivote es

por lo que en la quinta iteración x_6 debe estar a la base y x_4 ya no debe estar.

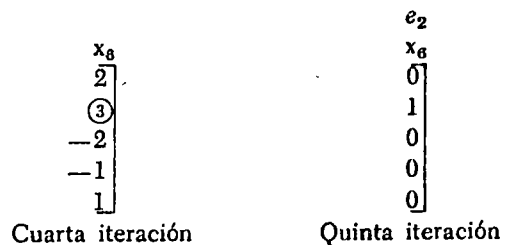
La columna x_6 debe cambiar a e_2 :

*Todavía hay elementos negativos en el renglón de evaluadores netos. Todavía puede mejorar la función objetivo.

*Función objetivo cuarta iteración:

$$\begin{aligned} z &= x_1 + x_2 - 1\ 000 a_8 \\ &= 110 + 20 - 0 \\ z &= 130 \end{aligned}$$

$$a_{20} = \begin{matrix} \textcircled{3} \\ \uparrow \end{matrix} \leftarrow$$



En esta ocasión el renglón del pivote debe dividirse entre el valor del pivote para normalizarlo:

segundo renglón: $0 \ 0 \ 0 \ \frac{1}{3} \ -\frac{2}{3} \ 1 \ 0 \ 0 \ 20 \ \text{r.p.n.}$

*Realizando las operaciones elementales de renglón se obtiene la tabla simplex para la quinta iteración:

*Tabla simplex quinta iteración:

beneficio unitario	base	$e_3 \ e_3 \ e_1$			$e_2 \ e_4$				b_j	
		x_1	x_2	x_3	x_4	x_5	x_6	x_7		a_8
0	x_3	0	0	1	$-2/3$	$1/3$	0	0	0	30
0	x_6	0	0	0	$1/3$	$-2/3$	1	0	0	20
1	x_2	0	1	0	$2/3$	$-1/3$	0	0	0	60
0	x_7	0	0	0	$1/3$	$1/3$	0	1	-1	70
1	x_1	1	0	0	$-1/3$	$2/3$	0	0	0	90
Coeficientes de costos c_j		1	1	0	0	0	0	0	-1 000	
Evaluadores netos					$1/3$	$1/3$			1 000	

*Puesto que todos los evaluadores netos de la tabla simplex

*Función objetivo máxima

(3.4.28) para la quinta iteración son positivos, la función objetivo ya es máxima. *Su valor es de

*Función objetivo:

$$\begin{aligned} z &= x_1 + x_2 - 1\,000 a_8 \\ &= 90 + 60 - 0 \\ z &= 150 \end{aligned}$$

*Las variables de la base valen:

*Variables de la base:

$$\begin{aligned} x_3 &= 30 \\ x_6 &= 20 \\ x_2 &= 60 \\ x_7 &= 70 \\ x_1 &= 90 \end{aligned}$$

*y las variables que no están en la base:

*Variables no base:

$$\begin{aligned} x_4 &= 0 \\ x_5 &= 0 \\ a_8 &= 0 \end{aligned}$$

Interpretemos estos resultados.

La función objetivo

que hemos maximizado al valor

representa el número total de piezas producidas de los artículos 1 y 2 (ver el enunciado del ejemplo 3.4.1).

El número de artículos 1 que se producen en este momento es

$$z = x_1 + x_2 - 1\,000 a_8$$

$$z = 150$$

$$x_1 = 90$$

y el número de artículos 2 es

$$x_2 = 60$$

La máquina A, que se emplea 2 horas para producir una pieza del artículo 2, está disponible únicamente 180 horas. La ecuación de restricción correspondiente

$$2x_2 \leq 180$$

se modificó a

$$x_2 + x_3 = 90$$

mediante la introducción de la variable de holgura x_3 . Como $x_2 =$

60 y $x_3 = 30$, la ecuación anterior multiplicada por dos

$$2x_2 + 2x_3 = 180$$

nos indica que la máquina no se emplea durante

$$2x_3 = 60$$

horas.

Para la máquina B la ecuación de restricción

$$x_1 + 2x_2 \leq 210$$

se modificó a

$$x_1 + 2x_2 + x_4 = 210$$

Por lo tanto, la máquina B no se emplea durante

$$x_4 = 0$$

horas. Es decir, la máquina B está plenamente aprovechada.

La ecuación de restricción

$$2x_3 + x_2 \leq 240$$

para la máquina C se modificó a

$$2x_3 + x_2 + x_5 = 240$$

por lo que el tiempo que no se emplea dicha máquina es de

$$x_5 = 0$$

horas. La máquina C también está plenamente aprovechada.

La ecuación de restricción para la máquina D

$$3x_1 \leq 330$$

$$x_1 \leq 110$$

se alteró a

$$x_1 + x_6 = 110$$

por lo que la máquina D no se emplea durante

$$3x_6 = 3 \times 20 = 60$$

horas.

El nivel mínimo de producción dado por

$$x_1 + x_2 \geq 80$$

efectivamente se cumple, pues

$$x_1 + x_2 = 90 + 60 = 150$$

y

$$150 > 80$$

Los resultados anteriores, obtenidos matemáticamente empleando el método simplex, coinciden con los resultados gráficos del comienzo de la subsección 3.4.4, resumidos en la tabla 3.4.3.

3.4.7 El problema dual

*En la sección 3.4.4 se observó que el ejemplo 3.4.4 de programación lineal, podía plantearse y resolverse empleando como variables los niveles de actividad, o bien los niveles de recursos. *En esta subsección se mostrará que a cada problema de programación lineal de maximización, está asociado un problema de programación lineal de minimización, y viceversa. *El problema original recibe el nombre de *primo*, y el asociado a él es su *dual*.

*Un problema de programación lineal se puede resolver en el espacio de actividades o en el espacio de recursos.

*En programación lineal, a todo problema de maximización está asociado uno de minimización, y viceversa.

*El primer problema se llama primo, y el segundo es su dual.

*El problema dual es muy importante en programación lineal, principalmente por las siguientes razones:

*Importancia del problema dual.

- 1) El problema dual tiene importantes interpretaciones económicas.
- 2) Ayuda a estudiar la teoría de programación lineal y a establecer métodos eficientes para resolver el problema.
- 3) En algunos casos, el uso del problema dual emplea menos capacidad de memoria en la computadora que el problema primo.
 - a) Si el problema primo tiene más restricciones que variables, el problema dual tendrá más variables que restricciones. En este caso es más eficiente resolver el problema dual, pues requerirá menos iteraciones.

Antes de enunciar los principales teoremas de dualidad, se definirá matemáticamente el dual de un problema de programación lineal. *Empleando notación matricial, la relación entre el problema primo y el problema dual es la siguiente:

*Relación matemática entre un problema primo y su dual:

Problema Primo	Problema Dual
Función objetivo:	
max: $z = c^T x$	min: $v = b^T u$ (3.4.29)

$$\begin{aligned} Ax &\leq b \\ x &\geq 0 \end{aligned}$$

*Debe observarse que si el problema primo es de maximización de la función objetivo, su dual es de minimización de otra función objetivo. La matriz A del problema primo corresponde a la matriz transpuesta A^T del problema dual. El vector b de restricciones del problema primo aparece transpuesto en la función objetivo del dual, y viceversa. En ambos casos los vectores de niveles x y u que constituyen las incógnitas son no negativos.

Como muestran las relaciones (3.4.29) y (3.4.30), para establecer el problema dual es necesario introducir un nuevo vector de incógnitas u . Su dimensión puede deducirse fácilmente de la relación (3.4.30). Si la dimensión de la matriz A es

su transpuesto es de dimensión

Para poder realizar el producto

de la ecuación (3.4.30), es necesario que el vector u sea de dimensión

Es decir, el número de componentes de u es igual al número de restricciones.

A continuación se establecerá el problema dual del ejemplo de programación lineal 3.4.4. Se dará una interpretación económica a las componentes del vector u . Además se ilustrará la solución gráfica y analítica de un problema de minimización.

Establezca el problema de programación lineal dual al del ejemplo 3.4.4, y resuélvalo gráfica y analíticamente.

De acuerdo con las relaciones (3.4.29) y (3.4.30), la formulación matricial del problema primo y de su dual es:

Restricciones:

$$\begin{aligned} A^T u &\geq c \\ u &\geq 0 \end{aligned} \quad (3.4.30)$$

*Correspondencias:

PRIMO	DUAL
maximización	minimización
A	A^T
restricciones b	objetivo b^T
objetivo c^T	restricciones c
$x \geq 0$	$u \geq 0$

$A_{m \times n}$
 m = número de restricciones
 n = número de actividades.

$$A^T_{n \times m}$$

$$A^T u$$

$$u_{m \times 1}$$

Ejemplo 3.4.6

Solución:

PROBLEMA PRIMO

PROBLEMA DUAL

Función objetivo:

$$\max: \begin{bmatrix} 5 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = z \quad \min: \begin{bmatrix} 108\,750 & 112\,500 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = v \quad (3.4.31)$$

Restricciones:

$$\begin{bmatrix} 40 & 55 \\ 75 & 30 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 108\,750 \\ 112\,500 \end{bmatrix} \quad \begin{bmatrix} 40 & 75 \\ 55 & 30 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \geq \begin{bmatrix} 5 \\ 4 \end{bmatrix} \quad (3.4.32)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Para resolver gráficamente el problema dual, es necesario determinar, en el plano (u_1, u_2) , la zona de soluciones factibles definida por las restricciones (3.4.32) del problema dual. Dicha región aparece hachurada en la figura 3.4.15.

Para resolver el problema gráficamente, debe trasladarse una recta de la función objetivo

$$v = 108\,750u_1 + 112\,500u_2$$

paralelamente a sí misma. Como el problema es de minimización, la traslación debe hacerse hasta que la recta quede lo más cerca posible del origen, pero todavía con por lo menos un punto dentro de la zona de soluciones factibles.

Tracemos la recta de la función objetivo correspondiente, por ejemplo, a $v = 20\,000$:

$$20\,000 = 108\,750u_1 + 112\,500u_2$$

Cuando u_2 vale cero, esta recta intersecta al eje de las abscisas u_1 en el punto

$$u_1 = \frac{20\,000}{108\,750} = 0.184$$

Cuando u_1 vale cero, la recta intersecta al eje de las ordenadas u_2 en

$$u_2 = \frac{20\,000}{112\,500} = 0.178$$

Esta recta se muestra en la figura 3.4.15. Traslado la recta hacia el origen, se obtiene gráficamente el punto en el cual la función objetivo es mínima.

De la gráfica, las coordenadas del punto mínimo son, aproximadamente,

$$\begin{aligned} u_1 &= .05 \\ u_2 &= .04 \end{aligned}$$

La función objetivo en este punto tiene un valor mínimo aproximado de

$$\begin{aligned} v &= 108\,750u_1 + 112\,500u_2 \\ &= 108\,750 \times .05 + 112\,500 \times .04 \\ v &\simeq 9937.5 \end{aligned}$$

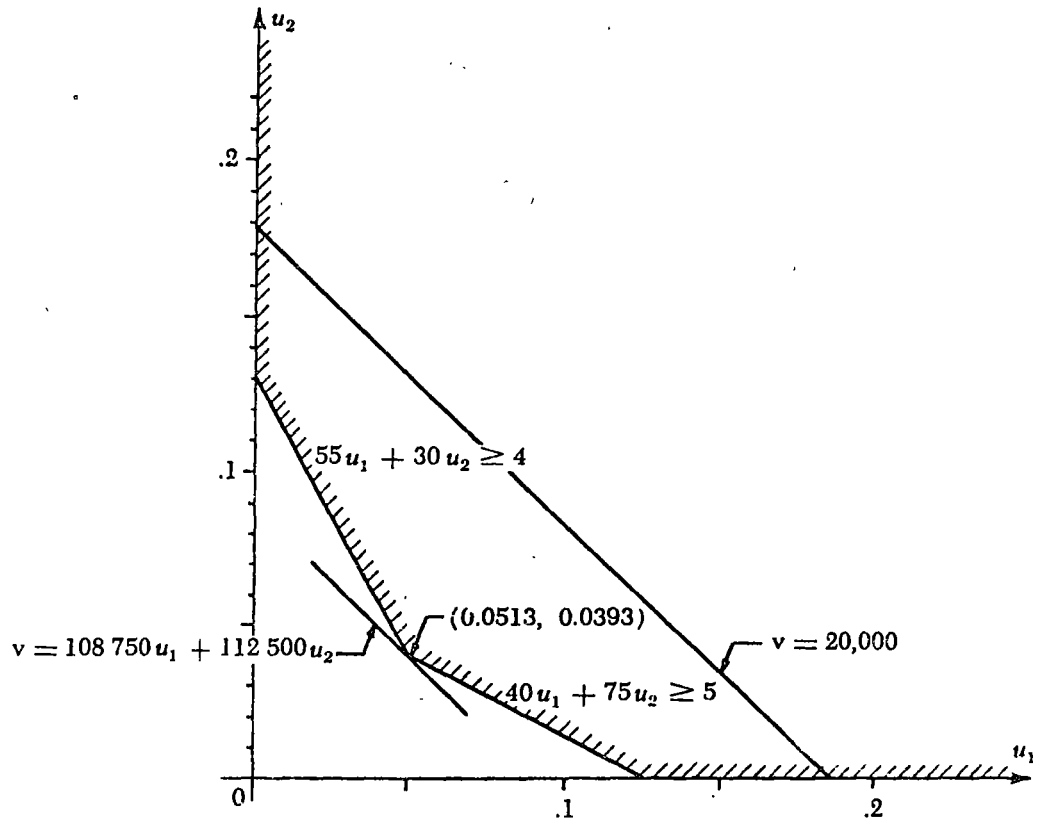


Fig. 3.4.15 Solución gráfica del ejemplo 3.4.6.

*Para obtener la solución analítica del problema dual, que es de minimización, debe empezarse por establecer la llamada *forma normal* del problema de programación lineal. Para pasar a la forma normal, las desigualdades de las restricciones deben convertirse en igualdades.

*Forma normal del problema de programación lineal: las restricciones deben expresarse como igualdades.

*En nuestro ejemplo dual, por tener desigualdades del tipo "mayor o igual que", debe restarse una variable de holgura y sumarse una variable artificial a cada ecuación.

*En desigualdades del tipo \geq , introduzca una variable de holgura con signo $-$, y una artificial con signo $+$.

Introduciendo las variables mencionadas en las restricciones (3.4.32), se obtiene:

$$\begin{aligned} 40u_1 + 75u_2 - s_1 + a_1 &= 5 \\ 55u_1 + 30u_2 - s_2 + a_2 &= 4 \end{aligned}$$

Estas igualdades y la función objetivo permiten establecer el primer arreglo simplex que aparece en la tabla 3.4.5. *Obsérvese que, por tratarse de un problema de minimización, las variables artificiales se suman a la función objetivo con un gran coeficiente positivo M.

*Las variables artificiales se suman a la función objetivo con un gran coeficiente positivo M.

*Por tratarse de un problema de minimización, la columna del pivote será aquella con el evaluador neto más positivo. La búsqueda termina cuando ya no hay evaluadores netos positivos.

*En problemas de minimización, la columna del pivote está definida por el mayor evaluador neto positivo. La búsqueda termina cuando ya no hay evaluadores positivos.

La tabla 3.4.5 indica los pasos sucesivos en la solución del problema. El último arreglo indica que la solución es

$$\begin{aligned} u_1 &= 0.0513 \\ u_2 &= 0.0393 \end{aligned}$$

con un mínimo de la función objetivo

$$\begin{aligned} v &= 108\,750u_1 + 112\,500u_2 \\ &= 108\,750 \times 0.0513 + 112\,500 \times 0.0393 \\ &= 1\,000. \end{aligned}$$

beneficio unitario	base	u_1	u_2	s_1	s_2	e_1 a_1	e_2 a_2	restricciones b_j	columna auxiliar
M	a_1	40	75	-1	0	1	0	5	0.067 ←
M	a_2	55	30	0	-1	0	1	4	0.133
Coeficientes de costos c_j		108 750	112 500	0	0	M	M		
Evaluadores netos		-108 750	-112 500	-M	-M				
		+ 95M	+ 105M						
			↑						
112 500	u_2	0.53	1	-0.013	0	0.013	0	0.067	0.125
M	a_2	39	0	0.4	-1	-0.4	1	2.000	0.0513 ←
Coeficientes de costos c_j		108 750	112 500	0	0	M	M		
Evaluadores netos		-48 750		-1 500		+1 500			
		+39M		+0.4M	-M	-0.4M			
			↑						
112 500	u_2	0	1	-0.0192	+0.0137	+0.0192	-0.0137		0.0393
108 750	u_1	1	0	0.0103	-0.0256	-0.0103	0.0256		0.0513
Coeficientes de costos c_j		108 750	112 500	0	0	M	M		
Evaluadores netos				-2 100	-1242.75	-1039.875	1242.75		
						-M	-M		

Tabla 3.4.5 Solución analítica del problema dual del ejemplo 3.4.4.

*Obsérvese que la solución del problema dual coincide con el valor de los recursos del ejemplo 3.4.4 (ver ecuaciones (3.4.12) y (3.4.13)). *Además, el valor máximo de 10 000 (= 10 millones) de la función objetivo del problema primo, coincide con el mínimo del problema dual. Estos resultados son generales.

*Solución del problema dual = valor de los recursos del problema primo
 *Máximo del problema primo = mínimo del problema dual

A continuación propondremos una interpretación económica para las variables u_1 y u_2 , que después veremos es correcta.

Para producir un cambio Δz en la ganancia, el capital asignado a la compra de materia prima cambia en cierta cantidad Δk . Desig-nemos con u_1 al cociente de estos cambios

$$u_1 = \frac{\Delta z}{\Delta K}$$

Análogamente, para producir un cambio Δz en la ganancia o be-neficio, el capital asignado al pago de mano de obra cambia en ΔL ; el cociente se define como u_2

$$u_2 = \frac{\Delta z}{\Delta L}$$

En otras palabras, u_1 es el valor del capital que se asigna a una unidad de materia prima, y u_2 es el valor del capital que se asig-na a una unidad de mano de obra.

De la tabla 3.4.4

	K Materia Prima	L Mano de Obra	Beneficio
Torbellino	40	75	5
Ciclón	55	30	4
Disponibilidad	108 750	112 500	

Tabla 3.4.4 Datos del ejemplo 3.4.4 en miles de pesos.

recordamos que, para producir un torbellino, se requiere de $K = \$40$ miles en materia prima y $L = \$75$ miles en mano de obra. El producto

$$K \frac{\Delta z}{\Delta K} + L \frac{\Delta z}{\Delta L} = 40u_1 + 75u_2$$

representa la ganancia, el beneficio que le produce al fabricante hacer un Torbellino, en términos de las variables u_1 y u_2 . De la tabla 3.4.4 vemos que el beneficio de producir un Torbellino es por lo menos de \$5 mil, es decir

$$40u_1 + 75u_2 \geq 5$$

Esta ecuación no es más que la primera ecuación de restricción (3.4.32) del problema dual.

En forma análoga se establece la ecuación

$$55u_1 + 30u_2 \geq 4$$

para la producción de Ciclones, que es la segunda ecuación de restricción (3.4.32).

Como se dispone de \$108 750 miles para la compra de materia prima y de \$112 500 miles para pagar mano de obra,

$$108\,750u_1 + 112\,500u_2$$

representa el gasto total para producir Torbellinos y Ciclones.

Para interpretar económicamente el problema dual, consideremos el punto de vista de la agencia de coches que compra los Torbellinos y Ciclones producidos por la fábrica.

*En la agencia de coches se quiere determinar el valor del capital que se asigna a materia prima (u_1) y el valor del capital que se asigna a mano de obra (u_2), para minimizar el costo total

*Minimizar

$$108\,750u_1 + 112\,500u_2$$

en la compra de los Torbellinos y de los Ciclones que produce la fábrica. Este costo está sujeto a dos restricciones:

$$\begin{aligned} 40u_1 + 75u_2 &\geq 5 \\ 55u_1 + 30u_2 &\geq 4 \end{aligned}$$

puesto que el fabricante gana por lo menos \$5 mil en la producción de cada Torbellino y \$4 mil en la fabricación de un Ciclón.

De este planteamiento, se observa que la interpretación dada a las variables u_1 y u_2 fue correcta.

Además, como ya se observó en el ejemplo 3.4.4,

$$\begin{aligned} u_1 &= 0.0513 = 5.13\% \\ u_2 &= 0.0393 = 3.93\% \end{aligned}$$

representa el máximo porcentaje sobre el capital que puede pagar el fabricante para financiar la producción de Torbellinos y Ciclones.

3.4.8 Planeación de la producción

En los ejemplos anteriores el número de variables ha sido peque-

ño, lo cual permitió resolver los problemas sin ayuda de una computadora. En la mayoría de los problemas reales, el número de variables es mucho mayor. A continuación se introduce un ejemplo representativo de una situación real, con un número mayor de variables.

Ejemplo 3.4.7

Un aserradero se establece provisionalmente en un local, del cual se tiene que cambiar al cabo de tres meses. Por ello, al iniciar sus operaciones y a los tres meses no debe tener madera en bodega. La tabla 3.4.6 muestra la demanda mensual de productos aserrados durante los tres meses en cuestión. Se produce 1 m³ de productos aserrados en 2 horas de trabajo. Durante los dos últimos meses, la demanda excede a la cantidad de madera que puede producirse sin trabajar horas extras. Para surtir la demanda, puede recurrirse a trabajar horas extras, o a producir un exceso de madera durante los meses de baja demanda y almacenarla para venderla posteriormente. El costo de almacenamiento de un mes al siguiente es de \$15 por 1 m³ de productos aserrados.

La tabla 3.4.6 muestra la demanda mensual, los costos de producción en tiempo normal y en tiempo extra, y las horas de trabajo disponibles en tiempo normal y en tiempo extra.

	M E S		
	1	2	3
Demanda mensual (100 m ³)	15	25	20
Costo de producción en tiempo normal (\$/100 m ³)	\$ 20 000	\$ 22 000	\$ 22 000
Costo de producción en tiempo extra (\$/100 m ³)	\$ 30 000	\$ 33 000	\$ 33 000
Horas de trabajo disponibles en tiempo normal (horas/mes)	3 200	3 200	3 200
Horas de trabajo disponibles en tiempo extra (horas/mes)	1 600	1 600	1 600

Tabla 3.4.6 Datos del ejemplo 3.4.7.

Obtenga un modelo matemático que permita encontrar el programa mensual de producción que minimice los costos.

Solución:

Debe empezarse definiendo las variables del problema. Analizando la información, se observa que se necesitarán las siguientes variables:

x_1 = producción el primer mes en horas normales (100 m³)

x_2 = producción el segundo mes en horas normales (100 m³)

x_3 = producción el tercer mes en horas normales (100 m³)

x_4 = producción el primer mes en horas extras (100 m³)

x_5 = producción el segundo mes en horas extras (100 m³)

x_6 = producción el tercer mes en horas extras (100 m³)

x_7 = producción almacenada al terminar el primer mes (100 m³)

x_8 = producción almacenada al terminar el segundo mes (100 m³)

A continuación es necesario definir las ecuaciones de restricción.

*Durante el primer mes se tiene una producción en horas normales y horas extras de:

°Primer mes:

Disponibilidad:

$$x_1 + x_4$$

*Parte de esta producción se destina a surtir la demanda, y parte se almacena:

°Destino:

$$15 + x_7$$

Ambas cantidades deben ser iguales:

$$x_1 + x_4 = 15 + x_7$$

*En el segundo mes, los productos aserrados disponibles son los que se producen ese mes en horas normales y en horas extras, y los almacenados durante el primer mes:

°Segundo mes:

Disponibilidad:

$$x_2 + x_5 + x_7$$

*Parte de esta madera surte la demanda, y parte se almacena:

°Destino:

$$25 + x_8$$

Ambas cantidades deben ser iguales:

$$x_2 + x_5 + x_7 = 25 + x_8$$

*Finalmente, para el último mes se dispone de la madera producida en el tercer mes en horas normales y en horas extras, y la

°Tercer mes:

almacenada durante el mes anterior:

Disponibilidad:

$$x_3 + x_6 + x_8$$

*Esta madera se destina exclusivamente a surtir la demanda:

*Destino:

$$20$$

puesto que toda la madera de la que se dispone al final del tercer mes debe ser vendida para dejar vacía la bodega. Igualando ambas cantidades se tiene:

$$x_3 + x_6 + x_8 = 20$$

Las horas de trabajo disponibles también imponen restricciones.

Como durante el primer mes hay una producción de $x_1 \times 100 \text{ m}^3$, y para producir 1 m^3 de madera asenada se requieren 2 horas de trabajo, durante el primer mes se trabajan

$$x_1 \times 100 \times 2 = 200 x_1$$

horas en tiempo normal. Durante este mes se dispone de

$$3\ 200$$

horas de trabajo en tiempo normal, por lo que

$$200 x_1 \leq 3\ 200$$

Un razonamiento similar explica las siguientes cinco desigualdades:

$$200 x_2 \leq 3\ 200$$

$$200 x_3 \leq 3\ 200$$

$$200 x_4 \leq 1\ 600$$

$$200 x_5 \leq 1\ 600$$

$$200 x_6 \leq 1\ 600$$

Desde luego no tiene significado una producción negativa, por lo que

$$x_i \geq 0 \text{ para toda } i$$

*Finalmente es necesario establecer la función objetivo, que es el costo total de producción que debe minimizarse.

*Función objetivo:
minimización de costos.

*Los costos de producción en horas normales para los tres meses es de:

*Costos de producción en horas normales:

Entre las referencias bibliográficas el lector encontrará varias obras que incluyen programas para computadora, para resolver modelos de programación lineal. Muchas computadoras incluyen paquetes de programación lineal en sus programotecas.



centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



METODOS NUMERICOS Y APLICACIONES
CON LA COMPUTADORA DIGITAL

COMPLEMENTO AL TEMA
PROGRAMACION FORTRAN

ING. CARLOS RAMOS LARIOS

ABRIL, 1978.

Improved Programming Technologies — An Overview

This document is intended to briefly describe to the reader six recently formalized techniques designed to improve the program development process: structured programming, top-down program development, chief programmer teams, development support libraries, HIPO (Hierarchy plus Input-Process-Output), and structured walk-throughs. These techniques are still evolving; initial use in a data processing activity should be subject to management review, to determine the form in which the techniques may best fit into each environment.

Introduction

The last decade has been characterized by significant improvements in hardware speed and capacity, configuration flexibility, and programming system capability. There have also been many improvements in the capabilities of programming languages, but, in general, improvements in the techniques used in the program development process have lagged behind those in other areas. This period has also been characterized by the increasing complexity of application systems and by their importance to the organization. And, in the same period, application development, maintenance, and modification activities have comprised an increasing portion of the data processing budget. Data processing management, therefore, has been searching for ways to improve the program development process, with the objective of producing application systems that meet the needs of their users, are more error-free, require less maintenance, are easier to modify, and are developed on schedule with improved productivity.

This document describes six evolving techniques which have been implemented in various ways in

some program development efforts within IBM and which may be of assistance in achieving management objectives. These techniques, some of which have elements that have been advocated or used in the past, are structured programming, top-down program development, chief programmer teams, development support libraries, HIPO (Hierarchy plus Input-Process-Output), and structured walk-throughs. The first four have frequently been used together in IBM's Federal Systems Division. HIPO and structured walk-throughs were developed separately and seem to logically complement structured programming, top-down programming, chief programmer teams, and development support libraries.

These techniques can be used individually or together. Since they are still evolving, their initial use in a data processing activity should be subject to management review, to determine the form in which they may best fit into its environment.

Chapter 1: Structured Programming

Traditionally, each programmer has applied his own set of rules to the construction of the logic of his program. He starts with this logic structure and, as he encounters additional combinations of conditions to be met, he adds them as afterthoughts rather than revising the logic of the program. The resultant control code might look like that shown in the left (Unstructured) column of Figure 1. This code contains a large number of GO TO statements and labels and its logic is not easy to follow. During subsequent unit and integration testing, disintegration of

the programmer's original structure occurs as new constraints and conditions are imposed upon it—leading to more GO TO statements, more labels, and a final program whose original logic may be completely obscured. Reading, understanding, and testing such programs is difficult. The degree of confidence in their quality or correctness tends to be low. In addition, such programs tend to be difficult to maintain and modify.

UNSTRUCTURED	STRUCTURED
<pre> IF p GOTO label q IF w GOTO label m L function GOTO label k label m M function GOTO label k label q IF q GOTO label t A function B function C function label r IF NOT r GOTO label s D function GOTO label r label s IF s GOTO label f E function label v IF NOT v GOTO label k J function label k K function END function label f F function GOTO label v label t IF t GOTO label a A function B function GOTO label w label a A function B function G function label u IF NOT u GOTO label w H function GOTO label u label w IF NOT t GOTO label y I function label y IF NOT v GOTO label k J function GOTO label k </pre>	<pre> ① IF p THEN A function B function ② IF q THEN ③ IF t THEN G function ④ DOWHILE u H function ④ ENDDO I function ③ (ELSE) ③ ENDIF ② ELSE C function ③ DOWHILE r D function ③ ENDDO ③ IF s THEN F function ③ ELSE E function ③ ENDIF ② ENDIF ② IF v THEN J function ② (ELSE) ② ENDIF ① ELSE ② IF w THEN M function ② ELSE L function ② ENDIF ① ENDIF K function END function </pre>

Figure 1. A comparison of structured and unstructured code

Research by computer scientists and mathematicians indicates that an alternative method of programming known as structured programming can help solve these problems. This technique involves coding programs using a limited number of control logic structures to form highly structured units of code that are more readable, and therefore more easily tested, maintained and modified.

Structured Programming Theory

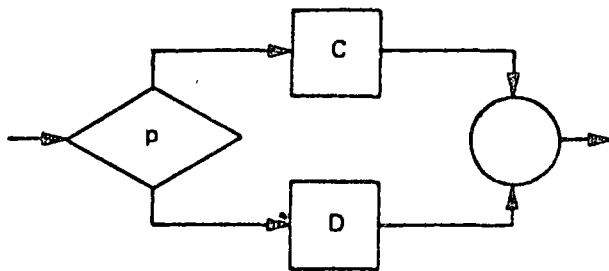
Structured programming is based on a mathematical-ly proven structure theorem¹ which states that any program can be written using only the three control logic structures illustrated in Figure 2:

- Sequence of two or more operations (MOVE,ADD,...)
- Conditional branch to one of two operations and return (the IF p THEN C ELSE D of Figure 2)
- Repetition of an operation while a condition is true (the DO E WHILE q of Figure 2)

Sequence of two operations



IF THEN ELSE: Conditional branch to one of two operations and return



DO WHILE: Operation repeated while a condition is true

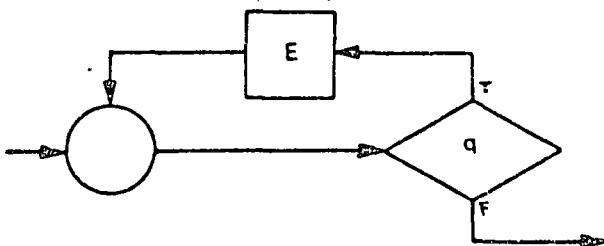


Figure 2. The three elemental logic structures of structured programming

Any program may be developed by the appropriate iteration and nesting of these three basic structures. Each of the three structures has only one entry and one exit. A program consisting solely of these structures is a proper program, a program with one entry and one exit. As illustrated in the structured code (right column) of Figure 1, it always proceeds from the beginning to the end without arbitrary branching. In PL/I, for instance, no GO TO statements are necessary. Proving the logical correctness of structured code is more feasible. The logic is easier to follow, permitting functions to be isolated, understood, and tested.

The use of the three control logic structures in structured programming is analogous to the hardware design practice of forming complex logic circuits from AND, OR, and NOT gates. This practice is based on a theorem in Boolean algebra which states that arbitrarily complex logic functions can be expressed in terms of basic AND, OR, and NOT operations. The use of three control logic structures in structured programming is similarly based on a solid theoretical foundation.

Extensions to the three basic logic structures are permitted as long as they retain the one-entry, one-exit property. An example of such an extension is the DOUNTIL structure (Figure 3), which provides for the execution of the function F until a condition is true.

DOUNTIL: Operation repeated until a condition is true

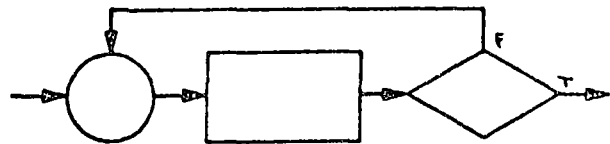


Figure 3. The DOUNTIL structure

Structured Programming Practices

Certain practices are followed to support the objective of producing readable, understandable structured programs—programs in which the writers can have a high degree of confidence.

Indenting within control structure blocks to reflect the logic of the program unit is one of these practices, as shown in the example of structured code in Figure 1. As illustrated by the number to the left of the statements, each logic structure nested within another is indented within it. All parts of a logic structure carry the same indentation level, and functions performed within a logic structure are indented within that structure. This practice highlights the

logic of the unit for the writer and the reader and thus contributes to the goal of more readable programs.

Limiting a unit of source code to a specified size—often one listed page, or fifty lines, permits the programmer to read and understand an entire logical expression or function without referring to multiple

pages or relying on his memory. Should the complete logical expression require more than 50 lines of source code, the programmer can segment the code through the use of such statements as %INCLUDE (in PL/I) or COPY (in COBOL) to specify the inclusion of another unit of code (see Figure 4).

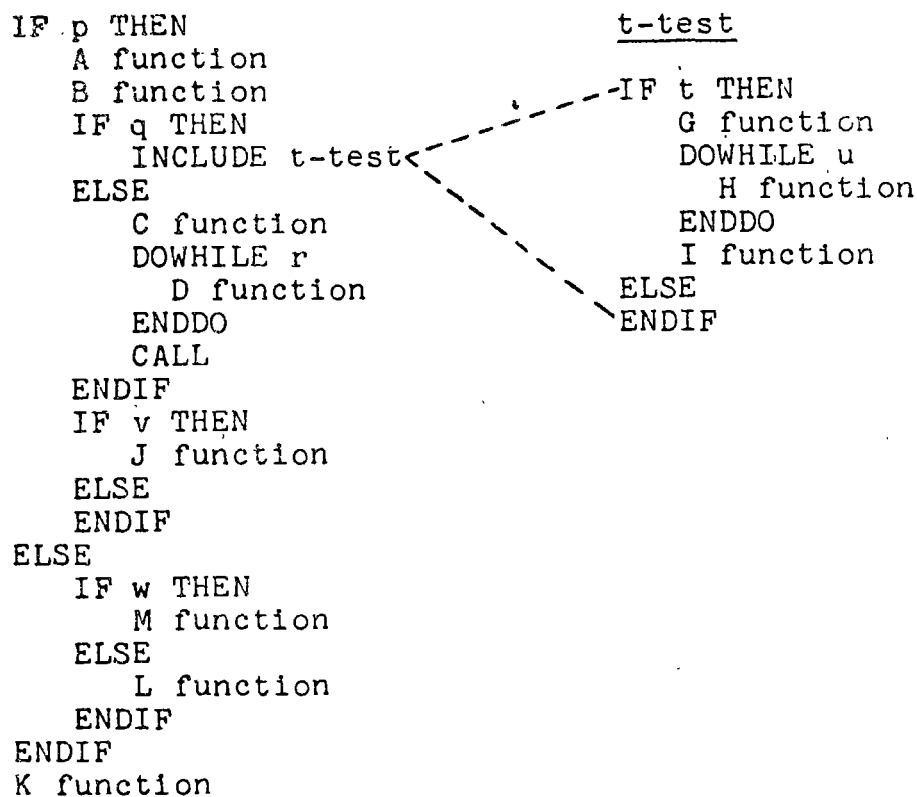
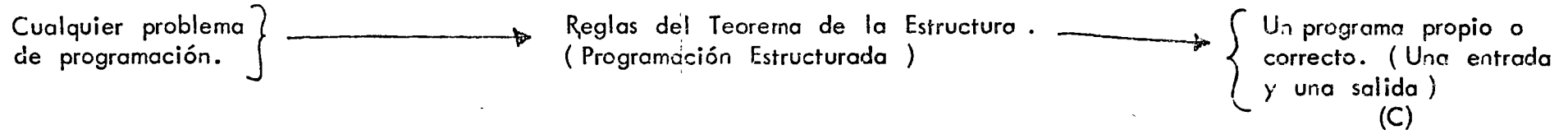


Figure 4. Segmentation

A graphic example of a program constructed of such units is shown in Figure 5. At the top are the job control and linkage editor statements that define the environment and major functions of the program. Subordinate to them is the hierarchy, or calling sequence, of the supporting units of code. Each unit specifies the invocation of the units immediately subordinate to it. Thus unit A would invoke units B and J, using COPY, CALL, PERFORM or %INCLUDE

statements; unit B would invoke C and F; unit C would invoke D and E; etc. The next technique to be described, top-down program development, assumes such a hierarchical structure.

¹ Bohm, C., and Jacopini, G., "Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules." *Communications of the ACM* 9, No. 3 (May 1966), 366-371.



1.- Usar solamente las siguientes estructuras lógicas (A) :

- Secuencia de una o más operaciones. (SEQUENCE)
- Bifurcación a una de dos operaciones y regreso. (IF THEN ELSE)
- Repetición de una operación mientras una condición sea verdadera. (DO WHILE)

2.- Combinar las estructuras (B) :

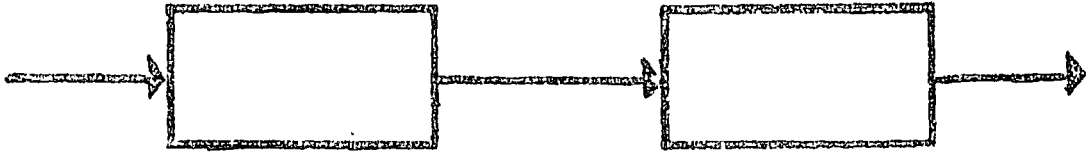
- Sustituyendo una por otra
- Anidando una dentro de otra

(A) : Representación gráfica de las estructuras en Fig. 1.20

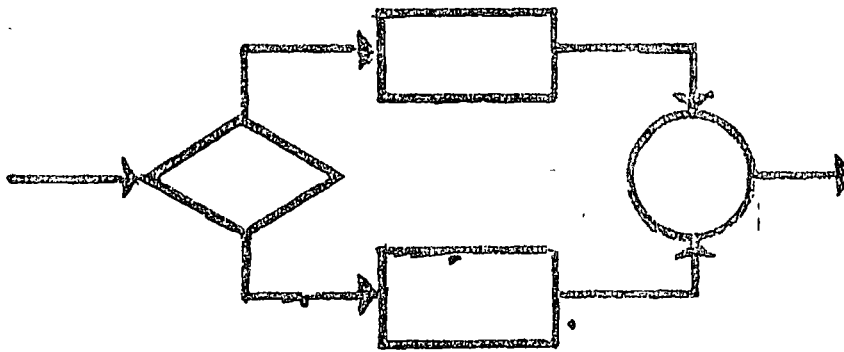
(B) : Ejemplo de combinación en Fig. 1.30

(C) : Características de un programa propio en Fig. 1.40

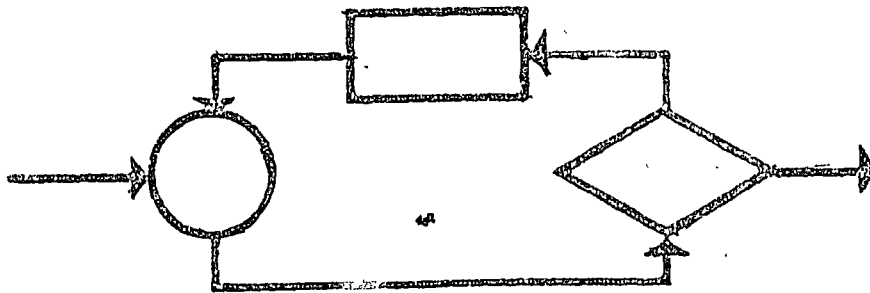
(5)



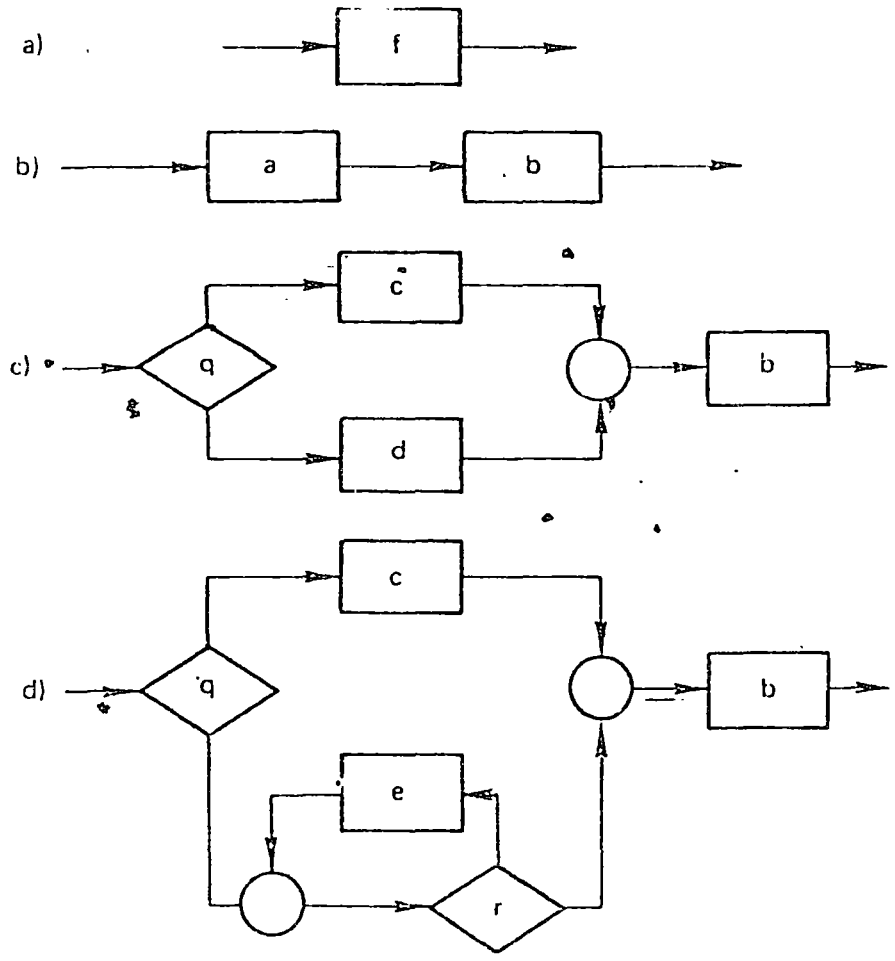
SEQUENCE



IF THEN ELSE



DOWHILE



11000
 11000
 11000
 11000

5

Fig. 1.30

CARACTERISTICAS DE UN PROGRAMA PROPIO.

- UNA ENTRADA Y UNA SALIDA

- FACILIDAD DE LECTURA
DE ARRIBA HACIA ABAJO

- NUNCA ROMPE SU SECUENCIA
NO HAY BIFURCACIONES INCONDICIONALES

- FACIL DE ENTENDER
SOLO CONTIENE ESTRUCTURAS LOGICAS

- FACILIDAD DE MANTENIMIENTO
REGLAS ESTABLECIDAS

EJERCICIOS DE PROG.
ESTRUCTURADA.

Diagrama estructurado.
Fig. 1.220

Ej. 1.- Totalizar el número de estructuras de
cada tipo mostradas en el diagrama.

Total de figuras de cada
tipo encontradas.
Solución en Fig. 1.320

Diagrama de bloque.
Fig. 1.230

Ej. 2.- Convertir el diagrama en estructurado.

Diagrama estructurado.
Solución en Fig. 1.330

Diagrama de bloque.
Fig. 1.240

Ej. 3.- Conversión del diagrama en estructurado.

Diagrama estructurado.
Solución en Fig. 1.340

⊙

TOTAL DE ESTRUCTURAS

SEQUENCE = 13

A A M

IF THEN ELSE = 6

P, Q, S, T, V, W

DO WHILE = 2

R, U.

10

Fig. 1.320

PRŒBLEMA

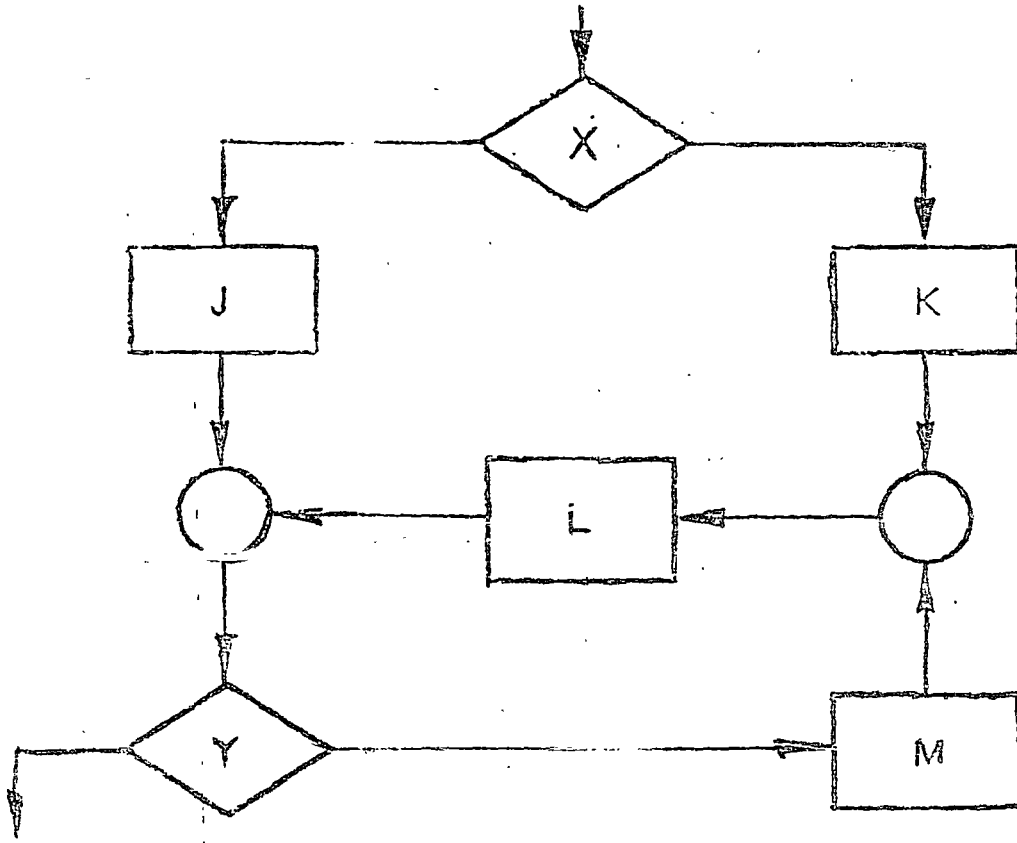


Fig 1.330

SŒLUCIŒN

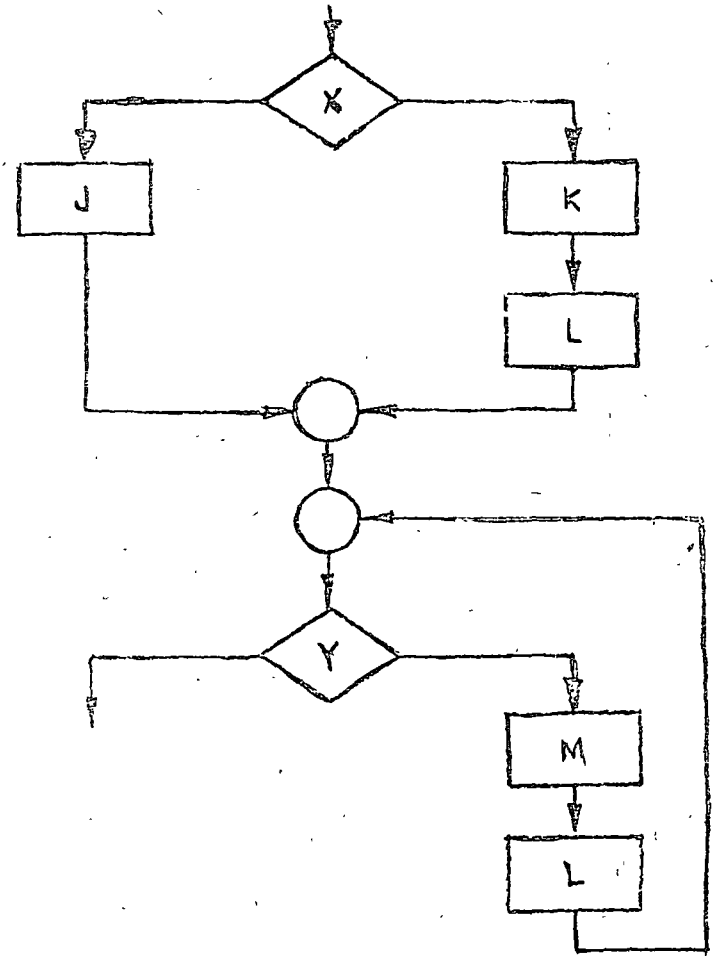


Fig 1.330

10-11

PROBLEMA

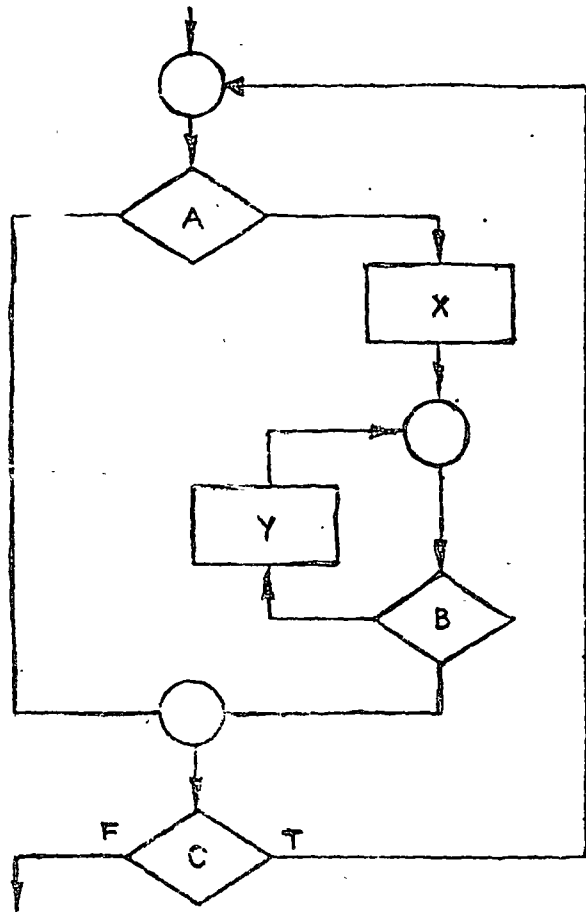


Fig 1.240

SOLUCIÓN

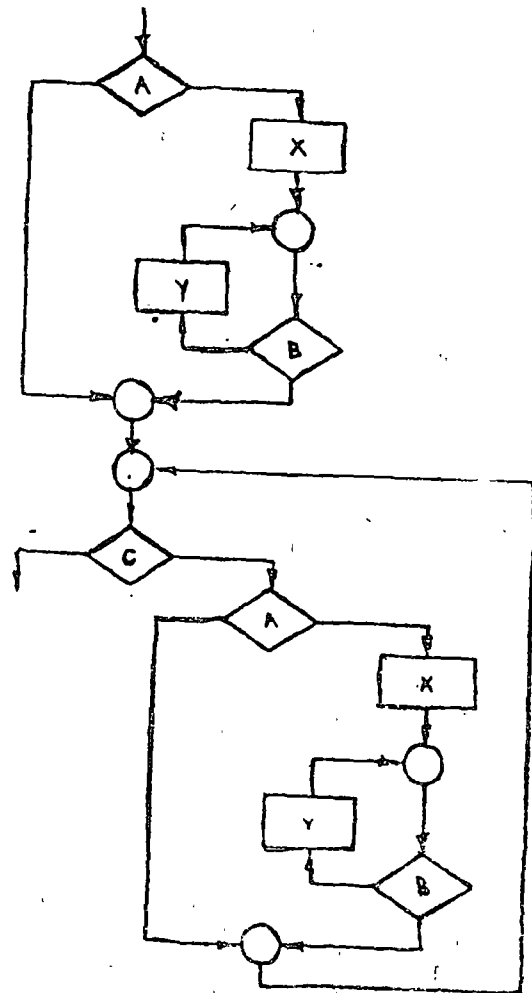


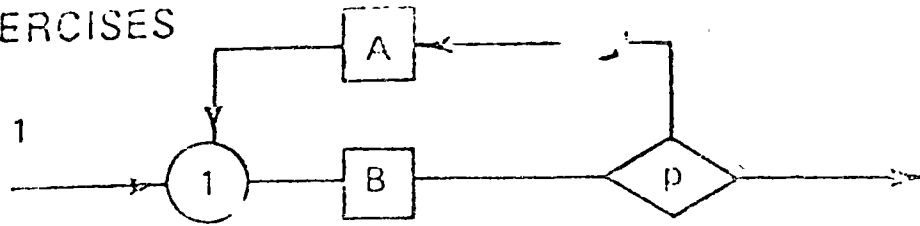
Fig 1.340

12-13

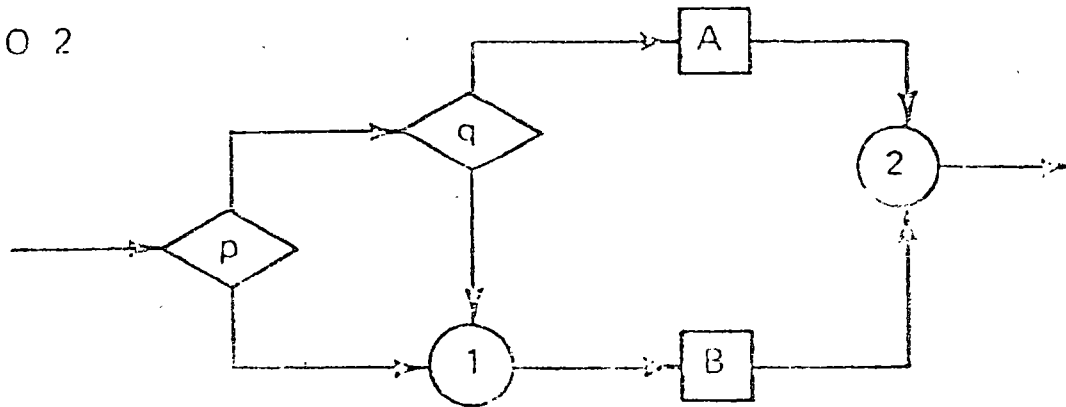
EXERCISES

Cambiar a Estructurado

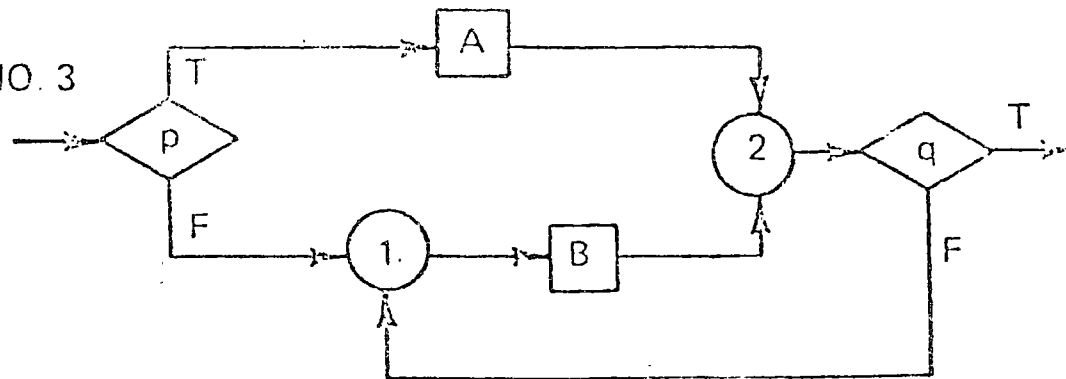
NO. 1



NO 2



NO. 3





centro de educación continua
división de estudios superiores
facultad de ingeniería, unam



MÉTODOS NUMÉRICOS Y APLICACIONES
CON LA COMPUTADORA DIGITAL

COMPLEMENTO A LAS RAÍCES
DE FUNCIONES

ING. HORACIO SANDOVAL R.

MAYO, 1978

RAICES DE FUNCIONES

(TRASCENDENTES Y POLINOMIALES)

FUNCIONES.

Una función es la correspondencia de cualquier elemento de un conjunto A a un elemento único de un conjunto B.

$$f: A \rightarrow B$$

Ejemplos:

$$y = 3x + 2$$

$$y = \frac{1}{2}ax^2 + vx + x_0$$

$$y = \text{Sen } x$$

$$y = \text{Sen } 3x + \ln x + e^{2x}$$

Las funciones se dividen en dos grandes grupos

- las polinomiales o Polinomios.
- las trascendentes.

Cuando una función se iguala a cero, $f(x) = 0$, los valores que satisfacen dicha ecuación se llaman raíces.

Sea el polinomio $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + ax + a_0$.

n es el mayor exponente e indica el grado del polinomio. Siempre es un número entero y finito
Todos los coeficientes a_i son reales.

En cualquier polinomio del tipo descrito, se cumplen las siguientes características:

- 1) Habrá n raíces. (del teorema fund. del Algebra)
- 2) Si n es impar, habrá cuando menos una raíz real cuyo signo es opuesto al del término constante.
- 3) Si n es par y el término constante es negativo, tiene como mínimo dos raíces reales, una positiva y otra negativa.
- 4) Se satisface la regla de Descartes.
- 5) En caso de existir raíces complejas, estas se presentarán en pares conjugados.

Regla de Descartes.

El número de raíces positivas es igual al número de cambios de signo de los coeficientes, o es menor a este número en una cantidad

Par.

Ejemplos. $4x^3 - 24x^2 + 23x + 18 = 0$ $-\frac{1}{2}, 2, \frac{9}{2}$

$x^4 - 16x^3 + 86x^2 - 176x + 105 = 0$ $1, 3, 5, 7.$

$6x^4 - 29x^3 + 40x^2 - 7x - 2 = 0$ $\frac{1}{3}, \frac{3}{2}, 1 \pm \sqrt{2}$

$x^4 - 2x^3 - 21x^2 + 22x + 40 = 0$ $-4, -1, 2, 5$

Recordatorio.

Sean los números complejos

$$\left. \begin{aligned} z_1 &= x_1 + i y_1 \\ z_2 &= x_2 + i y_2 \end{aligned} \right\} \text{ donde: } i = \sqrt{-1} \therefore i^2 = -1$$

z_1 y z_2 serán conjugados si cumplen con

$$x_1 = x_2$$

$$y_1 = -y_2$$

Ejemplo.

$$x^5 - x^4 + 8x^2 - 9x - 15 = 0$$

$$-1, \sqrt{3}, -\sqrt{3}, \underbrace{1+2i, 1-2i}$$

Complejas y conjugados.

Es posible encontrar raíces repetidas en cualquier caso.

Partiendo de la equivalencia

$$P_n(x) = (x - \lambda_1)(x - \lambda_2) \cdots (x - \lambda_n)$$

se llega a las siguientes relaciones entre los coeficientes y las raíces de una ecuación. ($a_n = 1$)

$$-a_{n-1} = \text{Suma de las raíces}$$

$$a_{n-2} = \text{Suma de los productos de las raíces tomados de dos en dos.}$$

$$-a_{n-3} = \text{Suma de los productos en grupos de tres.}$$

⋮

⋮

$$(-1)^n a_0 = \text{Producto de todas las raíces.}$$

Es fácil ver que de este conjunto de ecuaciones la primera y la última son las más simples y rápidas de aplicar

Ejemplos.

$$= x^4 - 2x^3 - 21x^2 + 22x + 40 = 0 \quad -4, -1, 2, 5$$

$$\sum \lambda_i = -4 - 1 + 2 + 5 = 2 = -a_{n-1}$$

$$\prod \lambda_i = (-4)(-1)(2)(5) = 40 = (-1)^n a_0$$

$$= x^4 - 16x^3 + 86x^2 - 176x + 105 = 0 \quad 1, 3, 5, 7$$

$$\sum \lambda_i = 1 + 3 + 5 + 7 = 16 = -a_{n-1}$$

$$\prod \lambda_i = (1)(3)(5)(7) = 105 = (-1)^n a_0$$

La importancia de conocer estas características surge cuando se quieren comprobar los resultados obtenidos por un programa, o cuando se necesita dar a la máquina el número de raíces por calcular, el intervalo o algún valor inicial.

Una vez definidas algunas características de las raíces de los polinomios, veamos varios de los métodos o algoritmos que existen para localizarlos empleando las computadoras digitales.

METODO DE HORNER.

Este método nos permite evaluar de manera rápida y eficiente una función polinomial para un valor dado de la variable independiente.

$$P_3(x) = a_3x^3 + a_2x^2 + a_1x + a_0 = \{[(a_3)x + a_2]x + a_1\}x + a_0.$$

Cuando el polinomio es de orden n , en la presentación tradicional se tienen n Sumas y $\frac{1}{2}(n+1)n$ multipli. y en la forma encajada o de Horner se tienen n sumas y n multiplicaciones.

Sea $n = 10$

$$\begin{array}{r} 10 \text{ Sumas} \\ 55 \text{ Mult.} \\ \hline 65 \text{ Oper.} \end{array}$$

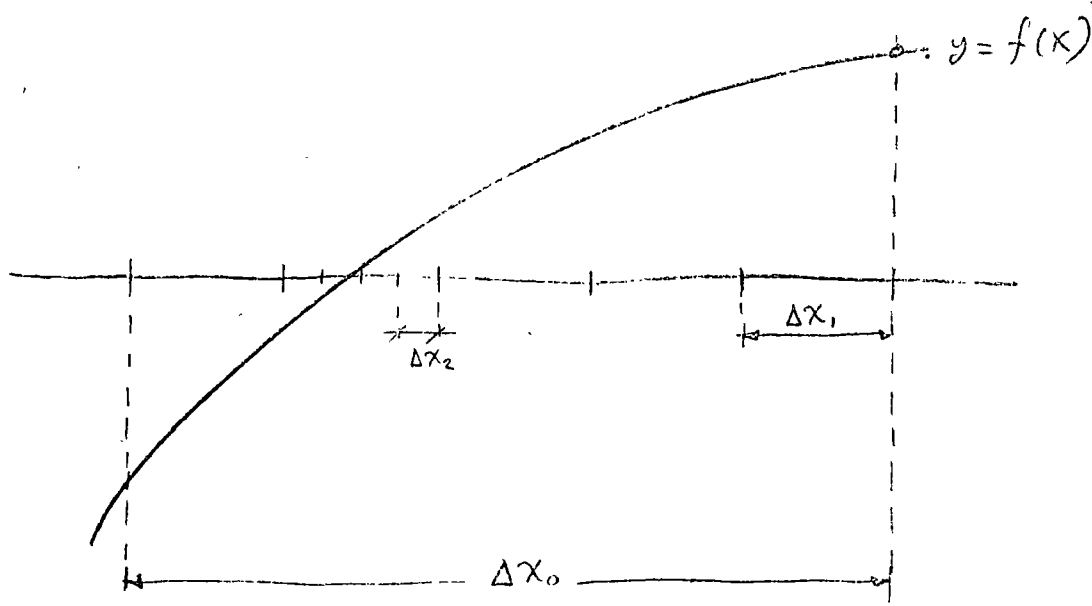
$$\begin{array}{r} 10 \text{ Sumas} \\ 10 \text{ Multip.} \\ \hline 20 \text{ Oper.} \end{array}$$

METODO DE TANTEOS.

En este método se valúa la función $f(x)$ para valores sucesivos de x hasta el momento en que se presenta un cambio de signo en $f(x)$, o sea que la función ha pasado por una raíz. De aquí se puede obtener mayor aproximación regresando al valor de x que precede el cambio de signo, y a partir de este valor, calcular otra vez los valores de $f(x)$ correspondientes a valores sucesivos de x ,

pero empleando ahora un incremento menor que el usado antes, hasta detectar el cambio de signo de la función $f(x)$.

Este procedimiento se repite con incrementos cada vez mas pequeños hasta lograr un valor suficientemente aproximado de la raíz.



Notas:

- Cuando se ha localizado un intervalo donde existe una sola raíz, no hay problema, se presenta el caso mostrado en la figura
- Si se parte unicamente de la función, sin otro conocimiento adicional, se debe tener cuidado con

al valor inicial seleccionado, ya que se pueden dejar raíces a la izquierda; suponiendo que nuestro barrido es hacia la derecha.

- Otro parámetro peligroso es el valor del incremento inicial, ya que en uno solo pueden pasar dos raíces, o en general un número par. Si por otro lado, se selecciona el incremento inicial muy pequeño, se consume mucho tiempo de procesamiento.

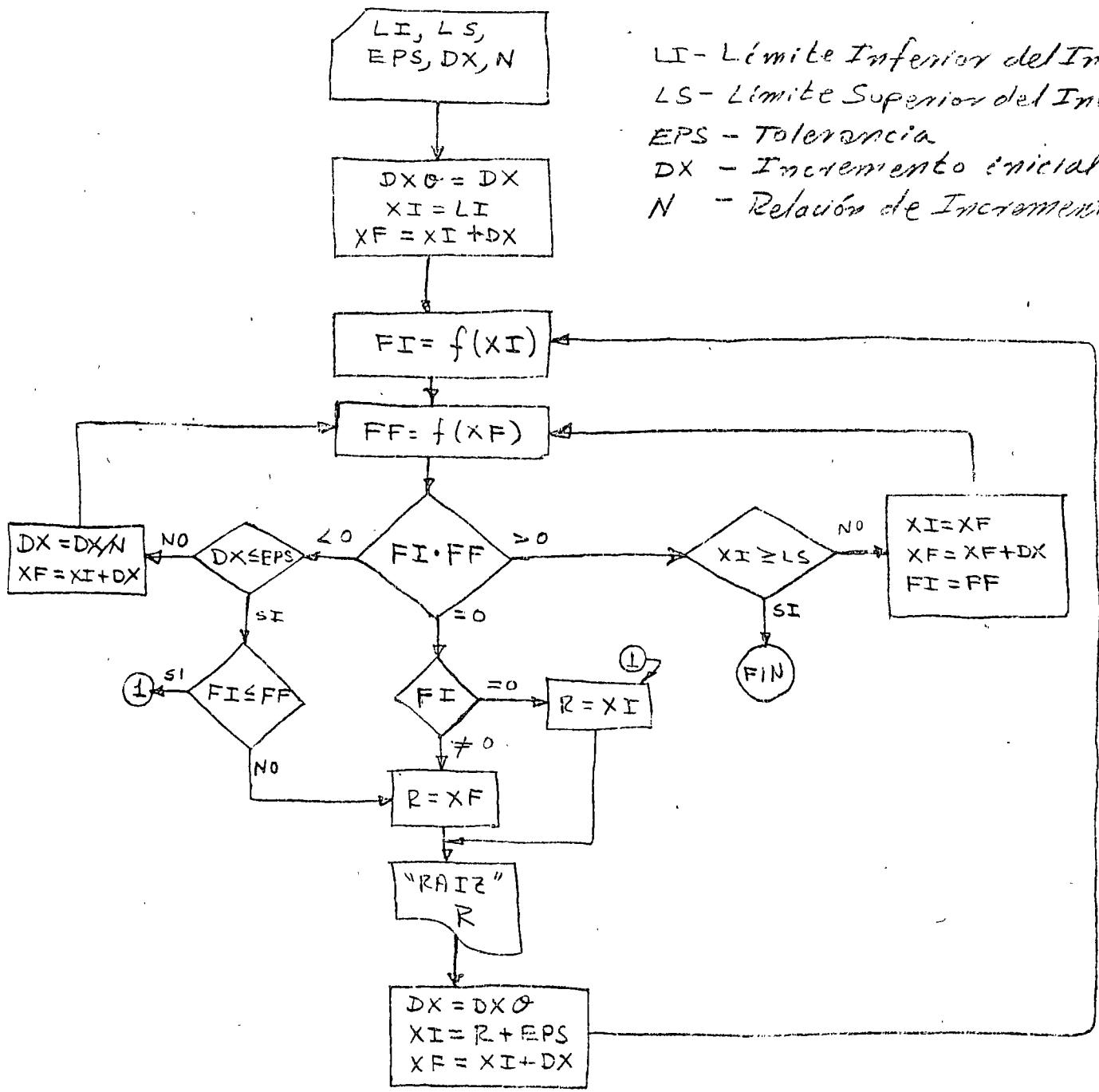
De aquí la importancia de obtener una gráfica aproximada de la función o, cuando es posible, acotar su comportamiento de acuerdo con algunas propiedades matemáticas.

Por tanto, los datos iniciales para aplicar este método pueden ser

- a) Valor inicial de la variable independiente o intervalo por analizar.
- b) Valor del incremento inicial
- c) Tolerancia para aceptar un valor de x como la raíz buscada.
- d) Proporción de un incremento con respecto al siguiente.

METODO DE TANTEOS PARA DETECTAR TODAS LAS RAICES CONTENIDAS EN UN INTERVALO DADO.

LI - Límite Inferior del Int.
 LS - Límite Superior del Int.
 EPS - Tolerancia
 DX - Incremento inicial
 N - Relación de Incrementos.



Se puede agregar una tolerancia en la función, o sea, cuando el cruce por cero es muy rápido (pendiente alta) una tolerancia dada en x , puede dar valores aún significativos en $y = f(x)$.

Si aceptamos que $N=5$ y el incremento se divide 7 veces el valor final está dado por.

# Cambios	veces el incremento inicial
1	$1/5 = 5^{-1}$
2	$= 25^{-1}$
3	$= 125^{-1}$
4	$= 625^{-1}$
5	$= 3125^{-1}$
6	$= 15625^{-1}$
7	$= 78125^{-1}$

o sea que la aproximación que se puede lograr es alta.

METODO DE BISECCION.

Es un caso particular del método de Tanteos, según el enfoque presentado, ya que cuando $N=2$, en principio se tiene el método de Bisección. Es decir, consiste en dividir el intervalo en dos partes iguales

$$\Delta x = \frac{\Delta x}{2}$$

Este método localiza una raíz, dado el intervalo que la contiene.

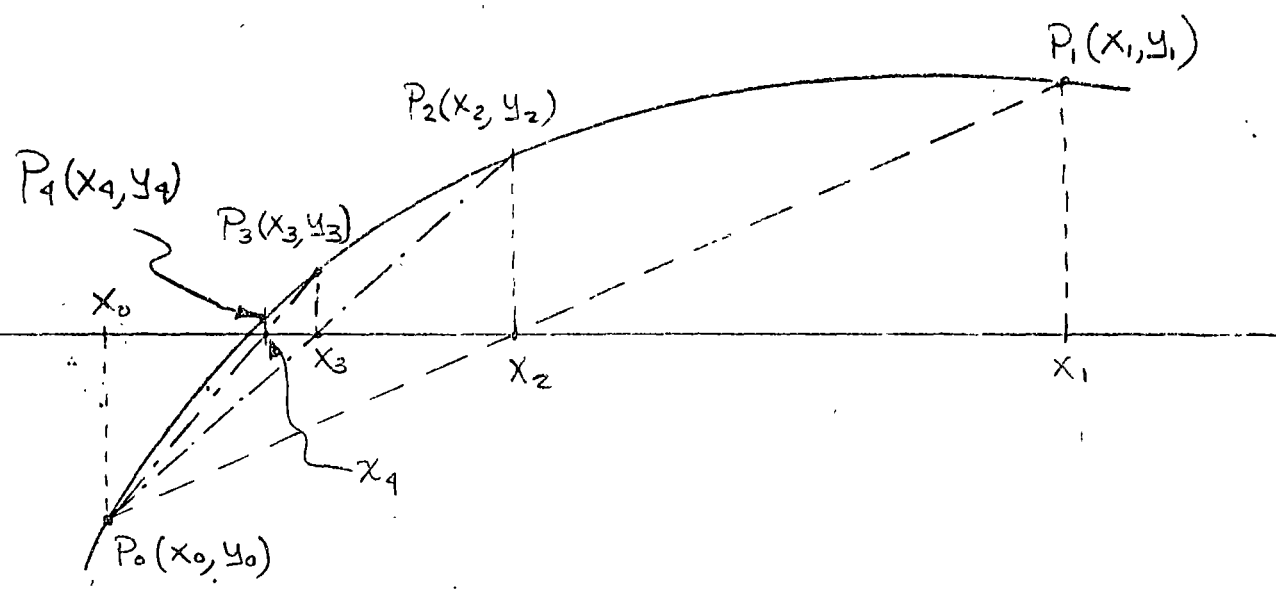
En cada paso el intervalo se divide entre dos, con lo que en 10 pasos se llega a $2^{10} = 1024 \approx 1000$, es decir un milésimo del intervalo original. Con 20 y 30 pasos se llegará a $\Delta x / 1000000$ y $\Delta x / 1000000000$ respectivamente.

¿Que método es mejor? ¿El de Tanteos o el de Bisección?

Esta pregunta simplista induce a pensar en el método por sí mismo, y a olvidar qué se busca y para qué. Evidentemente aún no existe "EL METODO" que funcione en todos los casos y satisfaga todos los propósitos.

METODO DE INTERPOLACION LINEAL
(REGULA FALSI o DE LAS CUERDAS)

Cuando se ha definido el intervalo que contiene una raíz, se puede aplicar este método. Para funciones trascendentes con discontinuidades en la primera derivada o muy complejas, este método unido al de tanteos es, generalmente muy eficiente.



Los pasos a seguir son:

a) Se define donde la recta que une P_0 y P_1 cruza el eje de los x , en la figura es x_2 .

Mediante triángulos semejantes se tiene

$$\frac{y_1 - y_0}{x_1 - x_0} = \frac{0 - y_0}{x_2 - x_0}$$

$$\therefore x_2 = \frac{x_0 y_1 - x_1 y_0}{y_1 - y_0}$$

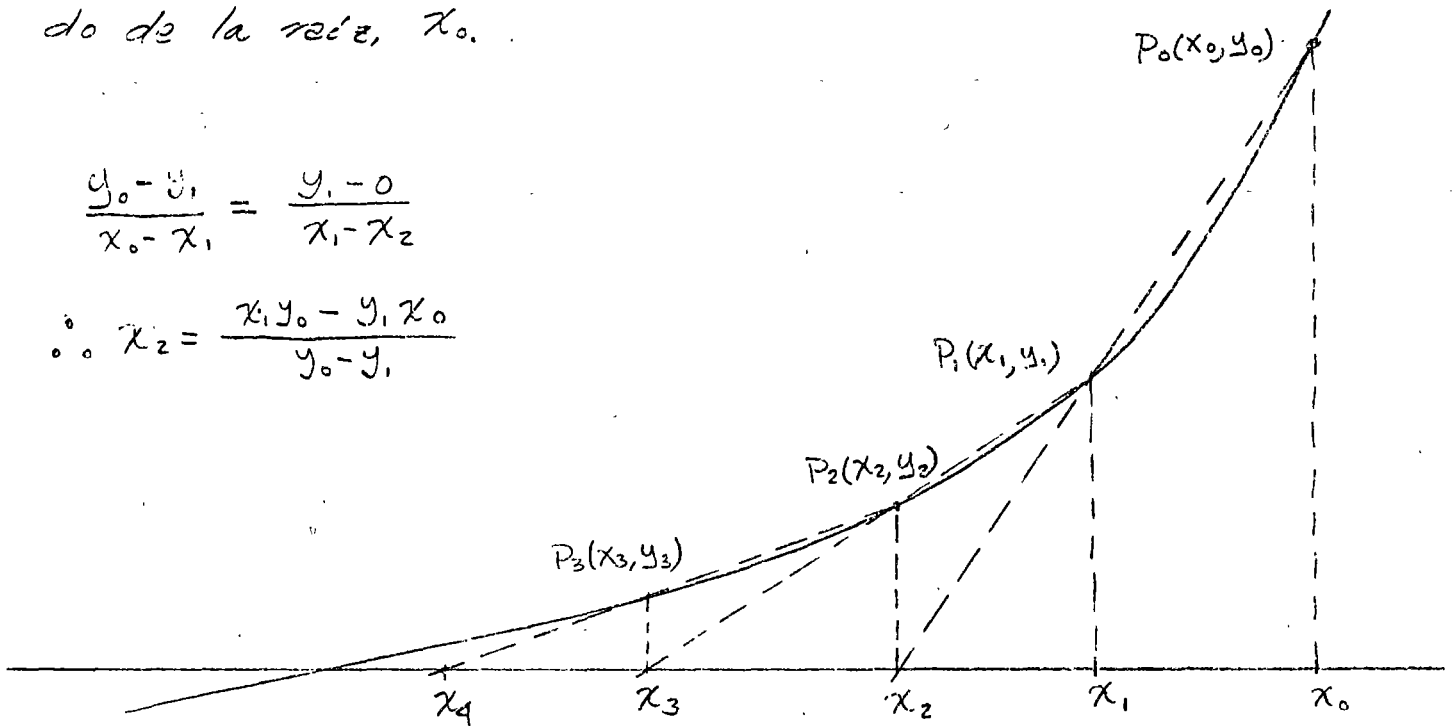
- b) Se calcula $y_2 = f(x_2)$
- c) Se analiza en que segmento quedó la raíz, en este caso está entre x_0 y x_2
- d) Se define un nuevo punto de cruce, de la recta que va de P_0 a P_2 , y el eje de las x .
- e) Se repiten los pasos anteriores hasta satisfacer la tolerancia solicitada.

METODO DE LA SECANTE.

Se aplica cuando solo se conoce un valor aproximado de la raíz, x_0 .

$$\frac{y_0 - y_1}{x_0 - x_1} = \frac{y_1 - 0}{x_1 - x_2}$$

$$\therefore x_2 = \frac{x_1 y_0 - y_1 x_0}{y_0 - y_1}$$



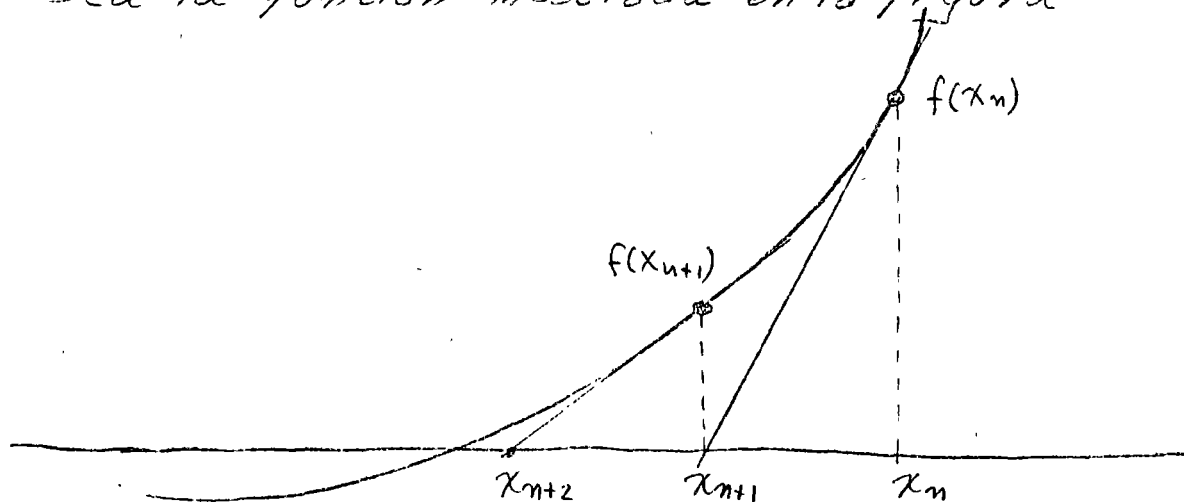
Pasos a seguir:

- Se selecciona un valor x_1 cercano a x_0 y se calcula la función para ese valor. Si $f(x_1) \cdot f(x_0) < 0$, se tiene el caso de interpolación lineal, por tanto aceptemos que $f(x_1) \cdot f(x_0) > 0$
- Se extrapola linealmente mediante la secante para definir la abscisa x_2 , más próxima a la raíz.
- Se evalúa la función en x_2 para definir $P_2(x_2, y_2)$.
- Con P_1 y P_2 se vuelve a extrapolar por la secante para definir x_3 .
- Se repiten los pasos anteriores hasta satisfacer la tolerancia solicitada.

METODO DE NEWTON-RAPHSON.

Este método es muy útil para mejorar una primera aproximación. (obtenida por tanteos, gráficamente o análisis numérico al evaluar en $-\infty, -1, 0, 1, \infty$)

Sea la función mostrada en la figura



Este método hace una extrapolación lineal mediante la tangente $f'(x_n)$, la cual está dada por

$$f'(x_n) = \frac{f(x_n)}{x_n - x_{n+1}}$$

$$\therefore x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

donde x_{n+1} es una mejor aproximación de la raíz. Se requiere por tanto que se conozcan el valor de la función y de su derivada en los puntos x_n . Cuando existe la solución converge rápidamente si se cumplen las siguientes condiciones:

- a) x_n debe estar suficientemente cerca de una raíz.
- b) $f'(x)$ no debe estar muy cerca de cero, o sea, dos raíces no deben estar muy próximas entre sí.
- c) $f''(x)$ no debe ser excesivamente grande, o sea, la curvatura de $f(x)$ no debe ser muy fuerte.

METODO DE NEWTON DE SEGUNDO ORDEN.

Cuando se requiere determinar con mucha precisión el valor de la raíz de una ecuación, se emplea este método, que tiene la ventaja de convergir rápidamente a la solución. Pero por razones prácticas solo puede aplicarse a ecuaciones cuyas primera y segunda derivadas son relativamente sencillas, ya que el tiempo que se consume en obtener dichas derivadas puede sobrepasar la ventaja de la convergencia rápida.

Desarrollando $f(x)$ con respecto a x , en serie de Taylor se tiene

$$f(x_{n+1}) = f(x_n) + f'(x_n)\Delta x + \frac{f''(x_n)\Delta x^2}{2!} + \frac{f'''(x_n)\Delta x^3}{3!} + \dots$$

Tomando solo los tres primeros términos e igualando a cero, se tiene

$$f'(x_n) + \Delta x \left[f''(x_n) + \frac{f''(x_n) \Delta x}{2} \right] = 0$$

y dado que $\Delta x = x_{n+1} - x_n = -f(x_n)/f'(x_n)$ se llega a

$$x_{n+1} = x_n - \left[\frac{f(x_n)}{f'(x_n) - \frac{f''(x_n)f(x_n)}{2f'(x_n)}} \right]$$

que se cumple de manera recursiva hasta satisfacer la aproximación deseada.

METODO DE APROXIMACIONES SUCESIVAS.

Sea la función $F(x)=0$, la cual puede reescribirse como $x = f(x)$ al aumentar x en ambos miembros ($f(x) = F(x) + x$).

Las raíces de ambas ecuaciones son las mismas.

El método de las aproximaciones sucesivas parte de una aproximación inicial x_0 , y de aquí se inicia el proceso calculando

$$x_1 = f(x_0)$$

y después $x_2 = f(x_1)$

y en el caso general se tiene $x_n = f(x_{n-1})$

donde la secuencia $\{x_n\}$ converge a la solución.

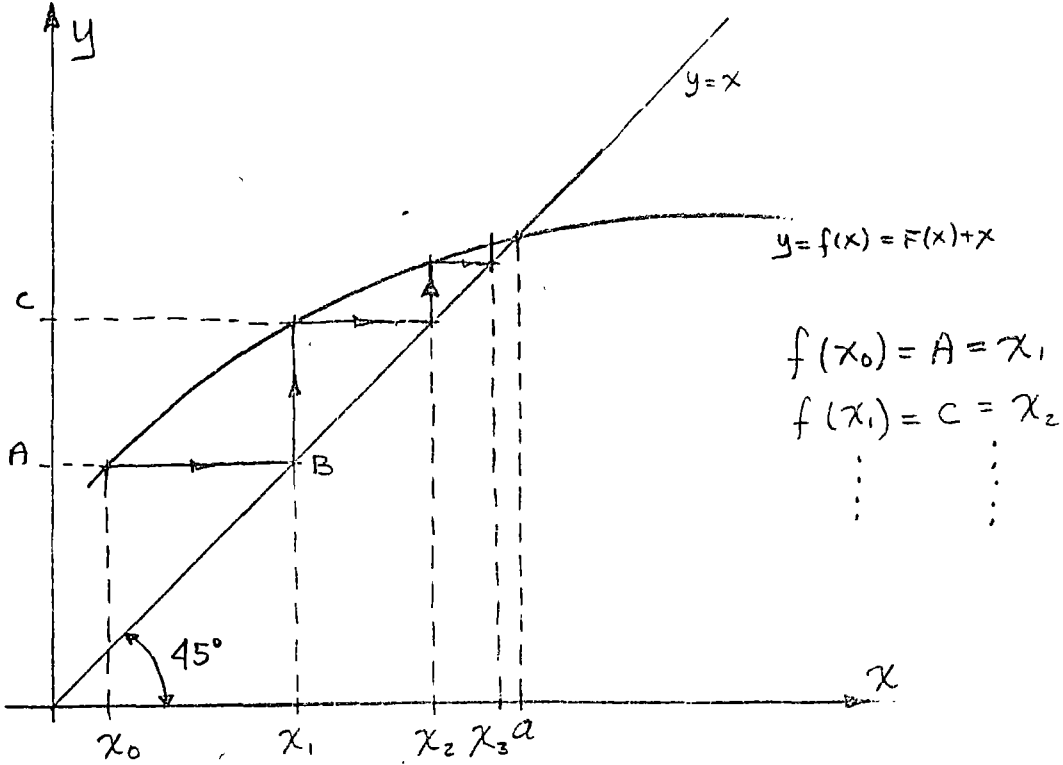
Solo resta indicar en que condiciones $f(x)$ converge.

Para el intervalo $a \leq x \leq b$ el método de aproximaciones sucesivas converge cuando, en todo el intervalo $|f'(x)| < 1$, o sea, el valor absoluto de la primera derivada es menor que la unidad.

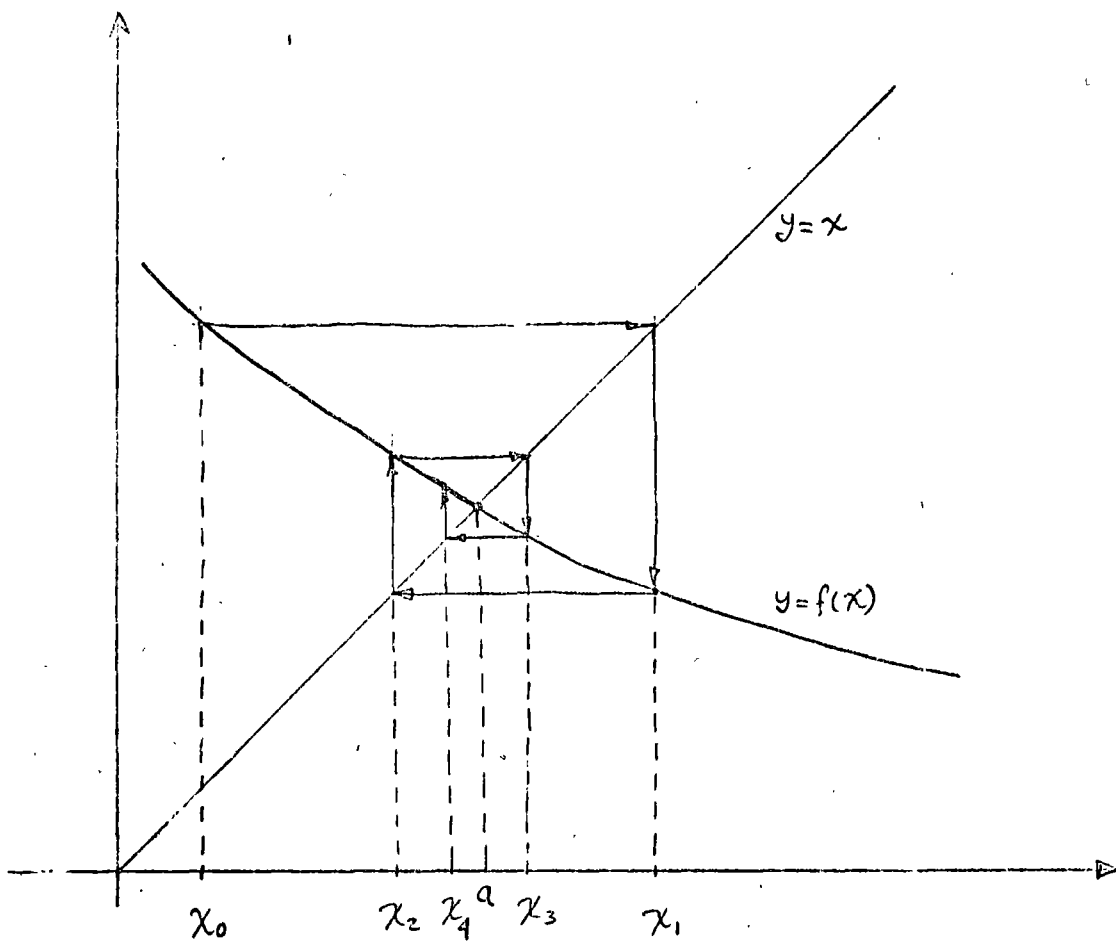
Y cuando el valor absoluto de la primera derivada es mayor que uno, $|f'(x)| > 1$, el proceso diverge.

Si en el intervalo analizado se tienen pendientes mayores y menores a la unidad, el proceso puede o no convergir.

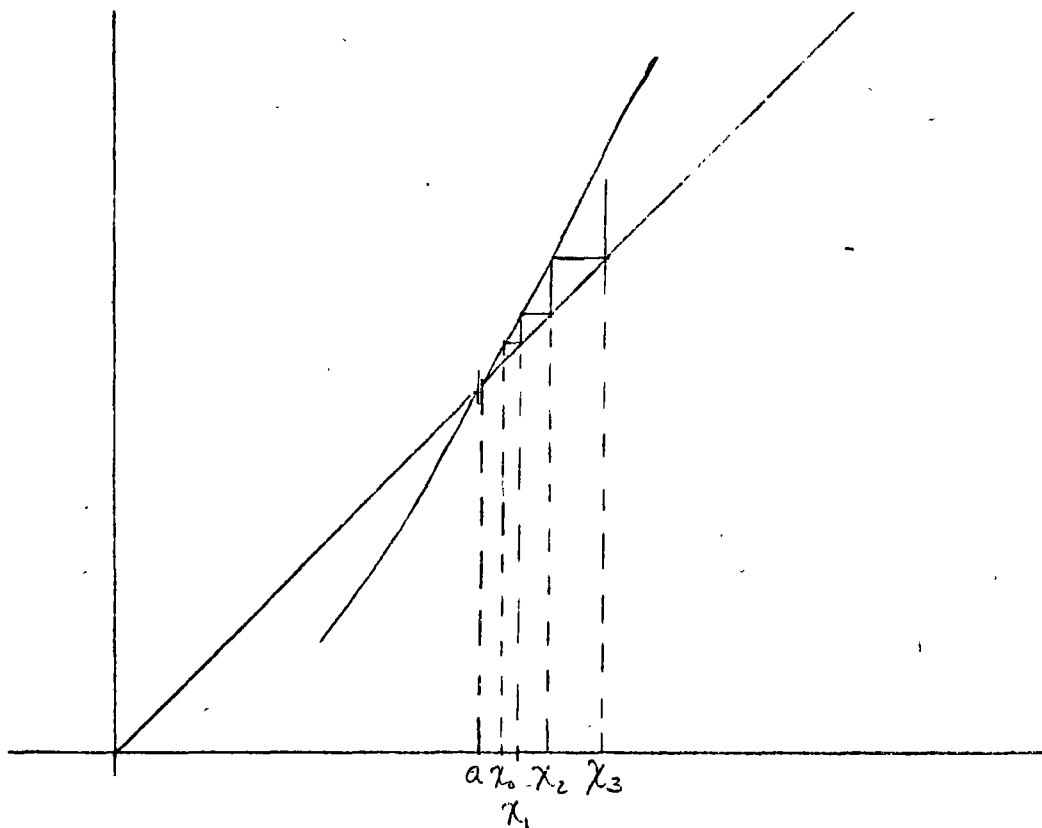
Veamos estos comportamientos gráficamente:



Caso convergente. $0 < f'(x) < 1$



Caso Convergente $-1 < f'(x) < 0$



Caso divergente $f'(x) > 1$

Hasta este punto, todos los métodos vistos, son aplicables a cualquier función monovaluada y continua, o sea funciones trascendentes y polinomiales.

METODO DE LA DIVISION SINTETICA ITERADA (Polinomios)

Sea

$$f(x) = x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n$$

factorizando se tiene

$$f(x) = (x-z)(x^{n-1} + b_1 x^{n-2} + \dots + b_{n-2} x + b_{n-1}) + R.$$

Cuando evaluamos esta función $f(x)$ con $x=z$ se tiene

$$f(z) = R.$$

Factorizando ahora el polinomio de coeficientes b , se tiene

$$x^{n-1} + b_1 x^{n-2} + \dots + b_{n-2} x + b_{n-1} = (x-z)(x^{n-2} + c_1 x^{n-3} + \dots + c_{n-2}) + R'$$

de manera que

$$f(x) = (x-z)^2 (x^{n-2} + c_1 x^{n-3} + \dots + c_{n-2}) + (x-z) R' + R$$

y se llega a $R' = f'(z)$ donde $f' = \frac{df(x)}{dx}$.

Iguando los coeficientes de x^n , se tiene.

$$b_0 = 1$$

$$b_k = a_k + z b_{k-1} \quad k=1, 2, \dots, n$$

$$b_n = f(z) = R = a_n + z b_{n-1}$$

$$c_0 = 1$$

$$c_k = b_k + z c_{k-1} \quad k=1, 2, \dots, n-1$$

$$c_{n-1} = f'(z) = R'$$

Empleando los valores de R y R' , tenemos

$$z_{i+1} = z_i - \frac{R}{R'}$$

donde z_{i+1} es una mejor aproximación de la raíz.
Cuando $R=0$, $z_{i+1} = z_i = \text{Raíz}$.

Ejemplo:

Encontrar la raíz real de $x^3 = 3.7x^2 + 6.25x - 4.069$
después de dos iteraciones y obtener el polinomio reducido,
tomando $z_0 = 1.25$.

Es útil formar los coeficientes como tabla

1	1	1
a_1	b_1	c_1
a_2	b_2	c_2
⋮	⋮	⋮
a_{n-2}	b_{n-2}	c_{n-2}
a_{n-1}	b_{n-1}	R'
a_n	R	

En nuestro ejemplo

$$\begin{array}{c|c|c} 1 & 1 & 1 \\ -3.7 & -2.45 & -1.20 \\ 6.25 & 3.1875 & 1.6875 \\ -4.069 & -0.08463 & \end{array}$$

$$b_1 = a_1 + z b_0 = -3.7 + (1.25) \cdot 1 = -2.45$$

$$b_2 = a_2 + z b_1 = 6.25 + (1.25)(-2.45) = 3.1875$$

$$b_3 = R = a_3 + z b_2 = -4.069 + (1.25)(3.1875) = -0.08463$$

$$c_1 = b_1 + z c_0 = -2.45 + (1.25) \cdot 1 = -1.20$$

$$R' = c_2 = b_2 + z c_1 = 3.18 + (1.25)(-1.20) = 1.6875$$

$$z_1 = z_0 - \frac{R}{R'} = 1.25 - \frac{-0.08463}{1.6875} = 1.30015$$

2ª Iteración

1	1	1
-3.7	-2.39985	-0.0997
6.25	3.12984	3.00022
-4.069	0.00026	

$$\therefore z_2 = z_1 - \frac{R}{R'} = 1.30015 - \frac{0.00026}{3.00022} = 1.30006$$

El polinomio reducido ya está dado con los valores de b :

$$x^2 - 2.39985x + 3.12984.$$

Seguendo en esta línea existen los métodos de Lin y de Bairstow que fundamentalmente se apoyan en la factorización del Polinomio en factores de primer y segundo orden.

Otro método famoso, pero poco confiable, para polinomios es el de Graeffe, y evidentemente existen muchos otros además de los aquí indicados.

INTERPOLACION

Cuando se tiene un conjunto de datos, dados por x y $f(x)$, a veces se requiere conocer la función para un cierto valor x^* , es decir $f(x^*)$, donde x^* no coincide con ninguno de los datos. Entonces se debe generar el valor $f(x^*)$ tomando en cuenta los ya existentes.

Cuando x^* está contenido entre los valores máximo y mínimo de las x , de los datos, la generación del nuevo valor se hace a través de Interpolación.

En el caso que x^* esté fuera de dicho rango, el nuevo valor $f(x^*)$ se genera mediante Extrapolación.

Debe sentarse claramente la diferencia entre Interpolación y Ajuste de Curvas. Este último concepto consiste en definir una función o una curva CONTINUA, a partir de un conjunto de puntos $(x, f(x))$, lo cual puede lograrse de muy diversas maneras, desde el trazo a mano alzada, hasta la definición analítica de la función que debe contener todos los puntos.

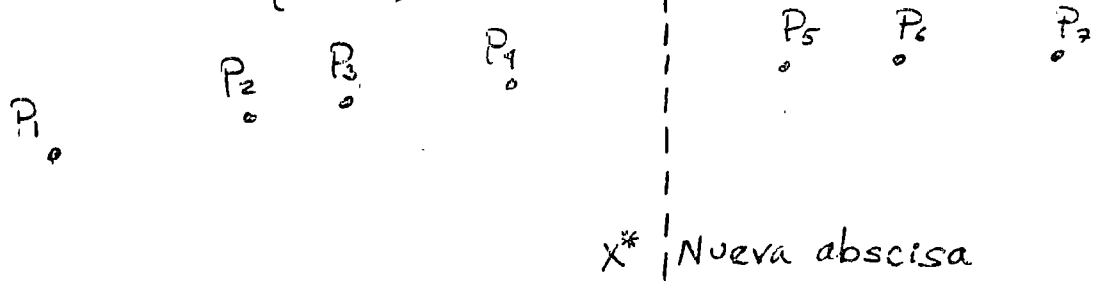
Ya dentro de ajuste de curvas, existen medios para comprobar o comparar si la función $f_a(x)$ ajustó mejor que la $f_b(x)$.

El más famoso de todos ellos, es probablemente el criterio de MINIMOS CUADRADOS.

Es importante la curva ajustada porque al interpolar, IMPLICITAMENTE se ajusta una curva.

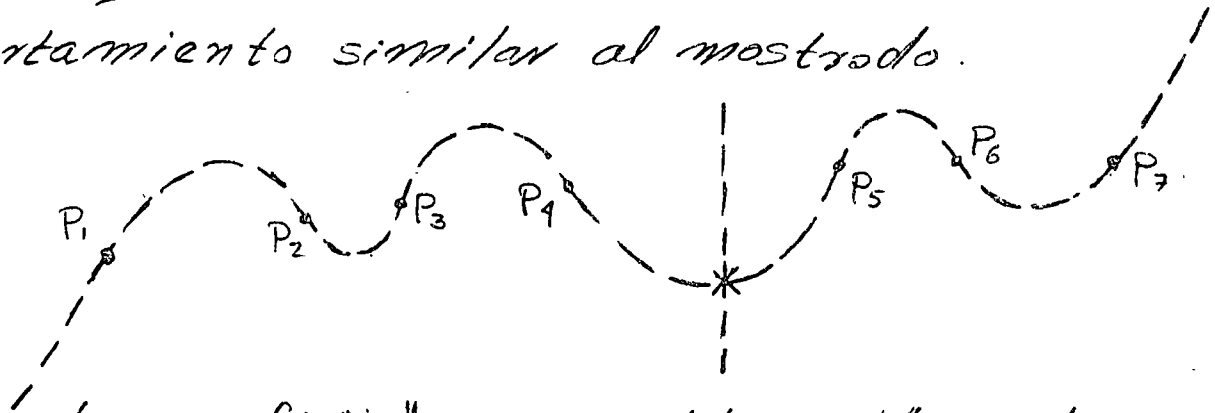
La mejor ayuda para seleccionar la familia de curvas de ajuste, y por tanto, efectuar una interpolación adecuada, es el comportamiento de la función; que solo conocemos en puntos aislados. Este es el problema medular, porque al no tener idea del comportamiento de la función, lo mismo da aplicar un método u otro, o sea, independientemente de los valores calculados, no se podrá confiar plenamente en ellos.

Ejemplo: Sea un conjunto de 7 datos (puntos) y se desea calcular el valor de la función para un octavo dato (cruz).



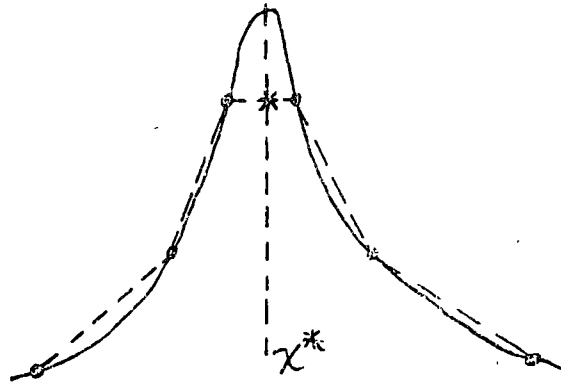
Al ser 7 datos podemos ajustar un polinomio de 6° grado para usar "toda la información disponible"

Seguramente el polinomio tendrá un comportamiento similar al mostrado.



con lo que $f(x^*)$, "muy probablemente" quede mal ubicado.

Sea ahora una curva típica de Resonancia.



y usemos el método más simple, interpolación lineal. Se ve inmediatamente que no es correcta la ubicación de $f(x^*)$.

Conclusión: NO EXISTE EL METODO.

En el primer ejemplo se "ajustó" un polinomio de 6º orden cuando probablemente una parábola o una exponencial serían preferibles y en el segundo ejemplo otro parábolo (2º Grado) daría resultados muy satisfactorios, mucho mejores que lo recta. Lo que pone en evidencia que la complejidad o sofisticación de los métodos no representan ninguna garantía en los resultados, sin antes haber estudiado el comportamiento del fenómeno o de la función muestreada.

Los métodos de interpolación se dividen en dos grupos:

- a) Los métodos globales que ocupan toda la información. En este tipo cualquier dato interviene en el cálculo de todos los nuevos valores. (Ejemplo 1)
- b) Los métodos locales solo emplean la información que está en la vecindad del punto por generar. (Ejemplo 2)

El método de interpolación más simple es la interpolación lineal.

Sea la recta definida por dos puntos $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$

$$y = \frac{y_1 - y_2}{x_1 - x_2} (x - x_1) + y_1 \quad x_1 \neq x_2$$

cuando se desean calcular muchos valores intermedios es mejor emplear

$$y = mx + n \quad \text{donde: } m = \frac{y_1 - y_2}{x_1 - x_2}$$

$$n = y_1 - mx_1 = \frac{y_2 x_1 - y_1 x_2}{x_1 - x_2}$$

este método es un caso particular de lo que se conoce como

INTERPOLACION DE LAGRANGE.

Sea la siguiente notación:

$$\pi_i(x) = (x-x_1)(x-x_2)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_{n+1})$$

$$\pi_i(x_j) = 0 = (x_j-x_1)\dots(x_j-x_{j-1})(x_j-x_{j+1})\dots(x_j-x_{i-1})(x_j-x_{i+1})\dots$$

$$\pi_i(x_i) = (x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_{n+1}) \neq 0$$

La fórmula para determinar el polinomio de orden n que contiene los $n+1$ puntos dados es

$$P_n(x) = \sum_{i=1}^{n+1} y_i \left[\frac{\pi_i(x)}{\pi_i(x_i)} \right]$$

Ejemplo: $n=1$

$$\begin{aligned} P_n(x) &= y_1 \left[\frac{(x-x_2)}{(x_1-x_2)} \right] + y_2 \left[\frac{(x-x_1)}{(x_2-x_1)} \right] \\ &= \frac{y_1(x-x_2) - y_2(x-x_1)}{x_1-x_2} = \frac{1}{x_1-x_2} (y_1x - y_2x + y_2x_1 - y_1x_2) \\ &= mx + \frac{y_2x_1 - y_1x_2}{x_1-x_2} = mx + h. \end{aligned}$$

que es la fórmula ya localizada.

Antes de hacer un ejemplo numérico, veamos que pasa cuando $x=x_k$, o sea, cuando se desea calcular una ordenada que es un dato.

$$P_n(x_k) = y_1 \left[\frac{\pi_1(x_k)}{\pi_1(x_1)} \right] + y_2 \left[\frac{\pi_2(x_k)}{\pi_2(x_2)} \right] + \dots + y_{n+1} \left[\frac{\pi_{n+1}(x_k)}{\pi_{n+1}(x_{n+1})} \right]$$

el k -ésimo sumando será

$$y_k \left[\frac{\prod_k(x_k)}{\prod_k(x_k)} \right] = y_k \left[\frac{x}{x} \right] = y_k \cdot 1$$

$\therefore P_n(x_k) = y_k$ con lo que verificamos el método.

Sean los puntos $P_1(1,2)$, $P_2(2,3)$ y $P_3(3,1)$. Definir el polinomio de 2º grado que los contiene.

$$P_2(x) = y_1 \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + y_2 \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + y_3 \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}$$

$$P_2(x) = 2 \frac{(x-2)(x-3)}{(1-2)(1-3)} + 3 \frac{(x-1)(x-3)}{(2-1)(2-3)} + 1 \frac{(x-1)(x-2)}{(3-1)(3-2)}$$

$$P_2(x) = 2 \frac{x^2 - 5x + 6}{2} + 3 \frac{x^2 - 4x + 3}{-1} + 1 \frac{x^2 - 3x + 2}{2}$$

$$P_2(x) = -\frac{3}{2}x^2 + \frac{11}{2}x - 2.$$

Comprobación

$$P_2(1) = -\frac{3}{2} + \frac{11}{2} - \frac{4}{2} = \frac{4}{2} = 2, \quad \text{etc.}$$

Si se tienen muchos puntos (x_i, y_i) a intervalo variable y se desea generar una nueva secuencia a intervalo constante, es obvio que no se debe generar un polinomio de orden muy alto, sino que Lagrange se aplica localmente a pocos puntos, recorriendo toda la secuencia. No se recomienda llegar a polinomios de 5º orden o mayores, porque el proceso se hace inestable y provoca oscilaciones fuertes. (Fig 1)

En este tema, como en muchos otros, de Análisis Numérico, los métodos existentes son mucho, aquí podríamos citar como métodos locales a:

Newton con varios métodos.

Aitken.

Gauss

Stirling

Everett

Bessel

etc, etc.

En cuanto a los métodos globales se cae de lleno en todos los conjuntos de funciones ortogonales, ya que aquí primero se hace el ajuste de la función Explícitamente y después se valúan puntos particulares. Los conjuntos más comunes son los de

Fourier

Legendre

Laguerre

Hermite

Jacobi

Chebyshev, etc, etc.

SOLUCION DE ECUACIONES DIFERENCIALES

MÉTODO DIRECTO DE INTEGRACIÓN NUMÉRICA.

Sea una ecuación diferencial del tipo

$$\frac{d^2 y}{dx^2} = g(x)$$

o sea, la derivada de la función solo depende de la variable independiente, cuyas condiciones iniciales son:

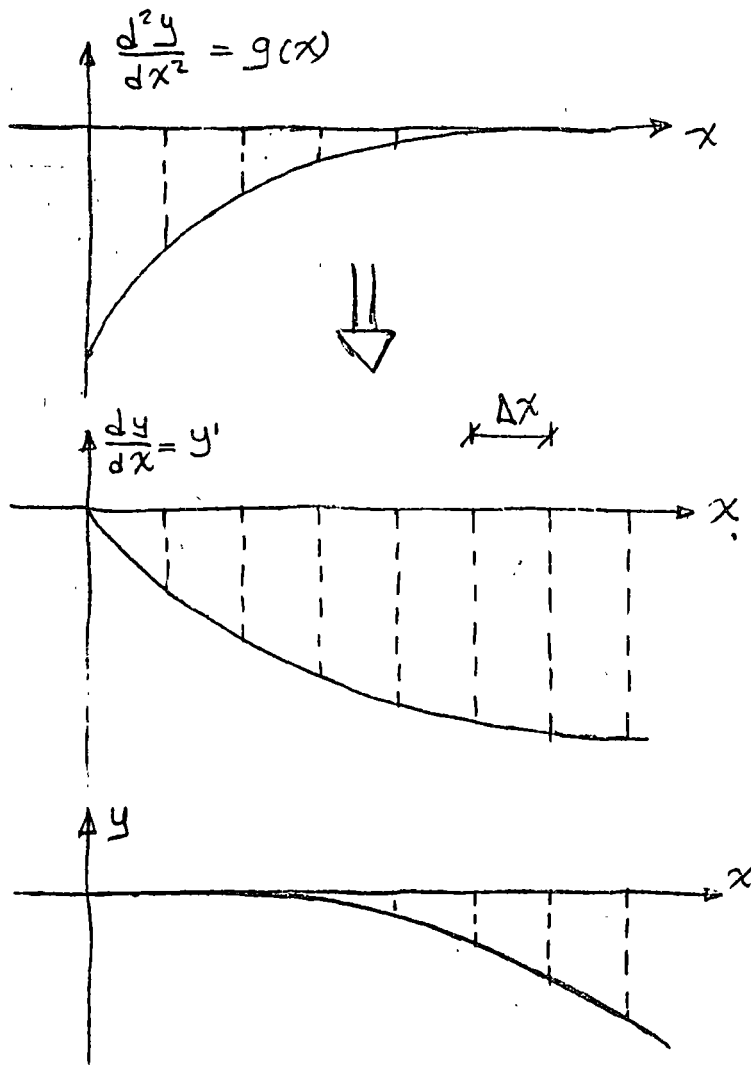
$$\text{para } x=0, \quad y=0, \quad y' = \frac{dy}{dx} = 0$$

La solución de esta ecuación implica definir los valores de y correspondientes a una serie de puntos x uniformemente espaciados. Esto se logra mediante dos aproximaciones sucesivas.

La curva de y' en función de x , se puede obtener ya que se conoce el valor de $y'(x=0)$ y la variación en y' de x_i a x_{i+1} esta representada por el área bajo la curva $g(x)$ entre los valores dados.

La curva de $y(x)$ se obtiene en forma similar debido a que conocemos $y(x=0)$ y el cambio en y a lo largo de un intervalo x esta dado por el área correspondiente bajo la curva $y'(x)$. Esta puede calcularse empleando Simpson, Trapecio o cualquier otra regla.

Sean las curvas.



Se sabe que

$$y(x=0) = 0$$

$$y'(x=0) = 0$$

$$y'(\Delta x) = \int_0^{\Delta x} y'' dx$$

$$= \frac{y''(x_0) + y''(x_1)}{2} \cdot \Delta x + y'(0)$$

$$y(x) = \int_0^{\Delta x} y' dx$$

$$= \frac{y'(x_0) + y'(x_1)}{2} \cdot \Delta x + y(0)$$

Se observa que la solución numérica de una ecuación diferencial, en lo que se conoce la curva de la derivada de mayor orden, se puede obtener por integración numérica directa siempre que se conozcan las condiciones iniciales, es decir, se requiere

$$y^n(x) = \frac{d^n y}{dx^n} = g(x) ; \quad \begin{array}{l} y(x=0) = y_0 \\ y'(x=0) = y'_0 \\ \vdots \\ y^{n-1}(x=0) = y_0^{n-1} \end{array}$$

METODO DE EULER.

Sea la ecuación

$$y^{(n)} = \frac{d^n y}{dx^n} = f(x, y, y', \dots, y^{(n-1)})$$

aquí ya no es posible efectuar la integración directamente. Ilustremos un nuevo método a través de una ecuación de primer orden

$$\frac{dy}{dx} = y' = f(x, y)$$

con la condición inicial $y(x=0) = y_0$.

Al conocer el valor inicial de y podemos determinar el valor inicial de y' a partir de la ecuación diferencial dada. El cambio en y al variar x de 0 a Δx se representa mediante el área bajo la curva y' entre los valores dados de x .

Sea la siguiente aproximación

$$y_1 - y_0 = y'_0 (\Delta x) = A_1$$

$$y_1 = y_0 + y'_0 (\Delta x)$$

Una vez definido un valor aproximado de y_1 podemos obtener y'_1 mediante la ecuación diferencial dada, o sea

$$y'_1 = f(x_1, y_1)$$

y considerando que $y_2 - y_1$ es aproximadamente igual al área bajo la curva y' (A_2), se tiene

$$y_2 - y_1 = y'_i (\Delta x) = A_2$$

$$y_2 = y_1 + y'_i (\Delta x)$$

con lo cual podemos calcular y'_2

$$y'_2 = f(x_2, y_2)$$

En general se tiene

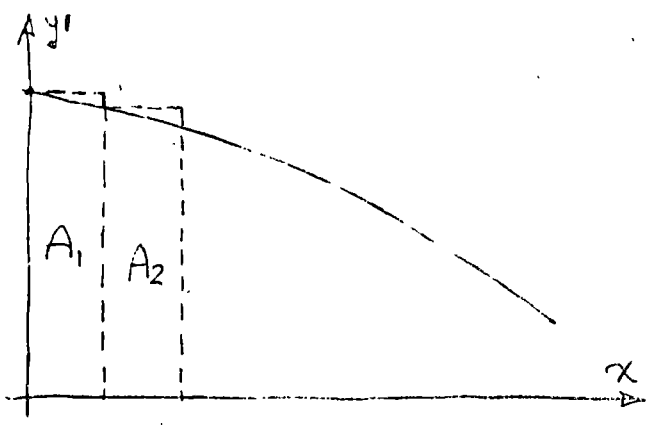
$$y_{i+1} = y_i + y'_i (\Delta x)$$

esta ecuación se conoce como fórmula de integración hacia adelante de Euler. A este criterio se le denomina "método que empieza por sí mismo", ya que solo requiere el valor de la variable de pendiente en un punto para iniciar el proceso.

En la ecuación general se tiene que el comportamiento de $y(x)$ se considera lineal, o sea, el método de Euler aproxima la función $y(x)$ por medio de una recta, en un intervalo Δx .

Dicha ecuación tiene la forma de un desarrollo en series de Taylor de y con respecto a x_i , con solo los dos primeros términos, por tanto el error de truncamiento que se introduce es del orden

$$E_T = c (\Delta x)^2$$



$$y_1 - y_0 = y'_0 (\Delta x) = A_1$$

$$y'_1 = f(x_1, y_1)$$

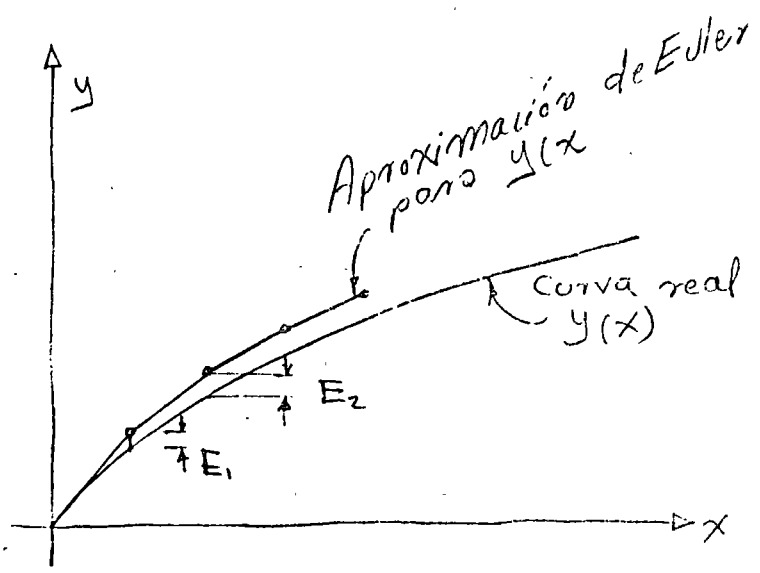
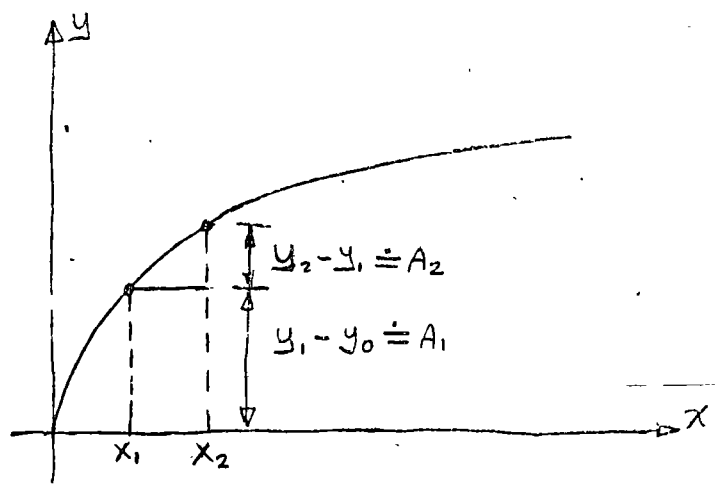
$$y_2 - y_1 = y'_1 (\Delta x) = A_2$$

$$y'_2 = f(x_2, y_2)$$

⋮

$$y_{i+1} = y_i + y'_i (\Delta x)$$

$$y'_i = f(x_i, y_i)$$



E_i - Error en el i -ésimo paso.

Recordando el método de Cálculo de Raíces de Newton-Raphson, Euler también "avanza por la tangente", solo que los errores son acumulativos.

Al igual que en otros procesos, el error que se comete es proporcional al paso Δx elevado en alguna potencia, de aquí la importancia de mantenerlo pequeño.

Ejemplo:

Sea una ecuación del tipo

$$\ddot{\theta} = f(\theta)$$

para este tipo de ecuaciones debe recordarse que una ecuación diferencial de orden n puede representarse como n ecuaciones diferenciales de primer orden.

Sean

$$\dot{\theta} = w \quad \text{----- (a)}$$

$$\ddot{\theta} = \dot{w} = f(\theta) \quad \text{----- (b)}$$

Aplicando el método descrito a la ecuación b, tenemos

$$w_{i+1} = w_i + \dot{w}_i (\Delta t) \quad \text{----- (c)}$$

y ahora en la ec. a.

$$\theta_{i+1} = \theta_i + w_i (\Delta t) \quad \text{----- (d)}$$

empleando las ecuaciones b, c y d se puede aplicar el método, solo resta indicar los valores iniciales. Sean

$$\theta_0 = 0$$

$$w_0 = 0$$

(7)

Definimos la secuencia del proceso numérico se tiene

$$\begin{aligned}\ddot{\theta}_i &= \dot{w}_i = f(\theta_i) & w_0 &= 0 \\ w_{i+1} &= w_i + \dot{w}_i (\Delta t) & \theta_0 &= 0 \\ \theta_{i+1} &= \theta_i + w_i (\Delta t)\end{aligned}$$

Sin embargo, mediante esta secuencia se obtiene que $\theta_i = 0$, lo cual sabemos es erróneo. Si Δt es pequeño, el error será pequeño. No obstante, se puede reducir este error y mejorar la aproximación de la solución, si aprovechamos la circunstancia de que conocemos w_i y w_{i+1} (en el primer paso serán w_0 y w_1). Usemos la regla trapezoidal para obtener θ_{i+1} en lugar de la regla rectangular indicada, esto es.

$$\theta_{i+1} = \theta_i + \left(\frac{w_i + w_{i+1}}{2} \right) \Delta t$$

por tanto: $\theta_1 = \frac{w_1 (\Delta t)}{2}$

que es una mejor aproximación al valor de θ_1 en el final del primer intervalo:

¿Podemos hacer lo mismo con w_{i+1} ?

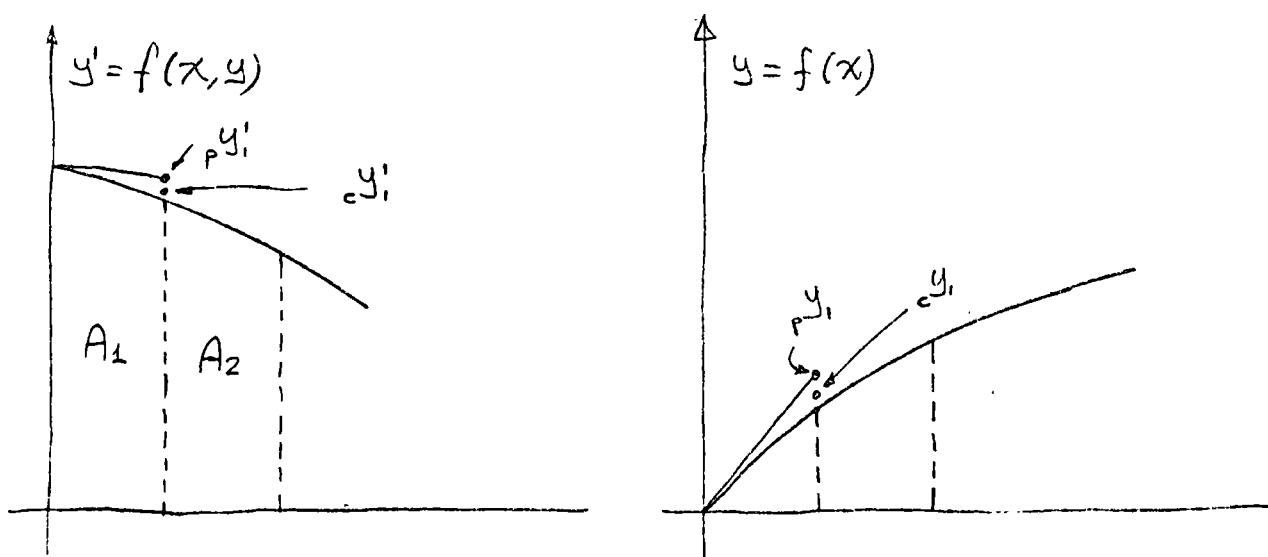
MÉTODOS MODIFICADOS DE EULER.

Método de Quinn o método mejorado de Euler.
Este método es del tipo predictor corrector que empieza por sí mismo.

Sea una ec. diferencial de primer orden

$$\frac{dy}{dx} = f(y, x)$$

con la condición inicial $y(x=0) = y_0$. Supongamos que las curvas y' y y son del tipo mostradas en la figura.



Evaluando la ecuación diferencial con el valor inicial definimos $y'(x=0)$, y de aquí podemos predecir un valor y_1 de la función: $p y_1 = y_0 + y'_0(\Delta x)$.

Debe recordarse que estamos empleando una integración rectangular.

Con este valor y_1 podemos encontrar uno de y'_1 , también predicho, $p y'_1$.

Con estos valores provisionales se puede encontrar un valor corregido de y_1, y_1 , empleando integración trapezoidal

$$c y_1 = y_0 + \frac{y'_0 + p y'_1}{2} \Delta x$$

esta última igualdad se conoce como "ecuación de corrección". Este último valor es el que se emplea para calcular el valor corregido de $y'_1, c y'_1$.

$$c y'_1 = f(x, c y_1)$$

Ahora tenemos un valor más aproximado de y' en $x = \Delta x$, por tanto podemos calcular otro valor de y_1 que será mejor, y después otro de y'_1 y así sucesivamente hasta que las variaciones en la función solución y , sean menores que una tolerancia fijada de antemano.

De aquí podemos dar el siguiente incremento en x , volver a corregir, etc.

Fórmula de Predicción

$$p y_{i+1} = y_i + y'_i (\Delta x)$$

Fórmula de Corrección

$$c y_{i+1} = y_i + \frac{y'_i + p y'_{i+1}}{2} \Delta x$$

con esto se llega a un error del orden $(\Delta x)^2$.

RUNGE-KUTTA.

Un método de Runge-Kutta es aquel que emplea una fórmula de recurrencia del tipo

$$y_{i+1} = y_i + a_1 k_1 + a_2 k_2 + a_3 k_3 + \dots + a_n k_n \quad \text{--- (1)}$$

para calcular valores sucesivos de la variable dependiente y de la ecuación diferencial,

$$\frac{dy}{dx} = y' = f(x, y)$$

donde:

$$k_1 = (\Delta x) f(x_i, y_i)$$

$$k_2 = (\Delta x) f(x_i + p_1 \Delta x, y_i + q_{11} k_1)$$

$$k_3 = (\Delta x) f(x_i + p_2 \Delta x, y_i + q_{21} k_1 + q_{22} k_2)$$

⋮

$$k_m = (\Delta x) f(x_i + p_{m-1} \Delta x, y_i + q_{m-1,1} k_1 + q_{m-1,2} k_2 + \dots + q_{m-1,m-1} k_{m-1})$$

donde a, p, q deben adoptar valores tales que la ecuación (1) suministre con precisión valores sucesivos de y .

Estos valores se determinan al hacer una equivalencia a una serie truncada de Taylor de y con respecto a x_i . Para ello se requieren los conceptos de diferencial total y serie de Taylor para funciones de dos variables.

- Si se tiene la ecuación $y' = f(x, y)$ la diferencial total dy' es

$$dy' = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy$$

$$\therefore \frac{dy'}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \frac{dy}{dx}$$

- Sea $z = f(x, y)$, entonces la serie de Taylor al rededor de x_i, y_i será:

$$z(x_i+h, y_i+j) = f(x_i, y_i) + h \left[\frac{\partial f}{\partial x}(x_i, y_i) \right] + j \left[\frac{\partial f}{\partial y}(x_i, y_i) \right] + \frac{1}{2!} \left\{ h^2 \left[\frac{\partial^2 f}{\partial x^2}(x_i, y_i) \right] + 2hj \left[\frac{\partial^2 f}{\partial x \partial y}(x_i, y_i) \right] + j^2 \left[\frac{\partial^2 f}{\partial y^2}(x_i, y_i) \right] \right\} + \dots$$

siendo h y j los incrementos de x y y respectivamente.

Desarrollando a grandes pasos el método de Runge-Kutta de 2º orden se tiene:

$$y_{i+1} = y_i + a_1 k_1 + a_2 k_2 \tag{2}$$

$$k_1 = (\Delta x) f(x_i, y_i)$$

$$k_2 = (\Delta x) f(x_i + p_1 \Delta x, y_i + q_{11} k_1)$$

donde se deben determinar a_1, a_2, p_1 y q_{11} de las series de Taylor se tiene:

$$y_{i+1} = y_i + (\Delta x) y'_i + \frac{(\Delta x)^2}{2!} y''_i + \dots$$

y empleando la ecuación diferencial original

$$y'_i = f(x_i, y_i)$$

y con la diferencial total.

$$y_i'' = \frac{\partial f}{\partial x}(x_i, y_i) + \left[\frac{\partial f}{\partial y}(x_i, y_i) \right] \left[f(x_i, y_i) \right]$$

se tiene.

$$y_{i+1} = y_i + (\Delta x) f(x_i, y_i) + \frac{\Delta x^2}{2!} \left\{ \frac{\partial f}{\partial x}(x_i, y_i) + \left[\frac{\partial f}{\partial y}(x_i, y_i) \right] \left[f(x_i, y_i) \right] \right\} + \dots$$

de lo ec 2 y esta última se encuentra que k_2 sera una función de $f, \frac{df}{dx}$ y $\frac{df}{dy}$ valuados en x_i, y_i . Por lo que k_2 se desarrolla en serie de Taylor para funciones de dos variables, respecto a x_i, y_i .

Despues de desarrollar e igualarlas se llega

a:

$$a_1 + a_2 = 1$$

$$a_2 P_1 = 1/2$$

$$a_2 q_{11} = 1/2.$$

Asignando arbitrariamente un valor a una de las incógnitas, se pueden encontrar los otros 3.

Si en particular hacemos $a_1 = 1/2$

se tiene $a_2 = 1/2$

$$P_1 = 1$$

$$q_{11} = 1$$

$$\therefore y_{i+1} = y_i + \frac{1}{2} (k_1 + k_2)$$

con $k_1 = (\Delta x) f(x_i, y_i)$

$$k_2 = (\Delta x) f(x_i + \Delta x, y_i + k_1)$$

Este conjunto en particular da valores similares a Euler Modificado.

METODOS NUMERICOS Y APLICACIONES CON LA COMPUTADORA DIGITAL
(DEL 7 AL 29 DE ABRIL DE 1978)

<u>NOMBRE Y DIRECCION</u>	<u>EMPRESA Y DIRECCION</u>
1. ALFONSO GUTBERTO ARAGON CARO Vicente Guerrero No. 338 Nte. Culiacán, Sin.	UNIVERSIDAD AUTONOMA DE SINALOA ESCUELA DE INGENIERIA Constitución y Andrade Culiacán, Sin.
2. JESUS A. BAROCIO JUAREZ Av. La Hacienda And. 8 Casa 6-1 Villa Coapa México 22, D. F. Tel: 5-94-74-74	COMISION DE AGUAS DEL VALLE DE MEXICO Balderas No. 55-2o. Piso México 1, D. F. Tel: 5-85-50-66 Ext. 600
3. LETICIA O. BORJA ABURTO Hda. Postejé No. 24 Floresta Coyoacán México 22, D. F.	SECRETARIA DE HACIENDA Y CREDI- TO PUBLICO Palacio Nacional Edificio 6-2o.p México 1, D. F. Tel: 5-85-40-11 Ext. 3011
4. ING. HECTOR L. FONSECA LOPEZ Planta el Novillo No. 6 Col. Electra Tlalnepantla, Edo. de México Tel: 3-97-42-81	COMISION FEDERAL DE ELECTRICIDAD Melchor Ocampo No. 463-3er. Piso Col. Cuauhtémoc México 5, D. F. Tel: 5-53-71-33 Ext. 2420
5. ING. RODOLFO GOMEZ VALLE Hacienda de la Purisima No. 64 Atzacapotzalco México 6, D. F.	COMISION FEDERAL DE ELECTRICI- DAD Melchor Ocampo No. 455-7o. Piso Col. Polanco México 5, D. F. Tel: 5-53-71-33 Ext. 2149
6. ING. ALEJANDRO J. GONZALEZ SILVA Ave. Rafael Cravioto No. 55 Huauchinango, Puebla Tel: 2-06-72	PETROLEOS MEXICANOS Sistema de Oleoductos y Gasoductos del Depto. Campo Catalina Huauchinango, Pue. Tel: 2-00-98
7. JOSE LUIS GUILLEN UTRILLA U. Lomas de Plateros Edificio G-6 Depto. 43 México 19, D. F. Tel: 6-51-03-59	COMISION DE AGUAS DEL VALLE DE MEXICO Balderas 55-2o. Piso México 1, D. F. Tel: 5-85-50-66

METODOS NUMERICOS Y APLICACIONES CON LA COMPUTADORA DIGITAL
(DEL 7 AL 29 DE ABRIL DE 1978)

<u>NOMBRE Y DIRECCION</u>	<u>EMPRESA Y DIRECCION</u>
8. J. DE JESUS HERNANDEZ TORRES Venecia 64-6 San Alvaro México 17, D. F. Tel: 3-99-23-59	BUFETE INDUSTRIAL DISEÑOS Y PROYECTOS, S. A. Tolstoi No. 22 Col. Anzures México 5, D. F. Tel: 5-33-15-00
9. ENRIQUE P. HIMMELSTINE CASTRO Mar de la Crisis No. 12 Los Olivos México 22, D. F. Tel: 5-44-37-11	SECRETARIA DE ASENTAMIENTOS HUMANOS Y OBRAS PUBLICAS Av. Fernando 268-8o. Piso Col. Alamos México 13, D. F. Tel: 5-90-89-74
10. FAUSTO A. JUAREZ RAMIREZ Vicente Beristain No. 10 Int. 9 Vista Alegre México 8, D. F.	SECRETARIA DE ASENTAMIENTOS HUMANOS Y OBRAS PUBLICAS Av. Fernando No. 268-8o. Piso Col. Alamos México 13, D. F. Tel: 5-90-89-74
11. AUDOMARO J. R.R. LASTRA L. Ingres 134-407 Col. Mixcoac México 19, D. F. Tel: 5-98-06-10	SECRETARIA DE HACIENDA Y CREDITO PUBLICO Palacio Nacional Edif. 6-2o. Piso México 1, D. F. Tel: 5-85-40-11
12. GUADALUPE H. MARIN PACHECO Unidad 1 No. 126 Col. Jardín Balbuena México 9, D. F.	COMISION DE AGUAS DEL VALLE DE MEXICO Balderas No. 55-3er. Piso México 1, D. F. Tel: 5-85-50-66 Ext. 309
13. ING. GERMAN MENDEZ MILLARES Retorno 23 No. 35 Col. Avante México 21, D. F. Tel: 5-75-99-97	I.P.N. ESIME CULHUACAN Av. Santa Ana No. 1000 San Francisco Culhuacan México 13, D. F. Tel: 5-81-87-23 Ext. 21
14. ING. JOSE DE JESUS MORALES MARTINEZ Privada de Tezozomoc No. 19 Manzana "A" Las Trancas México 16, D. F. Tel: 5-61-96-46	COMISION DE AGUAS DEL VALLE DE MEXICO Balderas No. 55-2o. Piso México 1, D. F. Tel: 5-85-50-66 Ext. 600

METODOS NUMERICOS Y APLICACIONES CON LA COMPUTADORA DIGITAL
(DEL 7 AL 29 DE ABRIL DE 1978)

NOMBRE Y DIRECCION

EMPRESA Y DIRECCION

15. JUAN RAMON ROJO QUINTERO
Sepulveda No. 64 Nte.
Culiacán, Sin.

UNIVERSIDAD AUTONOMA DE
SINALOA (ESCUELA DE INGENIERIA
CIVIL)
Constitución y Andrade
Culiacán, Sin.

16. MA. DE LOS ANGELES TOVAR ZAMBRANO
Cda. de Tapicería No. 6
México 2, D. F.
Tel: 7-890541

