DISEÑO OPTIMO DE SISTEMAS DE INGENIERIA
DIRECTORIO DE PROFESORES 1982

1. DR. ANTONIO MONTALVO ROBLES
   JEFE DE PROCESOS GEOFISICOS
   GERENCIA DE EXPLORACION
   PETROLEOS MEXICANOS
   MARINA NACIONAL 329 6ª piso
   MEXICO 17, D.F.
   TEL: 531. 63. 08

2. DR. JORGE ANGELES ALVAREZ (COORDINADOR)
   SUBJEFE DEL AREA DE INGENIERIA MECANICA
   DIVISION DE ESTUDIOS DE POSGRADO
   FACULTAD DE INGENIERIA, UNAM
   CIUDAD UNIVERSITARIA
   MEXICO 20, D.F.
   TEL: 550.52.15 ext. 4485

3. DR. ENRIQUE CHICUREL UZIEL
   INVESTIGADOR
   INSTITUTO DE INGENIERIA, UNAM
   CIUDAD UNIVERSITARIA
   MEXICO 20, D.F.
   TEL: 550. 52.15 ext. 4470

4. DR. JOSE MIGUEL COBIAN SELA
   PROFESOR TITULAR DEL AREA DE SISTEMAS
   DIVISION DE ESTUDIOS DE POSGRADO
   FACULTAD DE INGENIERIA, UNAM
   CIUDAD UNIVERSITARIA
   MEXICO 20, D.F.
   TEL: 550.52.15 ext. 4477

5. DRA. SUSANA GOMEZ GOMEZ
   INVESTIGADORA
   INSTITUTO DE INVESTIGACIONES EN MATEMATICAS
   APLICADAS Y EN SISTEMAS, UNAM
   CIUDAD UNIVERSITARIA
   MEXICO 20, D.F.
   TEL: 550.52.15 ext. 4572

6. DR. HORACIO MARTINEZ CARRANZA
   INVESTIGADOR DEL DEPTO. DE SIMULACION
   INSTITUTO DE INVESTIGACIONES ELECTRICAS
   INTERIOR INTERNADO PALMIRA
   CUERNAVACA, MOR.
   TEL: 91. 731.4.38.11 ext. 2132 ó 2121

DISEÑO OPTIMO DE SISTEMAS DE INGENIERIA 1982

Fecha	Tema	Horario	Profesor
Marzo 29	INTRODUCCION	9 a 17 h	Dr. Jorge Angeles Alvarez
	Antecedentes matemáticos y numéricos de las técnicas de optimización		
Marzo 30	CONDICIONES DE OPTIMALIDAD	9 a 11 a.m.	Dr. Antonio Montalvo Robles
	Métodos de Optimización sin Restricciones	11 a 17 h	" " " "
Marzo 31	" " " " "	9 a 12:30 h	" " " "
	METODOS DE OPTIMIZACION CON RESTRICCIONES	12:30 a 17 h	Dra. Susana Gómez Gómez
Abril 1°	" " " " "	9 a 17 h	" " " "
Abril 2	OPTIMACION DE ELEMENTOS DE MAQUINAS	9 a 11 a. m.	Dr. Enrique Chicurel Uziel
	DISEÑO OPTIMO DE FILTROS DIGITALES PARA EL PROCESAMIENTO DE OPTIMACION	11 a 13: 30 h	Dr. Horacio Martínez Carranza
	OPTIMACION DE TRENES DE INTERCAMBIADORES DE CALOR	15 a 16 h	Dr. Antonio Montalvo Robles
	PLANEACION DE REDES ELECTRICAS POR PROGRAMACION MATEMATICA	16 a 17 h	Dr. José Miguel Cobián Sela

edcs.

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO OPTIMO DE SISTEMAS DE INGENIERIA

METODOS DE OPTIMACION SIN RESTRICCIONES

DR. ANTONIO MONTALVO ROBLES

MARZO, 1982

METODOLOGIA

I. BUSQUEDAS UNIDIRECCIONALES

Casi la totalidad de las técnicas de minimización que se describirán en otras secciones más adelante, requieren de técnicas de minimización unidimensionales las cuales tienen como propósito el localizar el mínimo local de una función de una variable. La implementación de los métodos mencionados requieren del conocimiento previo de un cierto intervalo cual contiene al mínimo de la función objetivo $f(x)$. y además se supone que en el intervalo prescrito la función es unimodal. En la Tabla II se mencionan algunos de los métodos más conocidos para lograr la minimización deseada, todos los cuales tienen como próposito reducir el tamaño del intervalo $\Delta^{(o)}$ hasta un tamaño $\Delta^{(n)}$. Para comparar la rapidez relativa de los métodos, Wilde define una eficiencia para n evaluaciones de la función como :

$$\text{Eficiencia} = L = \frac{\Delta^{(n)}}{\Delta(o)}$$

En la tabla I se comparan los valores de E para varios métodos.

Métodos no-secuenciales	ε	Secuenciales	ε
Búsqueda uniforme	$\frac{2}{n+1}$	Búsqueda secuencial (Dicotomus)	$\frac{1}{2^{n/2}}$
Búsqueda uniforme (Dicotomus)	$\frac{1}{\frac{n}{2}+1}$	Búsqueda Fibonacci	$\frac{1}{F_n^{*}}$
		Sección Dorada	$(0.618)^{1-n}$

(*): Número de Fibonacci para n evaluaciones

TABLA II. Eficiencia de Técnicas de Búsqueda Unidimensional

En la Tabla III se compara el número de evaluaciones de la función que se requiere para reducir un intervalo inicial de 5×10^{1} a uno menor.

	NO SECUENCIAL		SECUENCIAL		
$\Delta^{(o)}$	Uniforme	Dicotomus	Dicotomus	Fibonacci	Sección Dorada
$5 \cdot 10^{-3}$	199	198	14	11	11
$5 \cdot 10^{-5}$	19,999	19.998	28	21	21

TABLA III. Número de evaluación de la función para reducir $\Delta^{(o)} = 5 \times 10^{-1}$

A continuación se describe el método de la sección dorada. Este método esta basado en la división de una linea en dos segmentos tales que la relación del tamaño original de la linea al segmento mayor es la misma

que la relación del segmento mayor al menor, es decir :

$$F_1 + F_2 = 1$$

$$\frac{1}{F_2} = \frac{F_2}{F_1} \quad ; \quad F_2^2 = F_1$$

de donde

$$F_1 = \frac{3 - \sqrt{5}}{2} = 0.38$$

$$F_2 = \frac{\sqrt{5} - 1}{2} = 0.62$$

Para iniciar la búsqueda del mínimo de $f(x)$, se necesita especificar (o averiguar) en que dirección ésta se llevará a cabo. (Se supondrá que se conoce).

Como primer paso se debe encontrar un intervalo Δ donde se encuentra el mínimo de $f(x)$ usando, por ejemplo, una serie de pasos cada vez más grande sobre la variable independiente. Suponga que esto se ha hecho y que los últimos tres puntos obtenidos en x son los siguientes : $x_3^{(o)}$, el último, $x_2^{(o)}$ y $x_1^{(o)}$, donde $f(x_3^{(o)}) \geq f(x_2^{(o)})$ y sea $\Delta^{(k)} = x_3^{(k)} - x_1^{(k)}$ (Fig. 6). Lo anterior una vez hecho la k-ésima etapa modifica el intervalo de la siguiente forma.

$$y_1^{(k)} = x_1^{(k)} + F_1 \Delta^{(k)}$$

$$y_2^{(k)} = x_1^{(k)} + F_2 \Delta^{(k)} = x_3^{(k)} - F_1 \Delta^{(k)}$$

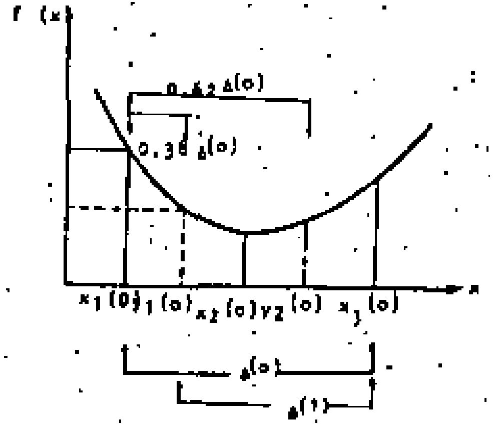
si $f(y_1^{(k)}) < f(y_2^{(k)})$: $\Delta^{(k+1)} = (y_2^{(k)} - x_1^{(k)})$, y $x_1^{(k+1)} = x_1^{(k)}$, $x_3^{(k+1)} = y_3^{(}$

si $f(y_1^{(k)}) > f(y_2^{(k)})$: $\Delta^{(k+1)} = (x_3^{(k)} - y_1^{(k)})$, y $x_1^{(k+1)} = y_1^{(k)}$, $x_3^{(k+1)} = x_3^{(}$

si $f(y_1^{(k)}) = f(y_2^{(k)})$: $\Delta^{(k+1)} = (y_2^{(k)} - x_1^{(k)}) = (x_3^{(k)} - y_1^{(k)})$, y

$$x_1^{(k+1)} = x_1^{(k)}, \ (x_3^{(k+1)} = y_2^{(k)}, \quad o$$

$$x_1^{(k+1)} = y_1^{(k)}, \quad x_3^{(k+1)} = x_3^{(k)}$$



$$y_1^{(o)} = x_1^{(o)} + 0.38 \ \Delta^{(o)}$$

$$y_2^{(o)} = x_1^{(o)} + 3.62 \ \Delta^{(o)}$$

FIGURA 6. Búsqueda mediante sección Dorada.

Otra clase de métodos para búsquedas unidireccionales. Localizar un punto x, cercano a x^* (el mínimo) mediante interpolación y extrapolación. En estos métodos se usan interpolaciones cuadráticas y cúbicas para aproximar el valor de la función. A continuación se describen un par de algoritmos que al usarlos en forma conjunta generan un algoritmo bastante poderoso para la localización de mínimos locales en una dirección: estos dos algoritmos son los siguientes

a) Davis-Swann - Campey (DSC) para definir el intervalo Δ donde se encuentra el mínimo, y

b) Powell, para definir la localización "exacta" del mínimo.

En el método DSC, se toman pasos de tamaño cada vez mayor hasta que se sobrepasa la localización del mínimo. A partir de ese momento se usa interpolación cuadrática (Figura 7). Los pasos que se siguen en este algoritmo son los siguientes

1. Evaluar $f(x)$ en el punto inicial $x^{(o)}$. Si $f(x^{(o)} + \Delta k$
   Si $f(x^{(o)} + \Delta x) \leq f(x^o)$ se continua con el paso 2.
   Si $f(x^{(o)} + \Delta x) > f(x^o)$ se define $\Delta x = -\Delta x$ (se cambia la dirección de búsqueda) y se continua con el paso 3.

2. $x^{(k+1)} = x^{(k)} + \Delta x$

3. Calcular $f(x^{(k+1)})$

4. Si $f(x^{(k+1)}) \leq f(x^{(k)})$, se duplica el tamaño de paso Δx y se repite el procedimiento desde el paso 2. Si $f(x^{(k+1)}) > f(x^{(k)})$ sea $x^{(m)} = x^{(k+1)}$, $x^{(m-1)} = x^{(k)}$, etc., reduzcase x a la mitad y regrese a los pasos 2 y 3 una vez más.

5. De los cuatro puntos igualmente espaciados x, en el conjunto $\{x^{(m+1)}, x^{(m)}, x^{(m-1)}, x^{(m-2)}\}$, se eliminan o $x^{(m)}$ o $x^{(m-2)}$, el que este más lejos de la x con el valor de la función más baja. Los tres puntos restantes se denotan como $x^{(a)}$, $x^{(b)}$ y $x^{(c)}$, donde $x^{(b)}$ es el punto central y $x^{(a)} = x^{(b)} - x$ y $x^{(c)} = x^{(b)} + \Delta x$.

6. Use interpolación cuadrática para estimar x^*.

$$x^* = \bar{x}^* = x^{(b)} + \frac{\Delta x\left(f(x^{(a)}) - f(x^{(c)})\right)}{2\left[f(x^{(a)}) - 2f(x^{(b)}) + f(x^{(c)})\right]}$$

Los pasos anteriores completan la primera etapa del método DSC. Para continuar, se reinicia en $\bar{x}^*$ o $x^{(c)}$, si $f(x^{(c)}) < f(\bar{x}^*)$, se reduce Δx y se empieza desde el paso 1 nuevamente.

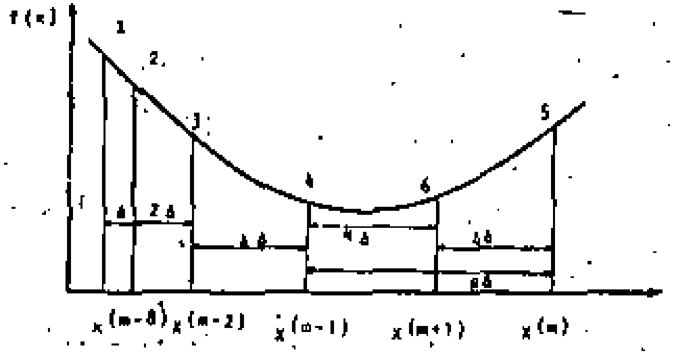


FIGURA 7. Método DSC para minimización unidimensional.

En el método de Powell, se usa una aproximación cuadrática usando los tres primeros puntos obtenidos en la dirección de búsqueda dada. La x correspondiente al mínimo de la función cuadrática se usa para efectuar una nueva aproximación y se continua de esta forma hasta localizar el mínimo de $f(x)$ (Figura 8).

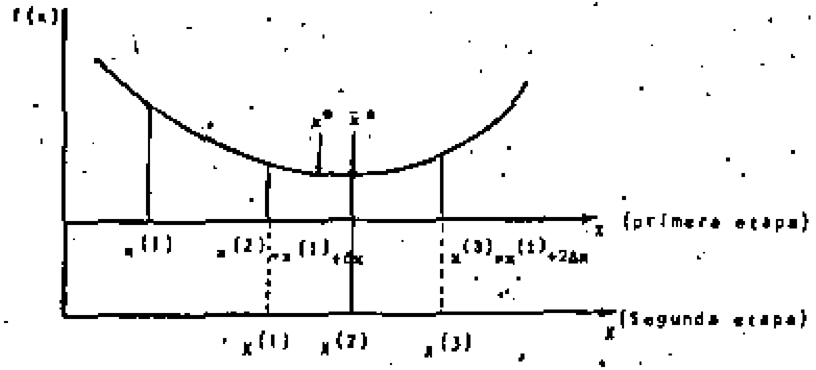


FIGURA 8. Método de Powell para minimización unidimensional.

Los pasos que deben seguirse para implementar el método de Powell son:

1. dados $x^{(1)}$ y Δ, calcule $x^{(2)} = x^{(1)} + \Delta$

2. calcule $f(x^{(1)})$ y $f(x^{(2)})$

3. Si $f(x^{(1)}) > f(x^{(2)})$, $x^{(3)} = x^{(1)} + 2\Delta$

   Si $f(x^{(1)} \leq f(x^{(2)})$, $x^{(3)} = x^{(1)} - \Delta$

4. Calcule $f(x^{(3)})$

5. Estime el valor de x en el mínimo de $f(x)$, $\bar{x}^{*}$, como

$$x^{*} = \bar{x}^{*} = \frac{\left[(x^{(2)})^{2} - (x^{(3)})^{2}\right] f(x^{(1)}) + \left[(x^{(\ )})^{2} - (x^{(1)})^{2}\right] f(x^{(2)}) + \left[(x^{(1)})^{2} - (x^{(2)})^{2}\right] f(x^{(3)})}{(x^{(2)} - x^{(3)})\, f(x^{(1)}) + (x^{(3)} - x^{(1)})\ f(x^{(2)}) + (x^{(1)} - x^{(2)})\ f(x^{(3)})}$$

6. Si $\bar{x}^{*}$ o algunos de entre $\{x^{(1)}, x^{(2)}, x^{(3)}\}$ que corresponda al valor más pequeño de $f(x)$, difiere en menos que la exactitud presenta para x, o la exactitud en el valor de $f(x)$, se termina la búsqueda. En caso contrario, evalúe $f(\bar{x}^{*})$ y se elimina del conjunto $\{x^{(1)}, x^{(2)}, x^{(3)}\}$ del que tenga el valor más alto de la función $f(x)$, a menos que se pierda el intervalo de x donde se encuentra el mínimo, en cuyo caso/se debe eliminar una x tal que el intervalo adecuado no se pierda. Se repita el procedimiento desde el paso 6.

II. MINIMIZACION SIN RESTRICCIONES USANDO DERIVADAS

El problema general de minimización sin restricciones se puede plantear como :

$$\text{Minimizar} : \quad f(x) , \quad x \in E^{n} \qquad (1)$$

donde $f(x)$ es la función objetivo. Según se vió en el capítulo anterior, se pretende encontrar un punto x^{*} tal que $\nabla f(x^{*}) = 0$. En la presente sección se atacará el problema definido en (1) mediante el uso de métodos que hagan uso de las primeras y segundas derivadas parciales $(\nabla f(x)$ y $\nabla^{2} f(x))$ de la función objetivo.

2.1. Método del Máximo Descenso (Gradiente)

Según se recordará del capítulo anterior, el gradiente de la función objetivo $f(x)$, en cualquier punto x, es un vector dirigido en la dirección del máximo incremento local en $f(x)$. Resulta obvio que se puede escoger como dirección de búsqueda, para minimizar $f(x)$, la dirección opuesta al gradiente, $-\nabla f(x)$, esto es, en la dirección de máximo descenso. Si se escoge como dirección de búsqueda la ya señalada, resultará lo siguiente.

a) sea $\nabla s_r = - f(x_k)$ la dirección de búsqueda en el k-ésimo paso del algoritmo.

b) Si $\Delta x_k = + \alpha_k S_k = - \alpha_k \nabla f(x_k)$, es el desplazamiento del punto x_k al x_{k+1}, es decir

$$x_{k+1} = x_k + \Delta x_k \qquad (\alpha_k \text{ un escalar positivo})$$

c) Entonces, la aproximación a primer orden de $f(x)$ quedará como

$$f(x_k + \Delta x_k) = f(x_k) + \nabla^{T} f(x_k) \Delta x_k$$

o bien

$$f(x_k + \Delta x_k) - f(x_k) = - \alpha_k \nabla^{T} f(x_k) \nabla f(x_k) < 0$$

es decir, se puede garantizar que para α_k suficientemente pequeño, el valor de la función en x_{k+1} decrecerá con respecto al valor previo en x_k, siempre y cuando $\nabla f(x_k) \neq 0$. (En los puntos (a) - (c), α_k se le conoce como tamaño de paso).

Existen varias alternativas para escoger el tamaño de paso α_r, de las cuales se pueden mencionar las siguientes

a) Fijar el valor de α_k de antemano e igual para todas las iteraciones del método. La desventaja de proceder de esta forma es que no se puede garantizar que de una iteración a

la siguiente, el valor de la función decrezca, ya que esta propiedad sólo es válida cuando $\alpha_k \longrightarrow 0$.

Debido a lo anterior, el método puede presentar las siguientes desventajas. En primer lugar, que si α_k no es lo "suficientemente pequeña" el método ascilará. Por otro lado, si α_k se escoge "demasiado pequeña" la convergencia puede volverse demasiado pequeña.

b) Una segunda alternativa es escoger α_k en cada iteración de forma tal que el valor de la función objetivo se reduzca para algún cierto valor de α_k. Para lograr lo anterior se puede proceder de la siguiente forma en cada etapa

```
a)    se escoge  $\alpha ref > 0$ ,  $0 < \gamma < 1$  un factor multiplicativo
i)    se fija  $\beta_0 = \alpha ref$, $i = 0$
ii)   $y^{(i)} = x_k + \beta_0 \delta_k$
iii)  calcular  $f(y^i)$
iv)   si  $f(y^i) < f(x_k)$  continuar con el paso vii)
v)    $i \longleftarrow i + 1$
vi)   $\beta_{i+1} = \gamma\beta_i$  ; repetir el procedimiento desde (ii)
vii)  $x_{k+1} = y^{(i)}$ ;  $\alpha_k = \beta_i$
      $f(x_{k+1}) = f(y^i)$
```

El procedimiento anterior garantiza que, si se permite un número ilimitado de iteraciones $\beta_{i+1} = \gamma\beta_i$, en cada etapa del método de máximo descenso, se obtendrá un descenso en la función objetivo. Tiene el defecto a que puede consumir demasiado tiempo en la búsqueda de un tamaño de paso adecuado α_k.

c) Una tercera alternativa es la siguiente. Supongase que de alguna forma la dirección de búsqueda δ_k, es decir, el nuevo iterando x_{k+1} caerá sobre la ecuación de un parámetro

$$x_{k+1} = x_k + \alpha\delta_k$$

siendo α el parámetro. La diferencia entre este procedimiento y los dos anteriores es que en el presente se pide que el tamaño de paso α sea tal que $f(x_{k+1}) = f(x_k + \alpha\delta_k)$ adquiera su mínimo valor 0, formalmente

$$\frac{d f(x_k + \alpha\delta_k)}{d\alpha} = 0$$

Por ejemplo, si $f(x)$ es una función cuadrática

$$f(x) = a + b^T x + \frac{1}{2} x^T Q x \quad \text{(Q : positiva definida, simétrica)}$$

entonces

$$\delta_k = - \nabla f(x_k) = - b - Qx_k$$

$$x_{k+1} = -\alpha_k (b + Q x_k)$$

$$df(x_k-\alpha_k(b+Qx_k)) = 0 = \nabla^T f(x_k)\, s_k + s_k^T Q (\alpha_k s_k)$$

de donde

$$\alpha_k = -\frac{\nabla^T f(x_k) s_k}{}$$

Si se supone que $f(x)$ no es una función cuadrática de x, entonces se pueden emplear cualquiera de las técnicas de búsquedas unidimensionales descritas en la sección anterior.

Una característica interesante de este procedimiento de minimización es que el gradiente en el nuevo punto, $\nabla f(x_{k+1})$ es ortogonal a la dirección de búsqueda empleada para localizar x_{k+1}, es decir $\nabla^T f(x_{k+1})\, s_k = 0$, lo cual se demuestra como sigue. Supóngase la misma función cuadrática ya empleada entonces

$$\nabla f(x_k) = b + Q x_k .$$

y de la expresión para $\dfrac{d f(\alpha)}{d\alpha} = 0$ se obtiene

$$(b + Q x_k)^T s_k + s_k^T Q \alpha_k s_k = 0$$

y como $0x_{k+1} - x_k = {}_k s_k$, entonces

$$s_k^T (b + Q x_k) + s_k^T Q (x_{k+1} - x_k) = 0.$$

o bien

$$s_k^T (b + Q x_{k+1}) = s_k^T \nabla f(x_{k+1}) = 0.$$

Lo anterior se ilustra en la Figura 9.

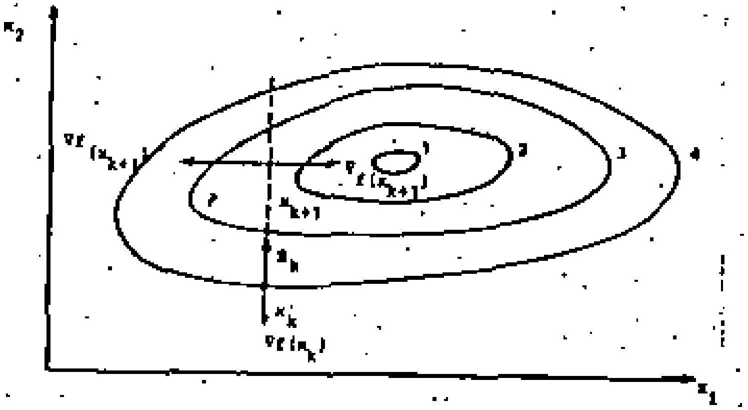


FIGURA 9. Ortogonalidad de las direcciones de Búsqueda

La implementación práctica del algoritmo de máximo descenso, con cualquiera de las tres alternativas discutidas para determinar α_k, quedaría de la siguiente forma.

1. $k = 0$

2. Estimar x_o (punto de arranque).

3. Calcular $f(x_k)$, $f(x_k)$

4. Si $\nabla^T f(x_k) \ \nabla f(x_k) \leq$ tolerancia, se ha encontrado, un punto estacionario de la función $f(x)$

5. $S_k = - \nabla f(x_k)$

6. Cálculo de α_k (ver texto)

   como resultado se calcula x_{k+1} y $f(x_{k+1})$

7. Calcular $\nabla f(x_{k+1})$.

8. $k \longleftarrow k+1$; se repite el procedimiento desde el paso 4.

Para finalizar la discusión sobre el método del máximo descenso es conveniente hacer notar que bajo algunas condiciones, por cierto no muy frecuentes, el algoritmo puede ser atraido por un punto silla ya que también en esta clase de puntos se satisface que $\nabla^T f(x) \ \nabla f(x) = 0$, no existiendo forma de detectar a priori que tal cosa sucederá. Ahora bien, para clasificar el tipo de punto donde se detuvo el algoritmo, es necesario analizar la matriz de segundas derivadas, de acuerdo a :

i) H, positiva definida — mínimo

ii) H, negativa definida — máximo

iii) H, semi-positiva o semi-negativa definida — punto silla.

2.2. Método de Newton

El método de Newton que a continuación se presenta esta basado en una aproximación a segundo orden de la función objetivo $f(x)$, es decir, usa información de segundo orden (matriz hessiana), de aquí que se le clasifique como método de segundo orden.

Considérese la aproximación a segundo orden de $f(x)$

$$f(x + \Delta x) = f(x) + \nabla^T f(x) \Delta x + \tfrac{1}{2} \Delta x^T \nabla^2 f(x) \Delta x$$

Si se supone que la aproximación anterior es buena, como de hecho lo es en la vecindad de un mínimo y de un máximo, entonces, al derivar parcialmente $f(x + \Delta x)$ con respecto a cada uno de los elementos de Δx se obtiene

$$\frac{\partial f(x+\Delta k)}{\partial \nabla_x} = \nabla f(x) + \nabla^2 f(x) \Delta x = 0$$

de donde

$$\Delta X = - H^{-1}(x) \nabla f(x)$$

donde $H(x) = \nabla^2 f(x)$ es la matriz hessiana $\left(hij = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}\right.$

y ΔX será la dirección del desplazamientó desde un punto x_k a x_{k+1} es decir

$$X_{k+1} = X_k \Gamma H^{-1}(X_k) \nabla f(x_k)$$

El empleo de la última fórmula para generar los iterandos del método de Newton puede provocar los siguientes problemas si es que el método se está empleando para minimizar una función $f(x)$. El problema es el siguiente. Note que tanto un máximo como un mínimo, así como los puntos silla, satisfacen $\partial f(x+\Delta x)/\partial\Delta x = 0$, por lo que no se puede garantizar que el método converga a un mínimo, si no se modifica adecuadamente. Esta modificación consiste en lo siguiente :

sea $\Delta X_k = -H^{-1}(x_k)\ \nabla f(x_k)$ la dirección de avance del punto x_k al x_{k+b} y considérese la aproximación a primer orden de $f(x)$ como sigue :

$$f(x_k + \Delta x_k) = f(x_k) + \alpha_k(\nabla^T f(x_k)\ \Delta x_k)\rho_k$$

o bien

$$\Delta f_k = f(x_k + \Delta x_k) - f(x_k) = \alpha_r(\nabla^T f(x_k)\ \Delta x_k)\ \rho k$$

donde α_k es el tamaño de paso, descrito en el inciso anterior, y ρ_k es el sentido de la dirección de búsqueda, el cual se determina como sigue : ya que se desea que $\Delta f_k < 0$ ($f(x_{k+1}) < f(x_k)$ y como $\alpha_k > 0$, entonces

$$\rho_k = \begin{cases} 1 & \text{si } \nabla^T f(x_k) H^{-1}(x_k) \nabla f(x_k) > 0 \\ -1 & \text{si } \nabla^T f(x_k) H^{-1}(x_k) \nabla f(x_k) < 0 \end{cases}$$

Y con esto se garantiza que si $\alpha_k (>0)$ es suficientemente pequeña entonces $f(x_{k+1}) < f(x_k)$ y el método de Newton convergerá a un mínimo, o en el peor de los casos a un punto silla, pero nunca a un máximo (se omite la discusión sobre el tamaño de paso α_k, por ser idéntica a la presentada con anterioridad).

Tomando en cuenta los puntos anteriores, la implementación del método de Newton queda como sigue

1. $k = 0$
2. Estimar x_o (punto de arranque)
3. Calcular $f(x)$.
4. Calcular $\nabla f(x_k)$
5. Si $\nabla^T f(x_k)\ \ \nabla f(x_k) \leq$ tolerancia, se ha encontrado un punto estacionario de $f(x)$ (mínimo o punto silla).
6. Calcular $H(x_k)$ y $H^{-1}(x_k)$
7. $$\rho_k = \begin{cases} 1 & \text{si } \nabla^T f(x_k)\ H^{-1}(x_k)\ \nabla f(x_k) > 0 \\ -1 & \text{si } \nabla^T f(x_k)\ H^{-1}(x_k)\ \nabla f(x_k) < 0 \end{cases}$$
8. $\Delta x_k = -\rho k\ H^{-1}(x_k)\nabla f(x_k)$
9. Calcular α_k (inciso 2.1)

   como resultado se obtiene x_{k+1}, $f(x_{k+1})$

10. $k \longleftarrow k+1$. Se repite el procedimiento desde (4).

Es fácil ver que si la función objetivo es cuadrática, p. ej.

$$f(x) = a + b^T x + \frac{1}{2} x^T Q x$$

el método de Newton converge en una sola iteración, ya que

$$\nabla f(x_k) = b + Q x_k$$

y

$$\nabla^2 f(x_k) = Q$$

entonces

$$x_{k+1} = x_k - \left(\nabla^2 f(x)\right)^{-1} \nabla f(x_k) = x_k - Q^{-1}(b + Qx_k)$$

$$\boxed{Q^{-1} b = x^*}$$

convergencia cuadrática

2.3. Conjugancia y Direcciones Conjugadas

Como se verá más adelante, una función objetivo cuadrática de n-variables que exhibe un mínimo, puede ser minimizada en n pasos (o menos) si estos pasos se toman en lo que se ha llamado direcciones conjugadas. En el inciso siguiente se limitará la discusión a funciones cuadráticas del tipo

$$f(x) = a + b^T x + \frac{1}{2} x^T H x \qquad (o)$$

donde H es una matriz positiva definida.

2.3.1. Conjugancia

Supóngase que la minimización de $f(x)$ empieza en x_o en la dirección S^o, escogida arbitrariamente (o por algún algoritmo); se supondrá $(S_o^o)^T S_o^o = 1.0$. El siguiente punto generado por el algoritmo será

$$x_1 = x_o + \lambda_o S^o \qquad (1)$$

donde el tamaño de paso λ_o se determina minimizando $f(x_o + \lambda_o S_o$ con respecto a λ, es decir

$$\frac{df(x_o + \lambda_o S_o)}{d\lambda} = 0 = \nabla^T f(x_o) S_o + (S_o)^T \nabla^2 f(x_o) \left[S_o \lambda\right]$$

de donde

$$\lambda_o = - \frac{\nabla^T f(x_o) S_o}{S_o^T \nabla^2 f(x_o) S_o} \qquad (2)$$

Una vez que se encuentra el siguiente iterando, x_1, se deberá seleccionar una nueva dirección de búsqueda para la minimización de $f(x)$. La nueva dirección S_1 se dice que es conjugada con S_o si $(S_1)^T \nabla^2 f(x_o) S_o = 0$. (En general, un conjunto de n direcciones independientes de búsqueda s_o, s_1, ... S_{n-1} son conjugadas con respecto a una matriz Q, positiva definida, si

$$s_i^T Q s_j = 0 \qquad 0 \le i \ne j \le n-1 \qquad (1)$$

Q, podría ser, por ejemplo, la matriz hessiana de la función objetivo H. Note además que si, Q = I, la matriz unitaria, conjugancia y ortogonalidad son sinónimos).

Debido a que los vectores si, son linealmente independientes además de conjugados, cualquier vector $V \in E^n$, se puede representar en términos de aquellos, como

$$V = \sum_{j=0}^{n-1} V_j \, s_j$$

$$V_j = \frac{s_j^T \; H(x) \; V}{s_j^T \; H(x) \; s_j}$$

Otra relación importante que se utilizará más adelante es la siguiente. Considerese la matriz P definida por

$$P = \sum_{j=0}^{n-1} a_j \; s_j \; s_j^T$$

Resulta obvio que si las a_j se escogen de manera tal que

$$P H s_k = s_k$$

entonces $P = H^{-1}$. Ahora bien,

$$P H s_k = \left(\sum_{j=0}^{n-1} a_j s_j s_j^T \right) H s_k = \sum_{j=0}^{n-1} a_j s_j (s_j^T H s_k)$$

$$= a_k \cdot s_k \; (s_k^T H s_k)$$

de donde si $a_k = (s_k^T H s_k)^{-1}$, entonces $P = H^{-1}$,

es decir

$$\boxed{H^{-1} = \sum_{j=0}^{n-1} \frac{s_j \; s_j^T}{s_j^T \; H \; s_j}}$$

si las direcciones de búsqueda empleadas en la minimización de f (x) se escogen conjugadas (esto se demostrará más adelante) a continuación se demuestra que : cualquier función cuadrática de n variables que exhibe un mínimo, puede ser minimizada en n pasos, si se emplean direcciones conjugadas, una dirección diferente en cada paso. Además, el orden en que se usan las direcciones de búsqueda es irrelevante para alcanzar el mínimo.

Demonstración. Sea $f(x) = a + b^T x + \frac{1}{2} x^T H x$, $\nabla f(x) = b + Hx$. $\nabla f(x) = H$, y en el mínimo de x, $\nabla f(x^*) = 0$, es decir $x^* = -H^{-1} b$

Para la n-ésima etapa se tiene, usando (1) y (2), que

$$x_n^n = x_0 + \sum_{k=0}^{n-1} \lambda_k \bar{s}_k$$

y como en cada etapa se usó el valor óptimo de , dado por la ecuación (2),

$$x_n = x_0 - \sum_{k=0}^{n-1} \frac{(\bar{S}_k)^T f(x_k)}{(\bar{S}_k)^T H \bar{S}_k} \bar{S}_k \qquad (5a)$$

Por otro lado

$$(\bar{S}_k)^T \nabla f(x_k) = (\bar{S}_k)^T (Hx_k + b)$$

$$= (\bar{S}_k)^T \left[H(X_0 + \sum_{i=1}^{n-1} \lambda_i \bar{S}_i) + b \right]$$

$$= (\bar{S}_k)^T (H_0 + b) \quad \text{o por conjugancia de la } \bar{S}_i$$

Entonces

$$x_n = x_0 - \sum_{k=0}^{n-1} \frac{(\bar{S}_k)^T (H x_0 + b) \bar{S}_k}{(\bar{S}_k)^T H \bar{S}_k} \qquad (5\ b)$$

Usando la relación (4 a) se obtiene que

$$X_0 = \sum_{k=0}^{n-1} \frac{(\bar{S}_k)^T H x_0}{(\bar{S}_k)^T H \bar{S}_k} \bar{S}_k$$

y por lo tanto

$$x_n = \sum_{k=0}^{n-1} \frac{(\bar{S}_k)^T b \bar{S}_k}{(\bar{S}_k)^T H \bar{S}_k} = - \sum_{k=0}^{n-1} \frac{(\bar{S}_k)^T H (H^{-1} b) \bar{S}_k}{(\bar{S}_k)^T H \bar{S}_k}$$

y usando (4 a) nuevamente, se obtiene

$$\underline{x_n = - H^{-1} b} \qquad \text{l.q.d.}$$

Un método para el cual se garantiza que alcanza el mínimo de una función objetivo cuadrática en un número específico de pasos, se dice que tiene la propiedad de terminación cuadrática. (El método de gradiente conjugado necesita de n pasos, mientras que el de Newton uno sólo).

2.3.2. Método de Gradiente conjugado

El método de Fletcher - Reeves de gradiente conjugado, que a continuación se describe, genera una secuencia de direcciones de búsqueda que son combinación lineal de $-\nabla f(x_k)$, la dirección de máximo descenso en el último punto, y de las k direcciones de búsqueda anteriores, s_0, s_1, ..., s_{k-1}, usando factores de peso tales que S_k sea conjugada a las direcciones anteriores.

Para ilustrar el método, sea $S_0 = - \nabla f(x_0)$, y

$x_1 = x_0 + \lambda_0^* S_0$; sea

$$S_1 = - \nabla f(x_1) + a_1 S_0 \qquad (6)$$

donde a_1 se escoge de forma tal que S_0 y S_1 sean conjugadas con respecto a H, es decir

$$S_0^T H S_1 = 0 \qquad (7)$$

Para eliminar S_o, considérese la aproximación a primer orden del gradiente, es decir

$$\nabla f(x_1) - \nabla f(x_o) = \nabla^2 f(x_o)\ (x_1 - x_o) \qquad (8)$$

$$= \lambda_o^* \ H\ S_o$$

$$S_o^* = \frac{1}{\lambda_o^*} H^{-1}\Big[\nabla f(x_1)\nabla f(x_o)\Big]$$

y como H es simétrica, entonces

$$S_o^T = \frac{\Big[\nabla f(x_1) - \nabla f(x_o)\Big]^T H^{-1}}{\lambda_o^*} \qquad (9)$$

Substituyendo (6) y (9] en (7) se obtiene

$$\Big[\nabla f(x_1) - \nabla f(x_o)\Big]^T \Big[-\nabla f(x_1) + \alpha_1 S_o\Big] = 0 \qquad (10)$$

Y ya que, según se vió con anterioridad $\nabla^T f(x_o)\ \nabla f(x_1) = \nabla^T f(x_1)\ S_o = 0$, entonces

$$\alpha_1 = \frac{\nabla^T f(x_1)\ f(x_1)}{\nabla^T f(x_o) S_o}.$$

o bien,

$$\alpha_1 = \frac{\nabla^T f(x_1)\nabla f(x_1)}{\nabla^T f(x_o)\nabla f(x_o)} \qquad (11)$$

La dirección de búsqueda S_2 se forma como una combinación lineal de $-\nabla f(x_2)$, S_1 y S_o, y se fuerza a que sea conjugada a S_1 y S_o

con lo que se obtiene la siguiente expresión para los factores de peso α_k

$$\alpha_k = \frac{\nabla^T f(x_k)\ \nabla f(x_k)}{\nabla^T f(x_{k-1})\ \nabla f(x_{k-1})} \qquad (12)$$

La implementación del algoritmo de gradiente conjugado de Fletcher - Reeves incluye los siguientes pasos :

1. Estimar x_o (punto de arranque)

2. $S_o = -\nabla f(x_o)$

3. En la k - ésima etapa del algoritmo, se determina el mínimo unidireccional de $f(x)$, a lo largo de la dirección de búsqueda S_k. Con esto se localiza x_{k+1}.

4. La nueva dirección de búsqueda se determina como

$$S_{k+1} = -\nabla f(x_{k+1}) + \frac{\nabla^T f(x_{k+1})\ \nabla f(x_{k+1})}{\nabla^T f(x_k)\ \nabla f(x_k)}\ S_k$$

Después de (n+1) iteraciones (k = n), se empieza un nuevo ciclo del algoritmo, es decir, x_{n+1} se convierte en x_o.

5. La búsqueda se da por terminada cuando en alguna iteración suceda que

$$S_k^T \cdot S_k \leq \text{tolerancia}$$

Nota que al igual que en el método de gradiente ordinario, no se necesita la inversión de matriz alguna, lo cual es una ventaja.

2.4. Métodos de Métrica Variable

Los métodos de Métrica Variable o Cuasi-Newton son métodos que aproximan el hessiano, o su inverso, usando unicamente información acerca del gradiente. La mayoria de estos métodos usan direcciones conjugadas conforme avanzan, lo cual hacen siguiendo el esquema general

$$x_{k+1} = x_k + \lambda_k s_k = x_k - a_k \eta(x_k) \nabla f(x_k) \qquad (1)$$

donde $\eta(x_k)$ representa una aproximación a $H^{-1}(x_k)$. (En el método de Gradiente ordinario $\eta(x_k) = I$, mientras que el método de Newton toma $\eta(x_k) = H^{-1}(x_k)$, con la desventaja de que hay que invertir el hessiano).

En una serie de métodos de Cuasi-Newton, $H^{-1}(x_{k+1})$ se aproxima de la información disponible en la k-ésima etapa, como

$$H^{-1}(x_{k+1}) \approx \omega\eta_{k-1} = \omega(\eta_k + \Delta\eta_k) \qquad (2)$$

donde η es una aproximación a H^{-1}, $\Delta\eta_k$ es una matriz que se especifica de acuerdo al método, y ω una constante de escalamiento que frecuentemente se fija en 1 . La selección de determina, esencialmente, el método de método variable. Para garantizar convergencia, $\omega\eta_{k+1}$ debe ser positiva definida y debe satisfacer la siguiente relación cuando reemplaza a H^{-1}

$$x_{k+1} - x_k = H^{-1}(x_k)\left[\nabla f(x_{k+1}) - \nabla f(x_k)\right] \qquad (3\ a)$$

que es una aproximación a primer orden del gradiente.

En la (k^{+1})-ésima etapa de cualquier método se conocen

x_k, $\nabla f(x_k)$ x_{k+1} $\nabla f(x_{k+1})$ y η_k , y se desea calcular η_{k+1}

De la relación obtenida con (2) y (3) como

$$\eta_{k+1}\Delta g_k = \frac{1}{\omega}\Delta x_k \qquad (3\ b)$$

donde $\Delta g_k = \nabla f(x_{k+1}) - \nabla f(x_k)$. Sea $\Delta\eta_k = \eta_{k+1} - \eta_k$ por lo que la ecuación

$$\Delta\eta_k \Delta g_k = \frac{1}{\omega}\Delta x_k - \eta_k \Delta g_k \qquad (3\ c)$$

debe ser resuelta para $\Delta\eta_k$, y esta se obtiene como sigue. Si el lado derecho de (3 c) se multiplica y divide por $y^T\Delta g_k$, el primer término, y $z^T\Delta g_k$ el segundo término se obtiene que:

$$\left[\Delta\eta_k - \left(\frac{1}{\omega}\frac{\Delta x_k y^T}{y^T\Delta g_k} - \frac{\eta_k \Delta g_k z^T}{z^T\Delta g_k}\right)\right]\Delta g_k = 0 \quad ,$$

o bien,

$$\Delta\eta_k = (\frac{1}{w}\ \frac{\Delta x_k y^T}{y^T \Delta g_k} - \frac{\eta_k \Delta g_k}{Z^T \Delta g_k} Z^T \qquad (4)$$

donde los vectores columna y, Z son arbitrarios, al igual que w. Si por ejemplo se escogen

$$w = 1$$

$$y = Z = \Delta x_k + \eta_k \Delta g_k$$

se genera el algoritmo de Broyden, mientras que si se escogen

$$y = \Delta x_k$$

$$Z = \eta_k \Delta g_k$$

entonces la matriz η_{k+1} se actualiza de acuerdo al método de Davidon-Fletcher-Powell. Ya que los vectores y, Z son arbitrarios se pueden efectuar varias selecciones, las que se discuten más adelante. Si los pasos x_k se determinan mediante minimizaciones unidireccionales de $f(x)$ en la dirección s_k, todos los métodos que calculan una η_{k+1} simétrica que satisfagan (3 b), generan direcciones que son mutuamente ortogonales (para funciones cuadráticas).

2.4.1. $\Delta\eta_k$ de Rango 1

Broyden demostró que si $\Delta\eta_k$ es simétrica con rango 1, la relación $\eta_{k+1}\Delta g_k = \Delta x_k$ se satisface, la única posibilidad de escoger $\Delta\eta_k$ es

(5)

El algoritmo funcionaría de la siguiente forma. Se escogen x_0 y $\eta_0 > 0$ y se usa una forma secuencial (1), ( ) y (2) hasta que por ejemplo $\nabla^T f(x_k)\nabla f(x_k) \leq \epsilon$. Por otro lado, si se usan minimizaciones unidireccionales, el método genera direcciones conjugadas y bajo algunas condiciones más o menos restrictivas, se puede demostrar que el algoritmo converge a la solución. Una característica atractiva de este método es que α_k en (1) no necesariamente tiene que ser un parámetro que minimize $f(x)$ a lo largo de s_k. El mismo Broyden demuestra que α puede tomar cualquier valor con la única condición de que no provoque que η se haga singular (denominador en (5)).

Si la función objetivo no es cuadrática, algunos de los aspectos poco satisfactorios al usar (5), son los siguientes :

1. puede dejar de ser positiva definida, en cuyo caso es necesario recurrir a alguna otra estrategia que lo garantice.

2. La corrección $\Delta\eta_k$ puede no quedar acotada (generalmente por errores de redondeo, incluso para funciones cuadráticas)

3. Si por coincidencia $\Delta x_k = -\alpha_k \eta(x_k)\nabla f(x_k)$ queda en la dirección del paso anterior, $\eta(x^B_{k-1})$ se vuelve singular.

En consequencia, en el algoritmo de Broydon, si sucede que

$$\eta_k \Delta g_k = \Delta x_k$$

o

$$(\eta_k \Delta g_k - \Delta x_k)^T \Delta g_k = 0$$

se fuerza a que

$$\eta_{k-1} = \eta_k \qquad (\Delta\eta_k = 0)$$

2.4.2. Método de Davidon - Fletcher - Powell

En este método la matriz $\Delta\eta$ se escoge que tenga rango 2. La η inicial normalmente se toma como $\eta_0 = I$. (se puede usar cualquier otra matriz simétrica positiva definida), con lo que el método arranca con la dirección del máximo descenso. - Conforme el método avanza, va existiendo un cambio del método de gradiente a Newton con lo que se obtiene una gran ventaja al usar las mejores características de ambos métodos.

Como se mencionó con anterioridad la relación para $\Delta\eta_k$ en el método de Davidon - Fletcher - Powell, $y_k = \Delta x_k$ y $z_k = \eta_k \Delta g_k$ con lo que al substituir en (4) se obtiene

$$\eta_{k-1} = \eta_k + A_L + B_k$$
$$= \eta_k + \frac{\Delta x_k (\Delta x_k)^T}{(\Delta x_k)^T \Delta x_k} - \frac{\eta_k \Delta g_k (\Delta g_k)^T \eta_k^T}{(\Delta g_k)^T \eta_k \Delta g_k} \tag{5}$$

en donde las matrices A_k y B_k son simétricas y si, además, η_k es también simétrica, entonces $_{k+1}$ también lo será. La relación anterior (Ec. (5)) produce resultados satisfactorios en la práctica siempre y cuando

1. El error al evaluar $\nabla f(x_k)$ no sea grande
2. η_k no se haga mal-condicionada

El papel de la matriz A_k en la ecuación (5) es garantizar que $\eta \longrightarrow H^{-1}$, mientras que la matriz B_k garantiza que η_{k+1} sea positiva definida en todos los pasos, y en el límite se cancela con η_0. Esto se puede ver como sigue

$$\eta_1 = I + A_0 - B_0$$
$$\eta_2 = \eta_1 + A_1 - B_1 = I + (A_0 + A_1) - (B_0 - B_1)$$
$$\eta_{k+1} = I + \sum_{i=0}^{k} A_k - \sum_{i=0}^{k} B_k$$

Para una función cuadrática la suma de las matrices A_i debe ser igual a H^{-1} cuando $k = n - 1$, y la suma de las matrices B_i deberá cancelar η_0 (I en este caso), se puede decir que el método de Davidon - Fletcher - Powell refleja, en cierta forma, toda la información ganada en iteraciones anteriores, a través de η_k.

Debe señalarse que el método que se está describiendo usa direcciones conjugadas si la función objetivo es cuadrática. Para que la última dirección, S_{n-1}, sea conjugada a todas las anteriores, se debe

cumplir que :

$$X^T_{n-1} \; H \; s_{n-1} = 0$$

si se substituye que $s_{n-1} = - \eta_{n-1} \nabla f(x_{n-1})$, entonces

$$X^T_{n-1} \; H \; \eta_{n-1} \; f(x_{n-1}) = 0 \qquad (6)$$

donde $X_{n-1} = \left[\Delta x_0, \; \Delta x_1, \; \ldots, \; \Delta x_{n-1}\right]$. Si $H \eta_{n-1} = I$ ($\eta_{n-1} = H^{-1}$), entonces $\nabla f(x_{n-1})$ es conjugada a todas las direcciones de búsqueda anteriores dadas por $\Delta x_0, \; \Delta x_1, \ldots, \; \Delta x_{n-1}$. Sabiendo que todas las direcciones de búsqueda son conjugadas, se puede demostrar que $\sum_{i=0}^{n-1} A_i = H^{-1}$, como sigue. Como $\Delta g_k = H \, \Delta x_k$, entonces el numador y denominador de cada A_i es

$$(\Delta x_k)(\Delta x_k)^T = (\lambda_k s_k)(\lambda_k s_k)^T = \lambda_k^2 \; s_k \; s_k^T$$

$$(\;x_k)^T \;\; g_k = (\;_k s_k)^T (H \;\; _k s_k) = \;\; _k^2 \; s_k^T \; H \; s_k$$

de donde

$$\sum_{i=0}^{n-1} A_i = \sum_{i=0}^{n-1} \frac{s_i s_i^T}{s_i^T H s_i} \qquad (7)$$

que es la fórmula (4 b). Sec. 2.3. obtenida con anterioridad.

Para terminar la presentación de este método, se hacen dos comentarios sobre la implementación práctica del mismo

1. En algunos problemas, los métodos de métrica variable fallan en alcanzar el mínimo de la función objetivo si el grado de precisión en la búsqueda unidimensional no es suficientemente . Se recomienda que la precisión en la búsqueda unidireccional sea al menos equivalente que en que se requiere para detener el algoritmo completo.

2. La búsqueda por el mínimo se debe detener, si al evaluar los vectores. $-\eta_k \; \Delta f(x_k)$ y- $-\alpha_k \; \eta_k \nabla f(x_k)$ ocurre cualquiera de los dos siguientes puntos.

   a) Cada componente en ambos rectores es menor que una, tolerancia dada.

   b) Las longitudes predichas al mínimo, de cualquiera de los dos vectores es inferior a una cierta tolerancia.

2.4.3. Algoritmos de Pearson

Pearson propuso una serie de algoritmos para calcular η_i usando direcciones que fueran conjugadas. Los algoritmos de Pearson se pueden obtener empleando diferentes vectores y, Z en la ecuación (4) del inciso 2.4., según se muestra a continuación

1. Pearson No. 2 Sea y = Z = Δx_k y w = 1.

Entonces

$$\eta_{k+1} = \eta_k + \frac{(\Delta x_k - \eta_k \Delta g_k)(\Delta x_k)^T}{(\Delta x_k)^T \Delta g_k}$$

$$\eta_0 = R_0$$

donde R_0 es cualquier simétrica positiva definida. Este algoritmo generalmente conduce a matrices mal condicionadas.

2. Pearson No. 3 Sea y = Z = $\eta_k \Delta g_k$, con w = 1.

Entonces

$$\eta_{k+1} = \eta_k + (\Delta x_k - \eta_k \Delta g_k) \cdot \frac{(\eta_k \Delta g_k)^T}{(\Delta g_k)^T \eta_k \Delta g_k}$$

$$\eta_0 = R_0.$$

Este algoritmo se comporta bastante parecido al de Davidon-Fletcher-Powell, excepto que el tamaño del paso es, en general, inferior al de este último.

3. Newton-Raphson Proyectado.

Pearson propuso este otro algoritmo al cual se puede obtener haciendo que x ⟶ w y Z = $\eta_k \Delta g_k$, con lo que se obtiene.

$$\eta_{k+1} = \eta_k - \frac{(\eta_k \Delta g_k)(\eta_k \Delta g_k)^T}{(\Delta g_k)^T \eta_k \Delta g_k}.$$

$$\eta_0 = R_0$$

Este método incluye la siguiente regla de reinicio cada n etapas, donde n es el número, de variables independientes, sobre la matriz η_k.

$$\eta_{k+1} = \eta_k + \frac{(\Delta x_k - \eta_k \Delta g_k)(\eta_k \Delta g_k)^T}{(\Delta g_k)^T \eta_k \Delta g_k}$$

es decir, cada n etapas se toma $\eta_k = R_k$.

BIBLIOGRAFIA

1. C.G. Broyden, Math. Computation, 21 : 368 (1967)

2. M.J. Box, D. Davis, and W.H. Swann. "Non-linear Optimization Techniques", Chemical Industries Monograph 5, Oliver and Boyd, Edinburgh, 1970

3. G.F. Coggins, Univariate Search Methods, Imperial chemical Industries, Ltd., Central Instr. Lab. Res., Note 64/11, 1964

4. W.C. Davidon, USAEC Doc. ANL - 5990 (Rev.), Nov., 1959

5. W.C. Davidon, Computes J., 10 : 406 (1968) ; Chap. 2 in R. Fletcher (ed.), "Optimization" Academic Press Inc. N.Y., 1969

6. R. Fletcher and M.J.D. Powell, Computer J., 6 : 163 (1963)

7. R. Fletcher and C.M. Reeves, Computer J., 7 : 149 (1964)

8. M.R. Hestenes, The Conjugate Gradient Method for Solving Linear Systems, in Proceedings of the Symposium on Applied Mathematics, Vol. VI, Mc. Graw-Hill Book Company, New York, 1956, pp. 83-102.

9. M.R. Hestenes and E.L. Steifel, J. Res. Nat. Bur. Std., B 49 : 409 (1952)

10. J.D. Pearson, Computer J., 13 : 171 (1969)

11. M.J.D. Powell, Computer J., 7 : 155 (1964)

12. M.J.D. Powell, Rank One Methods for Unconstrained Minimization, AERE Rept., TP 372, 1969

13. D.J. Wilde, "Optimum Seeking Methods", Prentice Hall, Inc., Englewood Cliffs, N.J., 1964.

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO OPTIMO DE SISTEMAS DE INGENIERIA

CONDICIONES DE OPTIMALIDAD

DR.ANTONIO MONTALVO ROBLES

MARZO, 1982

CONDICIONES DE OPTIMALIDAD

I. EL PROBLEMA GENERAL DE PROGRAMACION NO LINEAL

En el sentido más amplio, el problema general no lineal es el de encontrar un extremo (máximo o mínimo) de una función objetivo sujeta a restricciones de igualdad y/o no lineales. Sin embargo, en las siguientes secciones de estas notas han quedado excluidos los dos siguientes problemas:

a) Las variables estan restringidas a valores enteros (programación no lineal entera)

b) Las restricciones incluyen al parámetro tiempo en la forma de una ecuación diferencial (control óptimo).

En lo siguiente se supondrá que la función objetivo $f(x)$ es continua, $h_1(x), \ldots, hm(x)$ denotan las restricciones de igualdad y $g_{m+1}(x), \ldots, g_p(x)$ las restricciones de desigualdad, donde: $x = (x_1, \ldots, x_n)^T$ es un vector columna de componentes $x_1, \ldots, x_n$ en un espacio euclidiano n-dimensional. (Las variables $x_1, x_2, \ldots, x_n$ pueden ser parámetros de diseño, ajuste de controles, lecturas de instrumentos, etc., mientras que la función objetivo podría representar el costo, peso, ganancias, etc.; finalmente, las restricciones pueden representar requerimientos técnicos, condiciones de operación, etc., del proceso).

El problema de programación no lineal se puede establecer formalmente como:

$$\text{Minimizar:} \quad f(x), \; x \in E^n \qquad (1.1)$$

sujeta a m restricciones de igualdad, lineales y/o no lineales,

$$h_j(x) = 0 \qquad j = 1, \ldots, m \qquad (1.2)$$

y $(p - m)$ restricciones de desigualdad, lineales y/o no lineales,

$$g_j(x) \geq 0 \qquad j = m+1, \ldots, p \qquad (1.3)$$

O en forma alterna como:

$$\text{Minimizar:} \quad \{f(x) | x \in M\} \qquad (1.4)$$

donde M es el dominio de x para el cual (1.2) y (1.3) se satisfacen, es decir:

$$M = \{x | h_j(x) = 0, \; g_j(x) \geq 0, \text{ para todo } j\} \qquad (1.5)$$

Un ejemplo sencillo de programación no lineal es el que se muestra en la figura 1. y está dado por:

Minimice: $f(x) = x_1^2 + x_2^2 + 2 x 2$

sujeto a: $h_1(x) = x_1^2 + x_2^2 - 1 = 0$

$$g_2 = x_1 + 2x2 - \tfrac{1}{2} \geq 0$$

$$g_3 = x_1 \geq 0$$

$$g_4 = x_2 \geq 0$$

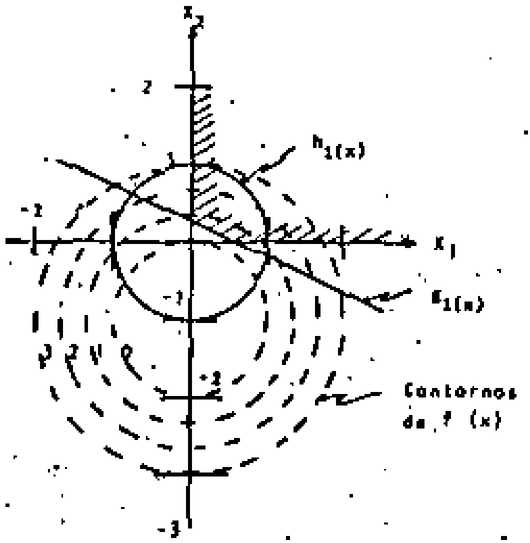


FIGURA I. Representación geométrica de un problema de programación

II. NOTACION Y TERMINOLOGIA

El vector columna $x^* = [x_1^*, \ldots, x_n^*]^T$ que satisface (1.1) - (1.3) se denomina punto óptimo, y el valor de $f(x^*)$ que le corresponde se denomina valor óptimo de la función objetivo. La pareja, x^*, $f(x^*)$ constituye la solución óptima. Para algunos problemas, pueden existir varias categorias de soluciones óptimas si la función objetivo no es unimodal (exhibe mas de un punto extremo) como se ilustra en la figura 2. La solución global óptima representa el valor más pequeño de $f(x)$, mientras que una solución óptima local (o relativa) representa el valor más pequeño de $f(x)$ en una cierta vecindad del vector x: es decir,

óptimo global x^* satisface $\quad f(x^*) < f(x) \forall x \in E^n$

óptimo local x^* satisface $\quad f(x^*) < f(x) ||x-x^*|| \leq \varepsilon(x^*)$

2.1. Concavidad y Convexidad

Los conceptos de concavidad y convexidad ayudan a determinar bajo que condiciones una solución óptima local es tambien solución óptima global.

Una función $\phi(x)$ se dice que es convexa en el dominio R, si para cualesquiera dos vectores x_1 y x_2 ϵR,

$$\phi\left(\theta x_1 + (1-\theta)x_2\right) \le \theta\phi(x_1) + \phi(x_2)(1-\theta) \qquad (1.6)$$

donde θ es un escalar $0 \le \theta \le 1$. Además, $\phi(x)$ es <u>estrictamente</u> convexa si, para $x_1 \ne x_2$, el signo $\le$ en (1.6) se puede remplazar por el signo de desigualdad $(<)$. Si en (1.6) la desigualdad contraria es la válida, se dice que la función $\phi(x)$ es cóncava $(\ge)$ o <u>estrictamente</u> cóncava $(>)$. Note que si $\phi(x)$ es cóncava (convexa), $-\phi(x)$ es convexa (cóncava). (Las funciones lineales son, simultáneamente, convexas y cóncavas).

Una función convexa diferenciable posee las siguientes propiedades.

a) $\phi(x_2) - \phi(x_1) \ge \nabla^T\phi(x_1)(x_2-x_1)$ para toda x_1 y x_2

b) La matriz de segundas derivadas parciales de $\phi(x)$ con respecto a x (matriz Hessiana) es positiva definida (o positiva semidefinida) para toda x si $\phi(x)$ es estrictamente convexa (o convexa).

c) Sobre el dominio de R, $\phi(x)$ posee un sólo mínimo.

(1.7)

Un conjunto de puntos (o región) se define como <u>conjunto</u> convexo en un espacio n-dimensional si, para toda pareja de puntos x_1 y x_2 en el conjunto, la línea recta que los une pertenece completamente al conjunto. Es decir, R es convexo si para toda x_1 y $x_2 \in R$

$$x = \theta x_1 + (1-\theta)x_2 \in R$$

De los conceptos de convexidad emerge un resultado importante en programación matemática: para el problema de programación no lineal conocido como el problema de <u>programación convexa</u>

Minimizar : $f(x)$

sujeta a : $g_j(x) \ge 0$ $\quad j = 1, \ldots, p$

$x \ge 0$

en el cual (1), $f(x)$ es una función convexa y (2) cada restricción de desigualdad es una función cóncava (las restricciones forman un conjunto convexo), se puede demostrar el siguiente resultado: <u>el mínimo local también es mínimo global</u> (Usando argumentos opuestos, el resultado opuesto, máximo, también es cierto).

2.2. <u>Factibilidad</u>

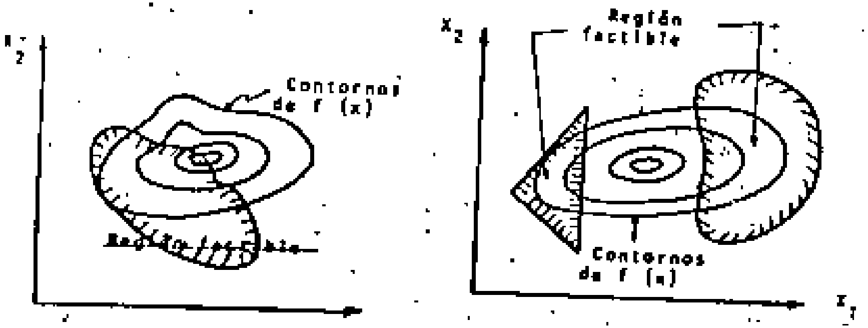
Cualquier vector x que satisface tanto las restricciones de desigualdad como las de igualdad se llama <u>punto factible</u>. El conjunto de todos los puntos que satisfacen las restricciones constituyen el <u>dominio factible</u> de $f(x)$, y se denotará por R; cualquier punto no en R se llama <u>punto no factible</u>.

Un <u>óptimo restringido</u> es uno para el cual el óptimo local cae en la frontera de la región factible. Si las restricciones son únicamente

de igualdad, un punto x factible debe caer en la intersección de todas las hipersuperficies que satisfacen $h_j(x) = 0$

Con respecto a las restricciones de desigualdad, un punto x se puede clasificar como <u>punto interior</u> (factible), <u>punto frontera</u> (factible) o <u>punto exterior</u> (no factible). Los puntos interiores son aquellos para los cuales $g_j(x) \quad 0$; para un punto frontera, g_j $g_j(x) = 0$ para al menos una restricción; y un punto exterior, $g_j(x) > 0$ para al menos una restricción. Las restricciones se llaman <u>activas</u> (o de atadura) si $g_j(x) = 0$.

Una región $\Re$ de vectores admisibles puede ser <u>convexa</u> o <u>no convexa</u>, según se describió con anterioridad, pero además puede ser <u>simplemente conexa</u> o <u>no-simplemente conexa</u> (ver Figura 2).



(a) simplemente conexa
(no convexa)

(b) no simplemente conexa
(<u>obviamente</u> no convexa)

FIGURA 2. Ejemplos de tipos de región

2.3. <u>El Gradiente</u>

El conjunto de puntos para los cuales una función f(x) exhibe un valor constante, se llaman <u>contornos</u> de f(x). Si la función f(x) es continua y diferenciable, el <u>gradiente</u> de la función existe y está <u>definido</u> como el <u>vector columna</u> formado por las primeras derivadas parciales de f(x) con respecto a x es decir:

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix} \qquad (1.8)$$

Se puede demostrar que en el espacio métrico euclidiano, el gradiente de una función escalar apunta en la dirección de máximo incremento en el valor de la función, <u>máximo ascenso</u>, y que es, además, ortogonal a las líneas de contorno. El negativo del gradiente apunta en la dirección del <u>máximo descenso</u> de f(x). Finalmente, cualquier vector V, ortogonal a $\nabla f(x)$, tal como la superficie tangente a f(x), está definido por $V^T \nabla f(x) = 0$ (Figura 3)

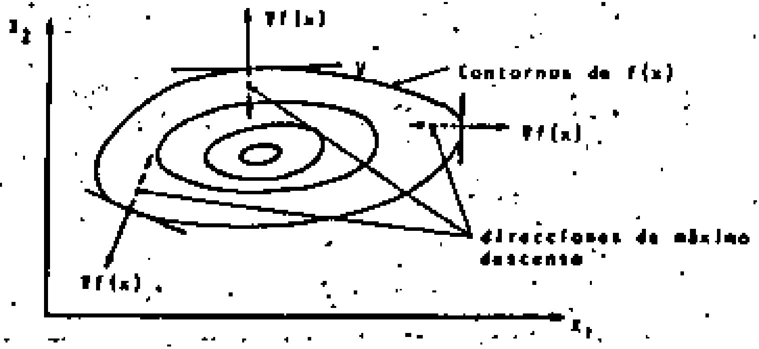


FIGURA 3. El gradiente, y la dirección de máximo descenso.

2.4. Aproximación de Funciones

Algunos de los procedimientos de programación matemática que se discutirán más tarde requieren de aproximaciones lineales o cuadráticas para $f(x)$, $g(x)$ y $h(x)$.

Una aproximación lineal, o de primer orden, para una función $f(x)$, se puede hacer truncando la serie de Taylor, alrededor de un punto x_0, como

$$f(x) = f(x_0) + \nabla^T f(x_0)(x-x_0) \tag{1.9}$$

Para obtener una aproximación cuadrática, en la misma serie de Taylor se pueden despreciar los términos mayores e iguales a tercer orden, obteniéndose

$$f(x) = f(x_0) + \nabla^T f(x_0)(x-x_0) + \frac{1}{2}(x-x_0)^T \nabla^2 f(x_0)(x-x_0) \tag{1.10}$$

donde $\nabla^2 f(x)$ es la matriz Hessiana a $f(x)$, $H(x)$, es decir

$$H(x_0) = (h_{ij}(x_0)) \quad i,j = 1, \ldots, n \tag{1.11a}$$

donde

$$h_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \tag{1.11b}$$

Se observa fácilmente que $H(x)$ es una matriz simétrica.

2.5. Condiciones Necesarias y Suficientes para que Una Solución sea Solución Optima

Para algunas clases especiales del problema general de programación no lineal (Ecs (1.1) - (1.3)) ha sido posible establecer criterios de optimalidad. Sin embargo, para funciones completamente generales, no ha sido posible establecer criterios de optimalidad precisos. En consecuencia, únicamente se describirán algunos casos especiales, los cuales, sin embargo, son bastante comunes y de importancia práctica. Las condiciones que determinan si un vector x resuelve o no el problema de programación no lineal serán presentadas en una serie de teoremas los cuales no serán demostrados (las demostraciones están fuera de los objetivos de este curso).

2.5.1. Programación no-lineal sin restricciones

El problema es el siguiente:

$$\text{Minimizar:} \quad f(x), \; x \in E^n \qquad (1.12)$$

Las condiciones necesarias para que x^* sea un mínimo local del problema (1.12) son:

1. $f(x)$ diferenciable en x^*
2. $\nabla f(x^*) = 0$, es decir, existe un punto estacionario de $f(x)$ en x^*

Las condiciones suficientes para que x^* sea un mínimo local del problema (1.12) son:

3. $\nabla^2 f(x) > 0$; es decir, la matriz Hessiana es positiva definida.

(Las condiciones para la existencia de un máximo son iguales, excepto que el hessiano deberá ser negativo definido).

2.5.2. Programación no-lineal con restricciones de Igualdad y Desigualdad

En este caso el problema es el siguiente:

$$\left.\begin{array}{llll} \text{Minimizar:} & f(x) & & x \quad E^n \\ \text{Sujeta a:} & h_j(x) = 0 & & j = 1, \ldots, m \\ & g_j(x) \quad 0 & & j = m+1, \ldots, p \end{array}\right\} \quad (1.13)$$

Las condiciones necesarias para que x^* sea un mínimo local se establecen en dos teoremas, el primero de los cuales (teorema 2) puede ser llamado condiciones de primer orden (debido a que las funciones que intervienen se consideran una vez diferenciables). El segundo teorema (teorema 3) se le denomina condiciones de segundo orden (se considera que las funciones son dos veces diferenciables).

Para establecer las condiciones necesarias, empezaremos con el siguiente concepto: si x^* es un mínimo local de $f(x^*)$, ésta no puede decrecer a lo largo de ningún arco "suave" dirigido desde x^* hacia la región factible. Sea el vector V tangente al arco que empieza en x^*. Usando los conceptos de Fiacco y Mc Cormick, se asignan tres diferentes categorías o clases al vector V, donde cada conjunto V_i incluye el conjunto de V tales que: (ver tabla I).

Todas las posibles perturbaciones de x^* caen en la unión de V_1 y V_2 y si $V \in V_2$, $f(x)$ decrece, mientras que si $V \in V_1$, $f(x)$ se incrementa o es constante. En esencia, las condiciones necesarias de primer orden imponen el requerimiento de que el conjunto V_2 esté vacío.

	RESTRICCIONES				
CLASE	DESIGUALDAD		IGUALDAD		FUNCION OBJETIVO
V_1	$\left\{ v^T \nabla g_j (x^*) \geq 0 \right\}$ para las restricciones activas	y	$\left\{ v^T \nabla h_j (x^*) = 0 \right\}$ $j = 1,2, \ldots, m$	y	$\left\{ v^T \nabla f (x^*) \geq 0 \right\}$
V_2	$\left\{ v^T \nabla g_j (x^*) \geq 0 \right\}$ para las restricciones activas	y	$\left\{ v^T \nabla h_j (x^*) = 0 \right\}$ $j = 1,2, \ldots, m$	y	$\left\{ v^T \nabla f (x^*) < 0 \right\}$
V_3	$\left\{ v^T \nabla g_j (x^*) < 0 \right\}$ para, al menos, una restricción activa	o	$\left\{ v^T \nabla h_j (x^*) \neq 0 \right\}$ para, al menos, una restricción		

TABLA I. Clasificación de los conjuntos V_i

Si V_2 está vacío, se puede demostrar la existencia de los multiplicadores de Lagrange, resultando el siguiente teorema.

TEOREMA I.

Si (a), x^* satisface el problema (1.13), (b) las funciones $f(x)$, $g_j(x)$, son una vez diferenciables, y (c) en x^* V_2 está vacío, entonces existen los vectores u^* y w^* (multiplicadores de Lagrange) tales que, (w^*, v^*, x^*) satisfacen

(1) $h_j(x^*) = 0$ $\qquad j = 1, \ldots, m$

(2) $g_j(x^*) \geq 0$ $\qquad j = m+1, \ldots, p$

(3) $u_j^* \, g_j(x) = 0$ $\qquad j = m+1, \ldots, p$

(4) $u_j^* \geq 0$ $\qquad j = m+1, \ldots, p$

(5) $\nabla L(x^*, u^*, v^*) = 0$

donde la función

$$L(x, u, w) = f(x) + \sum_{j=1}^{m} w_j h_j(x) - \sum_{j=m+1}^{p} u_j g_j(x)$$

puede ser considerada como una *función Lagrangiana generalizada*, asociada al problema (1.13).

Con el propósito de establecer bajo que circunstancias el conjunto V_2 está vacío, se necesita calificar, a primer orden, a las restricciones.

Sea x^* un punto factible del problema (1.13) y supóngase que $h_1(x), \ldots, hm(x), g_{m+1}(x), \ldots, g_p(x)$ son funciones una vez diferenciables. La calificación a primer orden de las restricciones es una condición que se impone unicamente sobre las restricciones (sin importar la función objetivo) y que consiste en que para cada punto frontera formado por el conjunto de, restricciones de igualdad y las activas de desigualdad, debe de existir una curva suave que termine en el punto frontera y que pertenezca completamente al conjunto de las restricciones. Si x^* es un mínimo local de $f(x)$, éste no puede decrecer a lo largo de tal curva, dirigida desde x^* hacia la región factible. Una condición suficiente para la calificación a primer orden de las restricciones que debe cumplirse es que todos los gradientes de las restricciones de desigualdad activas y los gradientes de las restricciones de igualdad evaluados en x^*, sean linealmente independientes. Este último se establece en el siguiente teorema.

TEOREMA 2.

Si las funciones $h_1(x), \ldots, hm(x), g_{m+1}(x), \ldots, g_p(x)$ son una vez diferenciables en x^*, y si la calificación a primer orden de las restricciones es válida en x^*, entonces la condición necesaria para que x^* sea un mínimo local del problema (1.7), es que existan multiplicadores de Lagrange u^* y w^* tales que (u^*, v^*, x^*) satisfagan las ecuaciones (1) - (5) del Teorema 1.

Para tomar en cuenta la curvatura de las funciones en el problema (1.13), Mc Cormick estableció las condiciones necesarias de segundo orden para que x^* sea un mínimo local. Supongase que las funciones $f(x)$, $h_1(x), \ldots, hm(x), g_{m+1}(x), \ldots, g_p(x)$ son dos veces diferenciables en x^*, un punto que satisface al problema (1.13). Sea V cualquier vector no cero tal que:

$V^T \nabla g_j(x) = 0$ para las restricciones de desigualdad activas

$V^T \Delta h_j(x) = 0.$ para las restricciones de igualdad

Entonces, si V es la tangente a una curva $\phi(0)$, $\theta \geq 0$, dos veces diferenciable, a lo largo de la cual $g_i\{\phi(0)\} = 0$ para todas las restricciones de desigualdad activas, y $h_j\{\phi(0)\} = 0$ para todas las restricciones de igualdad, la calificación a segundo orden de las restricciones en x^* es válida. Una condición suficiente para la calificación a segundo orden de las restricciones que debe cumplirse es que los gradientes de las restricciones de desigualdad activas en x^* y los gradientes de las restricciones de igualdad en x^* sean linealmente independientes.

Las condiciones necesarias de segundo orden se pueden establecer como sigue.

TEOREMA 3.

(a) Si las funciones $f(x)$, $h_1(x)$, ..., $h_m(x)$, $g_{m+1}(x)$, ..., $g_p(x)$ son dos veces diferenciables en x^*, y (b) si la calificación a primer orden de las restricciones es válida en x^*, entonces las condiciones necesarias para que x^* sea un mínimo local del problema (1.13) son que existan u^* y v^* tales que (c) ecuaciones (1) - (5) del Teorema 1 se satisfagan, y (d) para cada vector no cero V, para el cual $V^T g_j(x^*) = 0$, para las restricciones de desigualdad activas, y $V^T h_j(x^*) = 0$, para las restricciones de igualdad, se cumple lo siguiente:

$$(6) \quad V^T \nabla L \quad (x^*, u^*, v^*) \quad V \geq 0$$

Las condiciones suficientes para que x^* sea un mínimo local aislado del problema (1.13) son las mismas (a), (b) y (,) del Teorema 3, excepto la parte (d) [ecuación (6)], la cual debe ser substituida por:

(d'). Para cada vector V no cero para el cual $V^T \nabla g_j(x^*) = 0$ para las restricciones de desigualdad activas, $V^T \nabla g_j(x^*) > 0$ para las restricciones de desigualdad no activas y $V^T \nabla h_j(x^*) = 0$ para las restricciones de desigualdad, lo siguiente es verdadero:

$$(6') \quad V^T \nabla^2 L \quad (x^*, u^*, v^*) \quad V \qquad 0$$

Ejemplo 1. Condiciones necesarias y suficientes con restricciones de desigualdad.

Minimizar: $f(x) = x_1^2 + x_2$

Sujeta a: $g_1(x) = -(x_1^2 + x_2^2) + 9 \geq 0$

$g_2(x) = -x_1 - x_2 + 1 \geq 0$

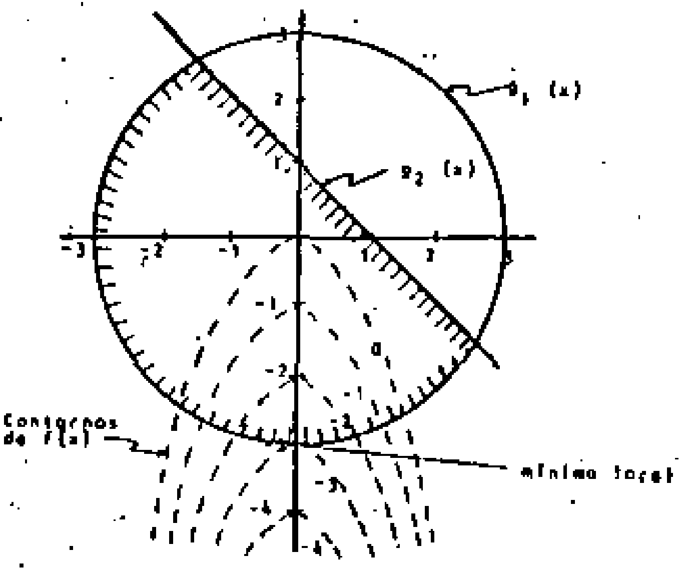


FIGURA 4. Región admisible y curvas de contorno del Ejemplo 1. Se observa que $g_1(x)$ es una restricción activa mientras que $g_2(x)$ no lo es.

Ya que sólo una de las restricciones está activa, no se necesita comprobar la calificación a primer - y segundo orden de las restricciones (Note que $f(x)$, $g_1(x)$ y $g_2(x)$ son dos veces diferenciables).

De acuerdo a los Teoremas (1) y (2) se necesita demostrar que existen u^* y x^* tales que

(2) $\quad g_j(x^*) \geq 0 \qquad\qquad -x_1^{*2} - x_2^{*2} + q \geq 0$

$$-x_1^* - x_1^* + 1 \geq 0$$

(3) $\quad u_j^*\, g_j(x^*) = 0 \qquad\qquad u_1^*(-x_1^{*2} - x_2^{*2} - q) = 0$

$$u_2^*(-x_1^* - x_2^* + 1) = 0$$

(4) $\quad u_j^* \geq 0. \qquad\qquad u_1^* \geq 0$

$$u_2^* \geq 0$$

(5) $\quad \nabla L(x^*, u^*) = 0 \qquad\qquad [L = f(x) - u_1 g_1(x) - u_2 g_2(x)]$

$$\begin{pmatrix} 2x_1^* \\ 1 \end{pmatrix} - u_1^* \begin{pmatrix} -2x_1^* \\ -2x_2^* \end{pmatrix} - u_2^* \begin{pmatrix} -1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Resolviendo las ecuaciones anteriores se puede verificar que

$$x^* = (0, -3)^T$$

$$\text{y} \qquad u^* = \left(\tfrac{1}{6}, 0\right)^T$$

La condición de segundo orden que debe ser satisfecha es :

$$v^T \nabla g_j(x^*) = 0 \qquad\qquad g_j(x^*) \;:\; \text{restricción activa}$$

es decir: $\quad (v_1, v_2)\begin{bmatrix} -2x_1^* \\ -2x_2^* \end{bmatrix} = v_1(0) + v_2(6) = 0$

de donde v_1 puede tomar cualquier valor, y $v_2 = 0$, substituyendo v en (6), se tiene

(6) $\quad v^T \nabla^2 L(x^*, u^*)\, v \geq 0$

$$\text{donde} \quad \nabla^2 L = \begin{pmatrix} 2(1+u_1) & 0 \\ 0 & 2u_2 \end{pmatrix}$$

__Ejemplo 5.__ Condiciones necesarias y suficientes con restricciones de Igualdad y de Desigualdad

Minimize: $\quad f(x) = x_1^2 + x_2$

Sujeta a: $\quad h_1(x) = x_1^2 + x_2^2 - 5 = 0$

$$g_2(x) = -(x_1 + x_2^2) + 1 \geq 0,$$

$$g_3(x) = -(x_1 + x_2) + 1 \geq 0.$$

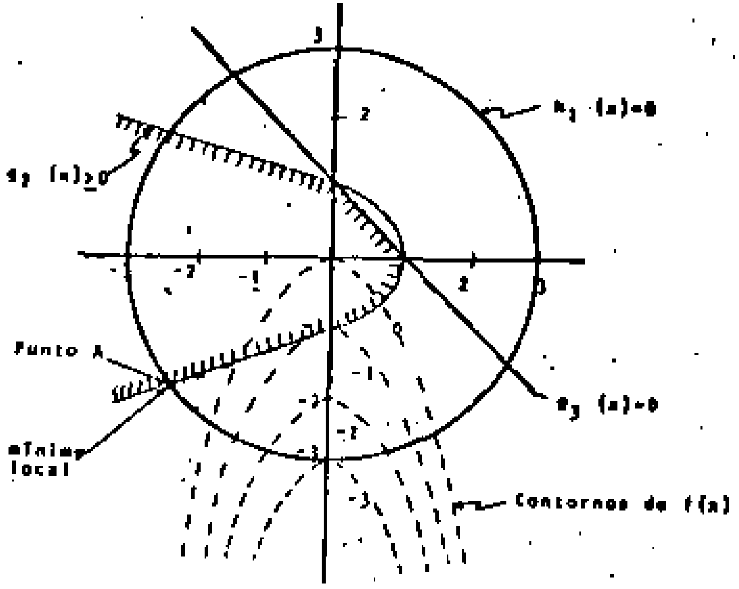


FIGURA 5. Región admisible y curvas de nivel del Ejemplo 2.

De acuerdo al Teorema 3. h_1, (x). $g_2(x)$ y $g_3(x)$ deben ser dos veces diferenciables. Además, los gradientes de las restricciones activas deberán ser linealmente independientes para que satisfagan la calificación a primer- y segundo orden. Suponga que A, localizado en $x^* = [-2.37, -1.84]^T$ en la intersección de $h(x^*) = g_2(x^*) = 0$ sea un candidato a mínimo local. Si se forma una combinación lineal entre los gradientes de $h_1(x)$ y $g_2(x)$, en x^*, se tiene que, para que sean

linealmente independientes

$$C_1 \nabla h_1(x^*) + C_2 \nabla g_2(x^*) = 0 \qquad C_1 \; {}^{y}/_{o} \; C_2 = 0$$

o

$$C_1 \begin{bmatrix} 2x_1^* \\ 2x_2^* \end{bmatrix} + C_2 \begin{bmatrix} -1 \\ -2x_2^* \end{bmatrix} = 0$$

y como

$$\det \begin{bmatrix} 2x_1^* & -1 \\ 2x_2^* & -2x_2^* \end{bmatrix} \neq 0 \qquad C_1 = C_2 = 0$$

es decir, los gradientes de las restricciones activas son linealmente independientes.

De acuerdo a los Teoremas (1) y (2) se debe cumplir que

(1) $h_1(x^*) = 0$ $\qquad x_1^{*2} + x_2^{*2} = 0$

(2) $g_j(x^*) = 0$ $\qquad -(x_1^* + x_2^{*2}) + 9 \geq 0$

$-(x_1^* + x_2^*) + 1 \geq 0$

(3) $u_j^* \, g_j(x^*) = 0$ $\qquad u_2^* \left[-(x_1^{*2} + x_2^{*2}) + 9 \right] = 0$

$u_3^* \left[-(x_1^* + x_2^*) + 1 \right] = 0$

(4) $u_j^* \geq 0$ $\qquad u_2^* \geq 0$

$u_3^* \geq 0$

$$(5)\ \nabla L(x^*, u^*, v^*) = 0 \qquad L = f(x^*) + v_1^* h_1(x^*) - u_2^* g_2(x^*) - u_3^* g_3(x^*)$$

$$\begin{bmatrix} 2x_1^* \\ 1 \end{bmatrix} + v_1^* \begin{bmatrix} 2x_1^* \\ 2x_2^* \end{bmatrix} - u_2^* \begin{bmatrix} -1 \\ -2x_2^* \end{bmatrix} - u_3^* \begin{bmatrix} -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

de donde se obtiene que :

$$v_1^* = -0{,}779$$

$$u_1^* = 1{,}05$$

$$u_3^* = 0$$

De acuerdo al Teorema 3, se debe encontrar V tal que $V^T \nabla g_j(x^*) = 0$ para las restricciones de desigualdad activas, y además $V^T \nabla h_1(x^*) = 0$, es decir

$$(v_1, v_2) \cdot \begin{bmatrix} -1 \\ -2x_2^* \end{bmatrix} = 0 \qquad V_1 + 2x_2^* V_2 = 0$$

$$v_1, v_2 \quad \begin{matrix} 2x_1^* \\ 2x_2^* \end{matrix} = 0 \qquad 2x_1^* V_1 + 2x_2^* V_2 = 0$$

de donde se obtiene que $V = (v_1, v_2)^T = [0, 0]^T$ y entonces existe una única solución al problema en la intersección de $g_2(x)$ y $h_1(x)$.

Note que en este problema en particular

$$V^T \nabla^2 L(x^*, u^*, v^*) V \geq 0$$

para toda $V \neq 0$, es decir $\nabla^2 L(x^*, u^*, v^*)$ es una matriz positiva definida. (condición (4')).

Los dos ejemplos analizados en este capítulo tienen el propósito de ilustrar las condiciones de optimalidad de primero y segundo orden. Ahora bien, los problemas que se pueden tratar analíticamente son demasiado sencillos, sin que esto quiera decir que para problemas muy grandes y/o complicados lo anterior no sea válido.

BIBLIOGRAFIA

1. A.V. Fiacco and G.P. Mc Cormick. "Non linear Programming", John Wiley and Sons, Inc., New York, 1968.

2. H.W. Kuhn and A. W. Tucker. "Non linear Programming. Proc. 2nd. Berkeley Symp. on Mathematical Statistics and Probability, Univ. of Calif. Press, Berkeley, 1951, pp. 481-492.

3. G.P. Mc. Cormick. SIAM J. Appl. Math., <u>15</u> : 641 (1967)

4. O.L. Mangasarian, "Non linear Programming", Mc. Graw-Hill, New York, 1969.

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO OPTIMO DE SISTEMAS DE INGENIERIA

ESTABILIZACION DEL METODO DE NEWTON PARA LA SOLUCION DE SISTEMAS DE ECUACIONES NO LINEALES

A V LEVY

A C SEGURA

MARZO, 1982

Resumen.

Un nuevo método general para resolver sistemas de ecuaciones no lineales de una manera eficiente, se ha obtenido. Este método tiene la característica de resolver aquellos problemas en los cuales la singularidad de la matriz Jacobiana se presenta. En estos problemas los métodos de Newton y sus variantes dejan de funcionar. El método está basado en la introducción de una nueva función, llamada función de tunelización, cuando se detecta la existencia de una singularidad. Esta función tiene la ventaja de preservar las soluciones de la función original y de eliminar la singularidad de la matriz Jacobiana.

La base teórica del presente método está íntimamente relacionado, por una parte, con la existencia de puntos singulares. Aunque todavía no existe una explicación teórica definitiva, es evidente de los resultados experimentales que la existencia de una o más singularidades no imposibilitan que el método tenga una alta probabilidad de convergencia y en algunos casos puede ser globalmente convergente. Por otra parte, en la ausencia de puntos singulares, resulta plausible esperar no solamente que el algoritmo sea globalmente convergente, sino que se vuelve idénticamente el Método de Restauración Modificado, Ref. [1]. Esta conjetura está basada en el comportamiento del método. Como un resultado particular de este método se obtiene el Método de Newton Estabilizado, de donde los resultados obtenidos para el primer método serán también válidos para el segundo método.

CONTENIDO.

1. Introducción.

En general los métodos de Newton, ref. [1], presentan malas propiedades de convergencia, puesto que divergen cuando la matriz Jacobiana se vuelve singular en algún punto nominal. A tal punto lo llamaremos punto singular o simplemente singularidad y lo denotaremos por x^*. En el presente trabajo se va a desarrollar un nuevo método para evitar la singularidad de la matriz Jacobiana, cuando ésta se presenta, lográndose la estabilización del método de Newton clásico.

Para eliminar una singularidad cuando se detecta su existencia, se transforma el problema original a otro problema equivalente, de modo que esta transformación preserve la solución del problema original. A esta transformación la llamaremos Función de Tunelización y se denotará por $T(x,k)$ donde k es un número real que la utilizaremos para eliminar la singularidad x^*. La construcción de la función de Tunelización es como sigue: Supongamos que se desea resolver un sistema de ecuaciones no lineales de la forma

$$\phi(x) = 0 \qquad (1)$$

donde ϕ es una función vectorial de dimensión q y x un vector de dimensión n con $q < n$. Supongamos que la ec. (1) tiene al menos una solución. Supongamos también que existe cuando menos un punto x^* en el cual $\phi_x(x^*) = 0$ y $\phi(x^*) \neq 0$.

Al utilizar el Algoritmo de Restauración Modificado, ref. [1], y el valor nominal x está cerca de x^*, entonces cuando $x \to x^*$, $\phi_x(x^*) \to 0$ produciéndose para $\alpha=1$, α el tamaño del paso, desplazamientos de gran magnitud y en general para satisfacer la propiedad de descenso del índice de comportamiento se necesitará utilizar α sumamente pequeña dando como resultado un avance muy lento del algoritmo y en el caso extremo cuando se llega a tener $\phi_x(x^*)=0$ ya no se logra avance alguno por no existir la inversa de $\phi_x(x^*)$. Considérese ahora un punto $x=x^*+\delta$ muy cercano a x^*, donde δ es un vector muy pequeño de dirección aleatoria. La función de tunelización se define como

$$T(x,k) = \frac{\phi(x)}{|(x-x^*)^T(x-x^*)|^k} \qquad (2)$$

El escalar $R = |(x-x^*)^T(x-x^*)|$ es el factor que nos ayudará a eliminar la singularidad x^* para un valor de k suficientemente grande. Es decir, si $\phi_x(x^*)=0$ entonces

$$T_x(x,k) = \frac{\phi_x(x)}{R^k} - \frac{2k}{R^{k+1}}|(x-x^*)\phi^T(x)| \qquad (3)$$

$$\longrightarrow -\frac{2k}{R^{k+1}}|(x-x^*)\phi^T(x)| \neq 0$$

Por lo tanto un nuevo desplazamiento puede obtenerse.

Como resultado se obtiene un Método de Restauración Estabilizado (MRE) cuya propiedad principal es su convergencia cuadrática cuando el punto nominal se elige muy cerca de la solución, permaneciendo en este caso k=0 constante, así como tener una mayor probabilidad de convergencia a la solución, que el método de Restauración sin estabilización. Cuando q=n este método se transforma en el Método de Cuasilinealización Estabilizado (MCE), donde los resultados obtenidos para el MRE serán válidos también para este Método.

1. Formulación del Problema.

Supóngase que se desea resolver un sistema de ecuaciones no lineales descrito por la ecuación

$$\phi(x) = 0 \qquad (4)$$

donde ϕ es una función vectorial de dimensión q y x un vector de dimensión n, con $q < n$. Supongamos que la primera derivada de $\phi(x)$ con respecto a x existe y es continua. Supóngase también que la ecuación tiene una solución.

Indice de Comportamiento. Como la ec. (4) es no lineal, está uno obligado a usar métodos aproximados para resolverla. En particular se usará cuasilinealización. Por lo tanto, es conveniente introducir la función escalar P(x) definida como

$$P(x) = \phi^T(x)\ \phi(x) \qquad (5)$$

llamado índice de comportamiento. La función P(x) mide el error cometido en las ecuaciones al usar métodos aproximados. Para un valor dado de x se tiene

$$P(x) = 0 \quad \text{si} \quad x=x^* \qquad (6)$$

$$P(x) > 0 \quad \text{si} \quad x\neq x^* \qquad (7)$$

donde x^* denota la solución exacta. Si se usa cuasilinealización,

se deben de obtener valor de x tales que

$$P(x) \leq \varepsilon \qquad (8)$$

donde ε es un número pequeño seleccionado de antemano y de acuerdo a la exactitud requerida en la solución.

3. **Detección de la Singularidad.**

Durante el desarrollo del Método de Restauración Modificado (MRM) observamos que la dificultad principal de este método es la presencia de la singularidad de la matriz Jacobiana en determinados problemas. En tales circunstancias al aplicar el ARM a tales problemas, el algoritmo diverge, porque al pedir que la propiedad de descenso se cumpla se necesitará tomar un valor de α sumamente pequeño sin lograr tal propósito. Por lo tanto, teóricamente hemos proporcionado una técnica para detectar la existencia de una singularidad. Desafortunadamente, en la práctica, este proceso es demasiado caro por requerir mucho tiempo de cómputo para llegar exactamente a la singularidad. Sin embargo, podemos dar una técnica de detección bastante aceptable que nos indique la existencia de una singularidad, sin llegar exactamente a ella. Esta técnica consiste en fijar un número máximo de bisecciones en α para satisfacer la propiedad de descenso del índice de comportamiento. Es decir, si Bisec denota el número de bisecciones hechas sobre α y B_{max} el número máximo de bisecciones permitidas sobre α, entonces

Entonces existe una singularidad si Bisec > B_{max} (9)

Entonces no existe una singularidad si Bisec ≤ B_{max} (10)

4. Eliminación de la Singularidad.

Una vez detectada la existencia de una singularidad x*el camino a seguir es tratar de eliminarla. Para ello reemplacemos a la función original por una función equivalente de tal manera que ésta función tenga la misma solución que la función original.

Función de Tunelización. Para eliminar la singularidad se propone la siguiente función,

$$T(x,k) = \frac{\phi(x)}{[(x-x^*)^T(x-x^*)]^k} \qquad (11)$$

llamada función de Tunelización. La función T(x,k) es una función vectorial no lineal de dimensión q, $x = x^* + \delta$ un punto cercano a x^* de dimensión n, δ un vector de dirección aleatoria y k un parámetro cuyo valor será determinado para eliminar la singularidad.

Selección del parámetro k. Supongamos que $\phi_x(x^*) = 0$. Derivando la ec. (11) con respecto a x, obtenemos

$$T_x(x,k) = \frac{1}{\rho^k}[\phi_x(x)] - \frac{2k}{\rho^{k+1}}[(x-x^*)\phi^T(x)] \qquad (12)$$

donde $T_x(x,k)$ es una matriz de dimensión n x q.

Si k=0, entonces las ecs. (11) y (12) se tranforman en

$$\begin{aligned} T(x,k) &= \phi(x) \\ T_x(x,k) &= \phi_x(x) \end{aligned} \qquad (13)$$

que coinciden con la formulación del MRM. Como $\phi_x(x^*) = 0$, se tiene $T_x(x,k) = 0$, produciéndose la divergencia del MRM para k=0.

Para poder calcular un nuevo desplazamiento es necesario que $T_x(x,k) \neq 0$. Ahora, incrementando k en incrementos $\Delta k > 0$, tal que $k = k + \Delta k$, se tiene

$$T_x(x,k) = \frac{1}{O(\delta^{2k})}[\phi_x(x^*)] - \frac{2k}{O(\delta^{2k+2})}[D(\delta)\,\phi^T(x)] \qquad (14)$$

Como $\phi_i(x) \neq 0$, $i=1,2,\ldots,q$, obtenemos

$$T_x(x,k) = -\frac{2k}{O(\delta^{2k+2})}[D(\delta)\phi^T(x)] \neq 0 \qquad (15)$$

que era lo que se pedía para poder calcular un nuevo desplazamiento, para un valor de k suficientemente grande.

En el análisis anterior hemos supuesto que

$$||x-x^*|| < \delta \qquad (16)$$

de modo que, para valores de k muy grande

$$\frac{1}{[(x-x^*)^T(x-x^*)]^k} \not\to 0 \qquad (17)$$

Por lo tanto, en la vecindad de un punto x^*, se tiene

$$|T_x(x,k)| \longrightarrow -\frac{2k}{\rho^{k+1}}\,|(x-x^*)\phi^T(x)| \neq 0 \qquad (18)$$

para k suficientemente grande.

Si la ec. (16) no se cumple entonces la singularidad de x^* se sustituye por el último punto x encontrado, generándose a continuación el vector $x=x^*+\delta$.

El procedimiento anterior para determinar el valor de k puede resumirse como sigue: Supóngase que para k=0 hemos descubierto una singularidad por el procedimiento de la sec. (3). Entonces

```
(i)   Calcular $T(x,k)$ y $T_x(x,k)$
(ii)  Si la ec. (16) se cumple pasar al paso (iv). En
      caso contrario pasar a (iii)
(iii) Efectuar la asignación
          $x_k^* \leftarrow x$
          $x = x^* + \delta$
          $k = 0$
      regresar a (i)
(iv)  Incrementar k, en incrementos $\Delta k > 0$. Regresar a(i).
```

5. Algoritmo de Restauración Estabilizado.

A continuación vamos a prestar el desarrollo del nuevo algoritmo para obtener el siguiente desplazamiento, después de haber detectado y eliminado una singularidad x^*. La característica de este algoritmo es una probabilidad de convergencia más grande que la del método de Restauración Modificado.

Índice de Comportamiento. Como la ec. (11) es también no lineal, está uno obligado a usar métodos aproximados para resolverla. Por lo tanto, es importante introducir el índice de comportamiento $Q(x,k)$, definido por

$$Q(x,k) = T^T(x,k)\ T(x,k) \tag{19}$$

que mide el error en las ecuaciones al usar métodos aproximados. Así dado cualquier punto nominal x se tiene

$$Q(x,k) = 0 \quad \text{si} \quad x=x^* \tag{20}$$

$$Q(x,k) > 0 \quad \text{si} \quad x\neq x^* \tag{21}$$

donde x* es la solución exacta. Si se usa cuasilinealización se deben de obtener valores de x tales que

$$Q(x,k) \leq \epsilon \tag{22}$$

donde ε es un número muy pequeño seleccionado de acuerdo a la exactitud requerida en la solución.

Como el problema original es resolver la ec. (4) y T(x,k) es una ecuación derivada de ésta, resulta natural considerar como criterio de convergencia del algoritmo la ec. (8), puesto que para $k > 0$ se tiene en general $P(x) < Q(x,k)$, por lo que, el criterio de convergencia (22) es reemplazado por la ec. (8).

Sea $x = x^* + \varepsilon$ un punto nominal dado. Considérese el desplazamiento Δx que nos lleve del punto nominal x a un punto variado $\tilde{x}$, tal que

$$\tilde{x} = x + \Delta x \tag{23}$$

Si se usa cuasilinealización, entonces la ec. (11) es aproximada por

$$\delta T(x,k) = -\alpha\, T(x,k) \qquad 0 < \alpha \le 1 \tag{24}$$

donde

$$\delta T_x(x,k) = T_x^T(x,k)\,\Delta x \tag{25}$$

denota la primera variación de T(x,k). $T_x(x,k)$ representa una matriz de dimensión n x q, cuya j-ésima columna es el gradiente de la función $T_j(x,k)$, $j=1,2,\ldots,q$, con respecto al vector x.

Sustituyendo la ec. (25) en la ec. (24) obtenemos

$$T_x^T(x,k)\,\Delta x = -\alpha\, T(x,k), \quad 0 < \alpha \le 1 \tag{26}$$

La ec. (26) representa un sistema lineal algebraico de q ecuaciones con n incógnitas. Si hacemos la hipótesis $q < n$, entonces la ec. (26) tiene una infinidad de soluciones Ref. [ ]. Sin embargo, se puede obtener una solución única si se dá la siguiente condición: "Una solución única de la ec. (26) se puede obtener si se busca un desplazamiento de longitud mínima en el sentido del principio de mínimos cuadrados".

Por consiguiente se desea resolver el siguiente problema:

$$\text{Min} \quad w = \frac{1}{2}\,\Delta x^T\,\Delta x \tag{27}$$

s.a.

$$T_x^T(x,k)\,\Delta x + \alpha\, T(x,k) = 0 \tag{28}$$

Por los métodos de la teoría de máximos y mínimos, se sabe que el problema anterior puede ser reformulado como el minimizar la función aumentada,

$$F(\Delta x,\lambda) = \frac{1}{2}\,\Delta x^T\Delta x + \lambda^T(T_x^T(x,k)\Delta x + \alpha T(x,k)) \tag{29}$$

donde $F(\Delta x,\lambda)$ es conocida como la función Lagrangiana y el vector λ de dimensión q como el multiplicador de Lagrange. Si

$$F_{\Delta x}(\Delta x,\lambda) = \Delta x + T_x(x,k)\,\lambda \tag{30}$$

denota el gradiente de la función aumentada $F(\Delta x,\lambda)$ con respecto

al vector Δx, el desplazamiento óptimo debe satisfacer

$$F_{\Delta x}(\Delta x, k) = 0 \qquad (31)$$

Esto produce la relación

$$\Delta x = - T_x(x,k)\ \lambda \qquad (32)$$

sustituyendo la ec. (32) en la ec. (26) se obtiene

$$A(x,k)\ \lambda = \quad \alpha\ T(x,k) \qquad (33)$$

donde A(x,k) es una matriz de dimensión q x q de la forma

$$A(x,k) = T_x^T(x,k)\ T_x(x,k) \qquad (34)$$

Para un valor dado de α la ec. (33) representa un sistema lineal de q ecuaciones con q incógnitas, y puede resolverse por eliminación gausiana para obtener el valor de λ.

<u>Cambio de Coordenadas</u>. Para simplificar el problema anterior introducimos la variable auxiliar

$$Z = \frac{\lambda}{\alpha} \qquad (35)$$

transformándose la ec. (33) en

$$A(x,k)\ Z = \quad T(x,k) \qquad (36)$$

que es equivalente a un sistema lineal algebraico en la incógnita Z.

Para obtener el valor de Z resolveremos la ec. (36) por eliminación gausiana, e inmediatamente después introducimos otra variable auxiliar <u>Y</u> definida por

$$Y = - T_x(x,k)\ Z \qquad (37)$$

donde Y es un vector de dirección. El valor del desplazamiento Δx puede obtenerse de la ecuación

$$\Delta x = - \alpha\ \underline{Y} \qquad (38)$$

El vector x̃ es calculado de la ec.(23). Si el punto x̃ satisface la ec. (8) entonces x̃ es la solución buscada y el algoritmo se termina. En caso contrario el punto x̃ se toma como punto nominal para la siguiente iteración. Como resultado obtenemos un Algoritmo de Restauración Estabilizado (ARE) y es usado iterativamente hasta lograr la convergencia a la solución buscada.

Si q=n, de la ec. (36) obtenemos

$$Z = A^{-1}(x,k)\ T(x,k) \qquad (39)$$

$$= T_x^{-1}(x,k)\ [\,T_x^T(x,k)]^{-1}\ T(x,k)$$

Sustituyendo la ec. (39) en la ec. (37) obtenemos

$$Y = -[T_x^T(x,k)]^{-1} T(x,k) \qquad (40)$$

Multiplicando por la izquierda la ec. (40) por $T_x^T(x,k)$, obtenemos

$$T_x^T(x,k)\,\underline{Y} = -T(x,k) \qquad (41)$$

La ecuación (41) es equivalente a un sistema lineal de n ecuaciones con n incógnitas y puede resolverse por eliminación gaussiana para obtener el valor de $\underline{Y}$. Una vez conocido el vector $\underline{Y}$ calcular Δx y $\tilde{x}$ de las ecuaciones (38) y [23] respectivamente. Si $\tilde{x}$ satisface la ec. (8) entonces $\tilde{x}$ es la solución buscada y el algoritmo si termina. En caso contrario, se toma a $\tilde{x}$ como punto nominal y el proceso vuelve a repetirse. Como resultado obtenemos el Algoritmo de Cuasilinealización Estabilizado (ACE), o algoritmo de Newton Estabilizado.

<u>Propiedad de Descenso del Indice de Comportamiento</u>. Para impedir que el índice de comportamiento $Q(x,k)$ aumente cuando pasamos del punto nominal x al punto variado $\tilde{x}$, requerimos que su primera variación sea negativo. La primera variación de $Q(x,k)$ está dado por

$$\delta Q(x,k) = 2\,T^T(x,k)\,\delta T(x,k) \qquad (42)$$

De la ec. (24) se tiene

$$\delta Q(x,k) = -2\,\alpha\,Q(x,k) \qquad (43)$$

Ahora, como $Q(x,k) > 0$, ya que x es un punto que no satisface la ec. (11), y para $\alpha > 0$, tenemos

$$\delta Q(x,k) < 0 \qquad (44)$$

Por lo tanto, si α es muy pequeño el descenso del índice de comportamiento está garantizado, es decir

$$\tilde{Q}(\tilde{x},k) < Q(x,k) \qquad (45)$$

Las ecs. (44) - (45) constituyen lo que llamamos la propiedad de descenso del índice de comportamiento. Para determinar el valor óptimo de α para el cuál la propiedad de descenso se cumple, se efectuará un proceso de bisección sobre α , ref. [1]

6. Resumen de los algoritmos.

Las diferentes etapas del desarrollo de los algoritmos, ACE y ARE pueden resumirse de la siguiente manera:

a) Algoritmo de Cuasilinealización Estabilizado.

1. Dar como punto nominal a x^*.
2. Generar el punto $x = x^* + \delta$. Asignar $k=0$.
3. Calcular $\phi(x)$, $\phi_x(x)$ y $P(x)$.
4. Si $P(x)$ satisface la ec. (8), entonces el algoritmo se termina. En caso contrario pasar al siguiente paso.
5. Calcular $T(x,k)$, $T_x(x,k)$ y $Q(x,k)$.
6. Calcular Y de la ec. (41). Asignar $\alpha=1$.
7. Calcular Δx y $\tilde{x}$ de las ecs. (38) y (23), respectivamente.
8. Calcular $\phi(\tilde{x})$, $T(\tilde{x},k)$ y $\tilde{Q}(\tilde{x},k)$.
9. Si $\tilde{Q}(\tilde{x},k) < Q(x,k)$ pasar al paso (13). En caso contrario pasar al siguiente paso.
10. Si la ec. (9) se cumple pasar al paso (11). En caso contrario el valor de α se reemplaza por $\alpha/2$ y se regresa a (7).
11. Si la ec. (16) se cumple pasar a (12). En caso contrario efectuar las asignaciones $x^* = x$, $x = x^* + \delta$ y $k=0$. Regresar a (3).
12. Incrementar k en incrementos $\Delta k > 0$. Regresar a (5).
13. Una vez conocido $\tilde{x}$ la iteración se termina. El punto $\tilde{x}$ se toma como punto nominal para la siguiente iteración. Regresar a (1).

Algoritmo de Restauración Estabilizado:

1. Dar como un punto nominal a x^*. Generar el punto $x = x^* + \delta$. Asignar $k=0$.
2. Calcular $\phi(x)$, $\phi_x(x)$ y $P(x)$.
3. Si $P(x)$ satisface la ec. (8) el algoritmo se termina. En caso contrario pasar al siguiente paso.
4. Calcular $T(x,k)$, $T_x(x,k)$ y $Q(x,k)$.
5. Calcular Z y Y de las ecs. (36) y (37) respectivamente. Asignar $\alpha=1$.
6. Calcular Δx y $\tilde{x}$ de las ecuaciones (38) y (23) respectivamente.
7. Calcular $\phi(\tilde{x})$, $T(\tilde{x},k)$ y $\tilde{Q}(\tilde{x},k)$.
8. Si $\tilde{Q}(\tilde{x},k) < Q(x,k)$ pasar al paso (12). En caso contrario pasar al siguiente paso.
9. Si la ec. (9) se cumple pasar al paso (10). En caso contrario el valor de α es reemplazado por $\alpha/2$. Regresar a (6).
10. Si la ec. (16) se cumple pasar al paso (11). En caso contrario efectuar las asignaciones $x^* = x$, $x = x^* + \delta$ y $k=0$. Regresar a (2).
11. Incrementar k en incrementos $\Delta k > 0$. Regresar a (4).
12. Con $\tilde{x}$ conocido la iteración se termina. Tomar a $\tilde{x}$ como punto nominal para la siguiente iteración, regresando a (2).

8. Conclusiones.

Un método general para resolver sistemas de ecuaciones no lineales de la forma $\phi(x) = 0$, donde $\phi(x)$ es una función vectorial de dimensión q y x un vector de dimensión n con $q \leq n$, se ha desarrollado. Este método tiene la característica de resolver aquellos problemas en los cuales la matriz $\phi_x(x^*)=0$ se presente, identificando a x^*como un punto singular. La eliminación del punto singular x^*se logra al introducir la función de tunelización $T(x,k)$ para un valor de k suficientemente grande. El método está basado en la consideración de los índices de comportamiento $P(x)=\phi^T(x)\ \phi(x)$ y $Q(x,k) = T^T(x,k)\ T(x,k)$, donde el primero es usado como criterio de convergencia y el segundo como una guía durante el desarrollo del método, ya que en, general, para $k > 0$ se tiene $Q(x,k) > P(x)$.

Un algoritmo de Restauración Estabilizado ha sido generado al considerar la existencia y eliminación de puntos singulares y al mismo tiempo requerir que la primera variación del índice de comportamiento $Q(x,k)$ sea negativo.

Si k=0 el ARE se transforma idénticamente en el Algoritmo de Restauración Modificado (ARM), el cúal no tiene la propiedad de detectar y eliminar puntos singulares. Esto significa que si el ARE es usado con k=0 entonces el algoritmo no puede (en general) converger a la solución.

La propiedad fundamental del ARE, la capacidad de poder detectar la existencia de puntos singulares y la capacidad de poder eliminarlos, hace que el algoritmo sea más confiable puesto que, en algunos casos puede volverse globalmente convergente.

Finalmente, cuando q=n el ARE se transforma en el Algoritmo la Cuasilinealización Estabilizado y los resultados obtenidos para el primero son también válidos para el segundo.

8. Condiciones Experimentales.

Con el objeto de comparar las ventajas logradas con los algoritmos ACE y ARE sobre los algoritmos ACD, ACM y ARD y ARM, respectivamente, se resolvieron varios ejemplos numéricos usando una computadora B6700 con aritmética de doble precisión. Todos los algoritmos fueron programados en Fortran IV.

Valores nominales. Para todos los algoritmos los valores nominales fueron elegidos como

$$x_j = x_i^* + \beta\, \delta_{ij} \quad i=1,\, 2, \ldots, n;\ j=1, 2, \ldots, 2n \qquad (46)$$

usando los siguientes valores de β y de δ_{ij}

$$\beta = 0,\ \pm 1,\ \pm 2, \ldots, \pm 50 \qquad (47)$$

$$\delta_{ij} = \begin{matrix} 1 & \text{si} & i = j \\ 0 & \text{si} & i \neq j \end{matrix} \qquad (48)$$

En la ec. (46), x^* representa la solución exacta, β es la distancia del punto nominal x a la solución exacta x^* y δ_{ij} es la dirección tomada sobre los ejes coordenados.

Condición de Paro. El criterio de convergencia para obtener la solución deseada, se escogió para todos los algoritmos como

$$F(x) \leq 10^{-20} \qquad (49)$$

y los criterios de no convergencia como

$$N \geq 100 \qquad (50.1)$$

$$N_b \geq 20 \qquad (50.2)$$

$$k_m \geq 10 \qquad (50.3)$$

donde N es el número máximo de iteraciones del algoritmo para lograr la convergencia a la solución deseada, N_b es el número máximo de bisecciones permitidas en el tamaño del paso α_{ki} para satisfacer la propiedad de descenso de cada algoritmo y k_m el valor máximo permitido del parámetro k de la ec. (11).

La condición (50.1) implica la convergencia muy lenta del algoritmo, la condición (50.2) indica un valor extremadamente pequeño del desplazamiento y por consecuencia la convergencia muy lenta, y la condición (50.3) indica que los algoritmos ACE y ARE no pudieron cancelar la singularidad.

El valor de Δk es 0.1 y el valor de k_{max} en las ecs. (9-10) es 5 y 15 para el ACE y el ARE, respectivamente. El valor de δ es 10^{-3}.

Porcentaje de Exito. Puesto que cada problema fue resuelto varias veces (100 en total) comenzando con los valores nominales dados por las ecuaciones (46)-(48), sean N_r el número total de corridas efectuadas para un cierto problema dado usando un algoritmo dado, y N_s el número total de corridas exitosas. El porcentaje de éxito está definido por

$$P = \frac{N_s}{N_r}$$

de donde $0 \le p \le 1$ y mientras más grande sea el valor de p más poderoso es el algoritmo.

<u>Radio de Convergencia</u>. Para hacer una comparación de la eficiencia de cada algoritmo, definimos el radio de convergencia de la siguiente manera: Dado un punto nominal x la solución conocida x^*, medimos la distancia de x a x^* como $R=||x^* - x^o|| \le B\,\delta_{ij}$. Si para todos los nominales se tiene $p=1$, decimos entonces que $R(R=B)$ es el radio de convergencia de un algoritmo dado para un problema dado.

<u>Tiempo de Computo Empleado</u>. Si T_i representa el tiempo en segundos de CPU emplado en la i-ésima corrida exitosa, para resolver un problema dado usando un algoritmo dado, definimos el tiempo promedio en segundos de CPU por corrida exitosa como

$$T_{av} = \frac{1}{N_s} \sum_{i=1}^{N_s} T_i$$

5. <u>Ejemplos numéricos</u>.

En esta sección se describen 13 ejemplos númeres. Por simplicidad se usará notación escalar.

Ejemplo 1.1 Considérese la ecuación no lineal

$$x - \mathrm{sen}\ )\ x + 1$$

Con solución conocida $x = -1.04$

Ejemplo 1.2 Considérese la ecuación no lineal

$$x - \mathrm{sen}(2x) = 0$$

con soluciones conocidas de $x=0$, ± 0.95

Ejemplo 1.3 Considérese la ecuación no lineal

$$\mathrm{sen}\ (x) - x + 2 = 0$$

con solución conocida $x = 2.56$

Ejemplo 1.4 Considérese el sistema no lineal

$$e^{\mathrm{sen}\ (x^3)} + x - 1 = 0$$

$$2x^2 + 3y^2 - 4xy + 8(y-x) - 1 = 0$$

con solución conocida $(x,y) = (0,0.12)$. Otras soluciones son por ejemplo $(x,y) = (0,-2.79)$, $(-1.65, -2.23)$.

Ejemplo 1.5 Considérese el sistema no lineal [25]

$$x^2 + x - y^2 - 1 = 0$$
$$y - \mathrm{sen}\ (x^2) = 0$$

con solución conocida $(x,y) = (0.73, 0.5)$. Admite también la

solución $(x,y) = (-1.67,\ 0.35)$.

Ejemplo 1.6 Considérese el sistema no lineal

$$0.05\ \mathrm{sen}(4\pi y) - x - 2y + 1 = 0$$

$$y - 0.5\ \mathrm{sen}\ (2\pi x) = 0$$

con solución $(x,y) = (1,0)$. Admite también la solución $(x,y) =$ $(1.6,\ -0.29)$, $(0.4,\ 0.35)$, $(1.05,\ -0.4)$, $(0.15;\ 0.4)$.

Ejemplo 1.7 Considérese el sistema no lineal

$$2\ \mathrm{sen}\ (\pi x)\ \mathrm{sen}\ (2\pi z) - y + 1 = 0$$
$$0.1\ y\ \mathrm{sen}\ (2\pi z) - 1.5\ x - z + 2.5 = 0$$
$$0.1\ y\ \mathrm{sen}\ (2\pi x) - z + 1 = 0$$

con solución conocida $(x,y,z) = (1,1,1)$

Ejemplo 1.8 Considérese el sistema no lineal

$$2\ \mathrm{sen}(0.4\pi x)\ \mathrm{sen}(0.4\pi z) - y = 0$$

$$0.1z\ \mathrm{sen}(2\pi z) - x - z + 2.5 = 0$$

$$0.1y\mathrm{sen}\ (2\pi x) - z + 1 = 0$$

con solución conocida $(x,y,z) = (1.4,1,0,1)$.

Ejemplo 1. 9 Considérese el sistema no lineal

$$2\mathrm{sen}(0.4\pi x)\ \mathrm{sen}\ (0.4\pi x) - y = 0$$
$$0.1\ y\ \mathrm{sen}(2\pi z) - x - z + 2.5 = 0$$
$$0.1y + \mathrm{sen}\ (2\pi x) - z + 1 = 0$$

con solución $(x,y,z) = (1.05,1.85,1.47)$. Admite también las soluciones $(x,y,z) = (1.91,0.87,056)$, $(1.56,1.55,0.79)$.

Ejemplo 1.10 Considérese el sistema no lineal [2]

$$(x-z)^2 + (y-u)^2 + (x+y+z+u)^2 - 16 = 0$$
$$x\mathrm{sen}(0.5\pi z) + y\cos(0.5\pi u) - 1 = 0$$
$$x + y^2 + z^3 + u^4 - 4 = 0$$
$$x + 2y + 3z + 4u - 10 = 0$$

con solución conocida $(x,y,z,u) = (1,1,1,1)$.

Ejemplo 1. 11 Considérese el sistema no lineal [2]

$$(x-y)^2 + (y-z)^2 + (2z+u-v)^2 = 0$$
$$x^2 + y^2 + z^2 + u^2 + v^2 - 5 = 0,$$
$$(x-1)^2 + (y-2)^2 + v^4 - 2 = 0$$
$$x + 2y^2 + 3z^3 + 4u^4 + 5v^5 - 15 = 0$$
$$x^2 + xyz - u^3 - 1 = 0$$

con solución conocida $(x,y,z,u,v) = (1,1,1,1,1)$.

Ejemplo II. 1 Considérese la ecuación no lineal

$$e^{\mathrm{sen}(x^3)} + x + (y-1)^2 - 1 = 0.$$

umerical Experiments Stabilized Newton

Fortran IV . B- 6700 . Doble precision

Convergence

$P(x) < 10^{-20}$

Non-convergence

$N \geq 100$, $N_b \geq 20$, $\lambda \geq 10$ $(D):=0.1$

Each problem solved with 100 nominal values

$$x_0^{i} = x^{i*} + \beta \, \rho_{ij} \qquad i=1,2,\ldots n, \quad j=1,2 \ldots 2n$$

$$\beta = 0, \pm 1, \pm 2 \pm 3, \ldots \pm 50$$

Probability of Success

$$p = \frac{N_s}{100}$$

N_s : N° of succesful runs

Average Computing Time

$$T_{av} = \frac{1}{N_s} \sum_{i=1}^{N_s} T_i$$

T_i : CPU Time in seconds

con solución conocida $(x,y) = (0,1)$. Además, admite también las soluciones $(x,y) = (0.26, 1.53)$ y $(0.26, 0.4683)$

Ejemplo II.2 Considérese la ecuación no lineal

$$e^{\operatorname{sen}(x^3)} + x - (y-1)^2 - (z-2)^2 - 1 = 0$$

con solución $(x,y,z) = (0,1,2)$. Además tiene más soluciones, por ejemplo $(x,y,z) = (0,26,1,53,2)$, $(0.26,1,2.53)$, $(0,26,0.47,2)$.

Overall Probability of Success.

$$\tilde{p} = \frac{1}{N^0_{ex}} \sum_{i=1}^{N^0_{ex}} p_i$$

Overall Average Computing Time

$$\tilde{T}_{av} = \frac{1}{N^0_{ex}} \sum_{i=1}^{N^0_{ex}} T_i$$

	$\tilde{p}$	$\tilde{T}_{av}$
Standard Newton	0.49	0.441
Damped Newton	0.67	0.370
"Stretched" Newton	0.86	0.993

Numerical Results: $N_{ex} = 500$

Ex. No	Newton		Damped Newton		Stabilized Newton	
	p	T_{av}	p	T_{av}	p	T_{av}
1	0.78	1.29	0.72	0.18	1.00	0.48
2	0.67	0.37	0.98	0.10	1.00	0.22
3	0.81	0.47	1.00	0.08	1.00	0.16
4	0.56	0.41	0.53	0.69	0.81	6.60
5	0.42	1.46	0.79	0.27	1.00	0.26
6	0.67	0.41	0.94	0.28	1.00	0.56
7	0.98	0.92	0.89	0.23	1.00	1.08
8	0.44	0.07	0.78	0.33	0.99	0.84
9	0.49	0.29	0.50	0.15	0.96	1.92
10	0.41	3.97	0.24	0.92	0.63	3.80
11	0.76	10.20	0.21	0.96	0.97	7.47

BIBLIOGRAFIA.

1. A. MIELE, S. NAQUI, A.V. LEVY, R.R. IYER. Numerical Solution of Non-linear Equations and Non-linear Two-Point Boundary Value Problems. Department of Mechanical and Engineering and Materials Science. Rice University, Houston, Texas.

2. A. MIELE, H. Y. HUANG and J.C. HEIDEMAN. Sequential Gradient-Restoration Algorithm for the Minimization of Constrained Function. Ordinary and Conjugate Gradient Versions. JOTA, Vol. 4, No. 4, 1969.

3. K. M. BROWN and J. E. DENNIS. On Newton-Like Iterations Function: General Convergence Theorems and a Specific Algorithms. Numerische Mathematicke, 12, 1968, Pags. 186-191.

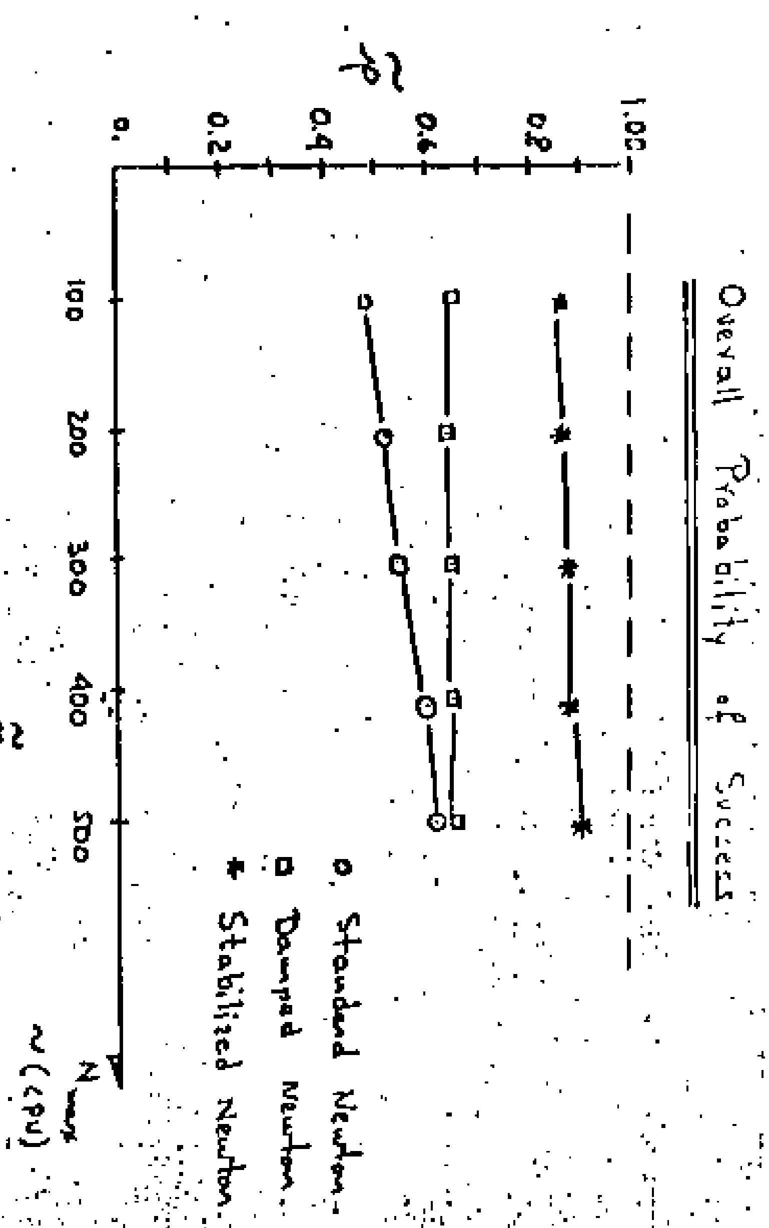
4. J. E. DENNIS. On Newton-Like Methods. Numerische Mathematike, 11, 1968, Pags. 324-330.

5. J. E. DENNIS. On the Kantorovich Hypothesis for Newton Method. SIAM J. on Numerical Analysis, 6, 1969, Pags. 495-507.

6. J. E. DENNIS. On the Newton Method and Nonlinear Synultaneous Displacements. SIAM J. on Numerical Analysis, Serv. B., Vol. 4, No. 1, 1967.

7. L. V. KANTOROVICH and G. P. AKILOV. Functional Analysis in Normed Spaces. N. Y. Pergammon Press, 1964.

8. M. M. VAINVERG. Varational Methods for the Study of Non linear Operator. Holden Day, Inc., San Francisco, 1964.



	$\tilde{p}$
Standard Newton	0.64
Damped Newton	0.68
Stabilized Newton	0.91

Ref. A.V. LEVY & A. SEGURA
DUNDEE (1979)

9. J. TODD. A Survey of Numerical Analysis. McGraw-Hill Book Company, New York.

10. J. S. LEE. Quasilinearization and Invariant Imbedding. Academic Press, New York, 1968.

11. D.G.LUENBERGER. Optimization by Vector Spaces Methods. John Wiley, New York, 1969, Pags. 277-281.

12. E. K. BLUM. Numerical Analysis and Computational Theory and Practice. Reading, Mass., Addison-Wesley, 1972, pags. 178-183.

13. B. NOBLE. Applied Linear Algebra. Prentice-Hall, Inc. 1969, pags. 77.

14. G. E. FORSYTHE, and C. B. MOLER. Computer Solutions of Linear Algebraic Systems. Prentice-Hall, Inc. Englewood Cliffs, N. J., 1967.

15. J. M. ORTEGA, and W. C. RHEINBOLDT. Iterative Solution of non Linear Equations in Several Variables. Academic Press, Inc., New York, 1970.

16. A. M. OSTROWSKI. Solution of Equation and Systems of Equations. Prentice-Hall, Inc. Englewood Cliffs, N. J., 1966.

17. J. F. TRAUB. Iterative Methods for the Solution of Equations. Prentice-Hall, Inc., Englewood Cliffs, N. J., 1964.

18. E. ISAACSON and H. B. KELLER. Analysis of Numerical Methods. John Wiley, New York, 1966.

19. P. J. ZELEZNIK. Quasi-Newton Methods for Nonlinear Equations. J. ACM. Vol. 15, No. 2, 1968.

20. F. FREUDENSTEIN and B. ROTH. Numerical Solution of Systems of Non linear Equations. J. ACM, Vol. 10, No. 4, 1963.

21. K. M. BROWN and W. GEARHART. Deflation Techniques for the Calculation of Further Solution of a Nonlinear Systems. Numerische Mathematike, 16, 1971.

22. K. M. BROWN. Solution of Simultaneous Non-linear Equations. C. ACM. Vol. 10, No. 11, 1967.

23. F. H. BRANIN. Widely Convergent Method for Find Multiple Solution of Symultaneous Non linear Equations. IBM J. Res. Develop, 1972.

24. R. P. BRENT. On the Davidenko-Branin Method for Solving Symultaneous Non linear Equations. IBM J. Res. Develop., 1972.

25. C. F. GERALD. Applied Numerical Analysis- Addison Wesley Reading Mass., 1973.

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO OPTIMO DE SISTEMAS DE INGENIERIA

ESTUDIO DE CASOS:

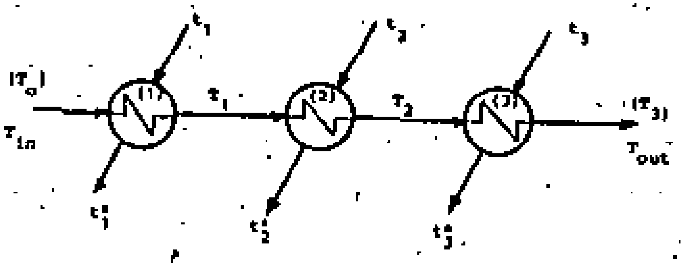
- Optimación de un tren de cambiadores de calor
- Ampliación de una planta química

DR. ANTONIO MONTALVO ROBLES

MARZO, 1982

OPTIMIZACION DE UN TREN DE CAMBIADORES DE CALOR

Se desea calentar un fluido desde una temperatura T_{in} hasta una temperatura de salida T_{out}, mediante intercambio de calor con tres corrientes liquidas calientes, en un sistema de tres cambiadores de calor que operan en contra corriente, según se muestra en la figura



Para el proceso anterior se especifican los siguientes parámetros :

WC_q : producto del flujo en masa por la capacidad calorifica (se supone idéntica en todas las corrientes)

t_i : temperatura de entrada de las corrientes calientes (i = 1, 2, 3)

U_i : coeficiente total de transferencia de calor en cada uno de los tres cambiadores (i = 1, 2, 3)

Si se supone que la inversión total requerida para el sistema de cambiadores es proporcional al area total de los mismos, el problema se puede plantear como el de seleccionar las areas A_i $(i=1,2,3)$ de manera tal que :

$$A_T = A_1 + A_2 + A_3$$

sea minima.

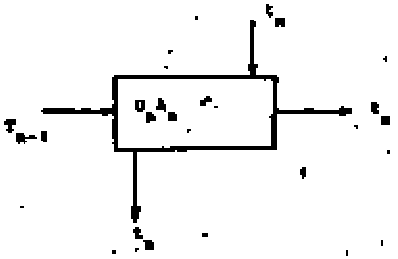
1°) Formular el problema como una optimización en estado estacionario. Identifique la función objetivo, variables de decisión y restricciones. Los siguientes parámetros se consideran fijos

Tin = To = 100	t_1 = 300	U_1 = 120
Tout = T_3 = 500	t_2 = 400	U_2 = 80
wc = 100,000	t_3 = 600	U_3 = 40

2°) Suponga que en el problema anterior las corrientes de salida caliente de los cambiadores 1 y 2 deben mezclarse y la temperatura resultante no deberá exceder los 230° . Plantee este nuevo problema tomando en cuenta la restricción planteada.

Modelo del cambiador de calor

Considérese el cambiador de calor a contra corriente que se muestra a continuación



Las ecuaciones que describen su funcionamiento son :

Q_n : rapidez de transferencia de calor en el n-ésimo cambiador

$$= WC\ (T_n - T_{n-1})$$

$$= WC\ (t_n - t_n^*)$$

$$= U_n A_n\ (t_n - T_n)$$

$$= U_n A_n\ (t_n^* - T_{n-1})$$

siendo las temperaturas de salida las siguientes :

$$T_n = \frac{T_{n-1} + a_n \phi_n}{1 + a_n}$$

donde $\phi n = \dfrac{U_n A_n}{WC}$

y $t_n^* = t_n - (T_n - T_{n-1})$

Función objetivo

Según se mencionó con anterioridad, se trata de minimizar el area total del sistema

$$A_T = A_1 + A_2 + A_3$$

Variables de decisión y restricciones

1- Sin restricciones en el mezclado (Parte 1)

a) T_1 y T_2 como variables de decisión

$Q_1 = W C\ (T_1 - T_o)$ $\qquad$ $A_1 = Q_1/U_1\ (\phi_1 - T_1)$

$Q_2 = W C\ (T_2 - T_1)$ $\qquad$ $A_2 = Q_2/U_2\ (\phi_2 - T_2)$

$Q_3 = W C\ (T_3 - T_2)$ $\qquad$ $A_3 = Q_3/U_3\ (\phi_3 - T_3)$

Restricciones sobre las variables independientes

$$T_o \leq T_1 \leq t_1$$

$$T_1 \leq T_2 \leq t_2$$

Restricciones sobre las variables dependientes

$$Q_i \geq 0 \qquad i = 1, 2, 3$$

o equivalentemente

$$A_i \geq 0 \qquad i = 1, 2, 3$$

b) Areas A_1 y A_2 como variables independientes

$$T_1 = \frac{T_o + a_1\ t_1}{1 + a_1} \quad ; \qquad a_1 = U_1\ A_1/W\ C$$

$$T_2 = \frac{T_1 + a_2\ t_2}{1 + a_2} \quad ; \qquad a_2 = U_2\ A_2/W\ C$$

Q_1, Q_2 y Q_3 como en el caso anterior, al igual que A_3

restricciones en las variables independientes

$$0 \leq A_1 \leq A_1^* \ \text{(dato)}$$

$$0 \leq A_2 \leq A_2^* \quad \text{(dato)}$$

restricciones en las variables dependientes : no hay.

c) Cargas térmicas Q_1 y Q_2 como variables de decisión

$T_1 = Q_1 / W C + T_o$; $A_1 = Q_1 / U_1 (t_1 - T_1)$

$T_2 = Q_2 / W C + T_1$; $A_2 = Q_2 / U_2 (t_2 - T_2)$

$Q_3 = W C (T_3 - T_2)$; $A_3 = Q_3 / U_3 (t_3 - T_3)$

Restricciones sobre las variables independientes

$$0 \leq Q_1 \leq W C (t_2 - T_o)$$

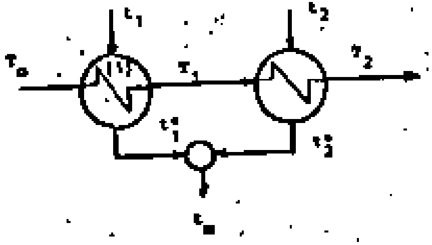
$$0 \leq Q_2 \leq W C (t_2 - T_o)$$

Restricciones sobre las variables dependientes

$$0 \leq Q_1 + Q_2 \leq W C (t_2 - T_o)$$

2- Con restricciones en la corriente de mezcla (Parte).

Las ecuaciones y restricciones analizadas en la parte 1 siguen siendo válidas aunque en este caso existe la necesidad de añadir otra restricción, según se analiza a continuación



ariable dependiente : $t_m^* = \frac{t_1^* + t_2^*}{2}$

restricciones sobre t_m :

1 Si $A_1 = \infty$ y $A_2 = \infty$ $\Longrightarrow$ $t_1^* = T_o$, $T_1 = t_1$

$t_2^* = T_1$, $T_2 = t_2$

entonces $\frac{t_1 + T_o}{2} \leq t_m^*$

y por lo tanto $\frac{t_1 + T_o}{2}$ t_m^* 230 (restricción del problema)

$\uparrow$ (restricción por area infinita en A_1 y A_2)

Las soluciones óptimas resultan ser :

Primera parte Sin restricción en t_m^*

$T_1 = 186.2$, $T_2 = 292.7$, $Q_1 = 8.62 * 10^6$, $Q_2 = 10.64 * 10^6$, $Q_3 = 20.73 * 10^6$

$t_m^* = 631.8$, $A_2 = 1239.1$, $A_3 = 5103.8$, $A_T = 7054.8$

Segunda parte Con restricción en t_m^*

$T_1 = 210.0$, $T_2 = 340.1$, $Q_1 = 11.00 * 10^6$, $Q_2 = 13.01 * 10^6$, $Q_3 = 16.01 * 10^6$

$t_m^* = 229.9$, $A_1 = 1017.9$, $A_2 = 2715.7$, $A_3 = 4009.3$, $A_T = 7743.0$

AMPLIACION DE UNA PLANTA QUIMICA

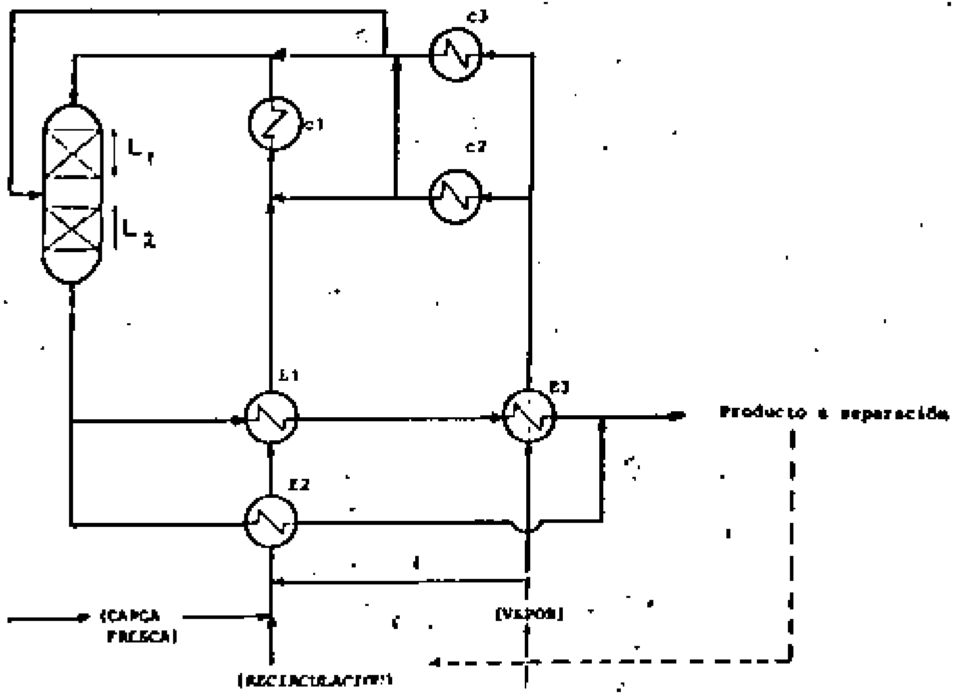
El presente estudio tiene el propósito de diseñar un nuevo reactor catalítico y fijar nuevas condiciones de operación para aumentar la capacidad de una planta que produce ES.

La plantá fue originalmente diseñada para producir 91 T/D de ES y se pretende aumentar la capacidad para producir 150 T/D. Dentro de las varias alternativas analizadas se pretende efectuar la optimización sobre el diagrama que se muestra a continuación :

C1, C2, C3 : calentadores

E1, E2, E3 : cambiadores

R : reactor catalítico



Bases de diseño

1. Composición de la carga fresca (%)

B	0.43
T	0.86
E B	98.46 ← (materia prima)
P E B	0.22

2. Composición de la carga total al reactor (sin vapor)

B	0.166
T	1.866
E B	95.217
E S	2.668 ← (producto final)
P E B	0.082

Descripción del Flujo

El EB fresco se une con la corriente de recirculación proveniente de otra sección de la planta. La corriente resultante se bombea al sistema de precalentamiento de carga, constituido por los cambiadores E1 y E2 ; antes de entrar a estos cambiadores la carga combinada se mezcla con aproximadamente el 9% del vapor total usado en la reacción. La mezcla, parcialmente vaporizada, se alimenta al cambiador E2 a una temperatura de 316° F , saliendo del mismo a 692° F , completamente vaporizada. Del cambiador E2 pasa al E1 de donde sale a 1092° F . El calentamiento final de la mezcla vapor-

hidrocarburos se suministra en el calentador C1, previa adición de vapor adicional proveniente del cambiador E3 y del calentador C2, de donde sale a una temperatura de 1150° F.

El 91% restante del vapor requerido por el proceso se precalienta en el cambiador E3. Este vapor entra a 366° F y sale del cambiador E3 ,a 748° F . El vapor así calentado, parte se alimenta al calentador C3 , de donde sale a 1300° F y el resto se alimenta al C2 para más tarde mezclarse con la corriente de vapor-hidrocarburos antes de entrar al calentador C1. Por otro lado, el vapor que sale del calentador C3 se divide en dos corrientes, a saber: una parte sirve para dar la temperatura final y la relación (vapor / hidrocarburos) a la entrada del reactor R, mientras que la otra se usa para elevar la temperatura de los gases de reacción entre los hechos catalíticos del reactor.

La mezcla vapor - hidrocarburos que sale del reactor R a 1119° F se aprovecha para precalentar los hidrocarburos y el vapor que entran al proceso.

(Nota: La mayoría de los datos dados con anterioridad son resultado de la optimización que se describe más adelante).

Datos

Carga fresca : cantidad máxima disponible
composición y temperatura

Recirculación : Cantidad y composición y temperatura

Calentador C3 : temperatura de salida

Vapor : temperatura de entrada

EB en producto a separación : cantidad (140 T/D)

Se requiere calcular lo siguiente :

1.. Calcular el volúmen de catalizador en el lecho L1

2.. Calcular el volúmen de catalizador en el lecho L2

3.. La cantidad (%) del flujo de salida del reactor R que pasa por los cambiadores E1 y E3

4.. La cantidad total de vapor

5.. La cantidad de vapor que se inyecta al cambiador E2

6.. La cantidad de vapor que pasa por el calentador C2 para que la mezcla a la salida del calentador C1 sea de 1150° F

7.. La relación (vapor/hidrocarburos) a la entrada del lecho L1 del reactor R y a la entrada del lecho L2 del mismo reactor.

El objetivo es el siguiente

1. Minimizar la cantidad total de catalizador en el reactor R (lechos L1 y L2)

2. Que la producción sea lo más cercana a 140 T/D

3. Que el consumo de carga fresca sea la menor posible (Q)

4. Que el producto a deparación esté lo más frío posible (T).

Con lo anterior se puede plantear la siguiente función objetivo :

$$F = K_1 (L_1 + L_2) + K_2 (\text{prod} - 150)^2 + K_3 (Q) + K_4 (T)$$

donde K_1, K_2, K_3 y K_4 son constantes que se usan para "condicionar adecuadamente" la función objetivo.

Restricciones :

Debido a condiciones de operación de los equipos, se deben tomar en cuenta las siguientes restricciones

$$1000°F \leq \begin{matrix}\text{Temp. entrada}\\ \text{al reactor}\end{matrix} \leq 1280°F$$

$$\begin{matrix}\text{Temp. salida}\\ \text{del primer}\\ \text{lecho (L1)}\end{matrix} \leq \begin{matrix}\text{Temp. entrada}\\ \text{al segundo}\\ \text{lecho (L2)}\end{matrix} \leq 1250°F$$

$$1.0 \leq \begin{matrix}\text{Relación (Vap./hidrocarburos)}\\ \text{entrada al}\\ \text{primer lecho}\end{matrix} \leq 3.0$$

$$\begin{matrix}\text{Relación}\\ \text{(Vap./hidrocarburos)}\\ \text{entrada al 1}^{er}\text{ lecho}\end{matrix} \leq \begin{matrix}\text{Relación}\\ \text{(Vap./hidrocarburos)}\\ \text{entrada al 2º lecho}\end{matrix} \leq 3.0$$

Modelo de los equipos

Los equipos que es necesario simular $^{y}/_{o}$ dimensionar son los siguientes :

1. Calentadores (C1, C2, C3)

2. Cambiadores de calor

   a) Gas - Gas (E1 y E3)

   b) Gas - Liquido con evaporación (E2)

3. Reactor químico catalítico (R)

De los equipos mencionados sólo se describirá el modelo usado para el reactor catalítico, por ser este el equipo más importante de la planta. Para el resto de los equipos los modelos matemáticos son bastante convencionales (Ver., p.ej., D.D. Kern (Process Heat Transfer", Mc Graw-Hill Book Co., New York).

Cinética del sistema reaccionante

Se considera que en los lechos catalíticos L1 y L2 se llevan a cabo las siguientes reacciones. (A : vapor de agua).

1) $EB \rightleftarrows ES + H$ (catalítica)

2) $EB \longrightarrow B + E$ (catalítica)

3) $H + EB \longrightarrow T + M$ (catalítica)

4) $M + A \longrightarrow CO + H$ (catalítica)

5) $CO + A \longrightarrow CO2 + H$ (catalítica)

6) $E \longrightarrow AC + H$ (descomposición en fase vapor)

Las expresiones para la velocidad de reacción de cada una de las reacciones anteriores, son las siguientes

$$r_1 = \left(P_{EB} - \frac{P_{ES}\; P_H}{K_p} \right) \exp \left[-\frac{5715}{T} - 6.16 \right]$$

$$r_2 = \exp \left(-\frac{25600}{T} + 12.8 \right) P_{EB}$$

$$r_3 = \exp \left[-\frac{11000}{T} - 1.0 \right] P_{EB}\; P_H$$

$$r_4 = \exp \left(-\frac{7900}{T} - 3.36 \right) P_H$$

$$r_5 = P \exp \left(-\frac{8850}{T} + 3.8 \right] P_{CO}\; P_A$$

$$r_6 = 2.5^{*} 10^{6} \exp \left[-\frac{38000}{T} \right] P_B / T$$

donde

$$K_p = T^{0.549} \exp \left[-\frac{14516}{T} + 11.41 \right]$$

P : Presión en atmósferas

T : Temperatura

P_i : Presión parcial a cada componente

El calor liberado por cada una de las reacciones se tomó igual a

$$\Delta H_1 = 28{,}843 + 1.09\ T$$

$$\Delta H_2 = 25.992 - 1.09\ T$$

$$\Delta H_3 = 12{,}702 - 3.15\ T$$

$$\Delta H_4 = 50{,}046 + 3.96\ T$$

$$\Delta H_5 = 10{,}802 + 2.5\ T$$

$$\Delta H_6 = 38{,}278 + 11.45\ T$$

Con la anterior información, es posible establecer las relaciones que describen la variación, con la longitud, de cada uno de los componentes así como de la presión y temperatura, las cuales tendrán la forma general

$$\frac{d\, n_i}{d\, \ell} = f_i\ (n_1, n_2, \ldots, n_{NC}, T, P) \qquad i = 1,2, \ldots \text{ de componentes.}$$

$$\frac{d\, T}{d\, \ell} = g\ (n_1, n_2, \ldots, n_{NC}, T, P)$$

$$\frac{d\, P}{d\, \ell} = h\ (n_1, n_2, \ldots, n_{NC}, T, P)$$

con sus condiciones iniciales asociadas ($\ell = 0$)

Hay que tomar en cuenta que el volúmen de catalizador en cada uno de los lechos es una incognita, por lo que el límite superior de integración en cada lecho ($\ell = \ell_1$ y $\ell = \ell_2$) son variables, además de que entre ambos lechos existe una sección de mezclado con vapor.

Resultados

Los resultados que produjeron un valor mínimo de la función objetivo fueron los siguientes :

Temperatura de entrada a la primera cama	1243° F
Temperatura de entrada a la segunda cama	1143° F
Longitud de la primera cama	211.4 cm
Longitud de la segunda cama	157.1 cm
(Vapor/hidrocarburos) primera cama	2,738 lb/lb
Gasto de hidrocarburos en la primera cama	249.31 lbmol/h
(Vapor/hidrocarburos) segunda cama	2,923 lb/lb
ES a purificación	150.4 T/D
Conversión total	65.06 %
Selectividad total	81.84 %
Fracción de la salida del reactor a E 1 y E 3	64 %
Temperatura del producto a separación	562° F
Fracción del vapor total al cambiador E 2	8.3 %
Fracción del vapor total al calentador C 3	0 %

Comentarios Finales

El problema anteriormente descrito resulta ser, desde el punto de vista de minimización de funciones, uno de los más complejos ya que una sola evaluación de la función objetivo requiere de los siguientes cálculos

a) Solución de sistemas de ecuaciones diferenciales ordinarias no lineales

b) Cálculos iterativos en los cambiadores de calor ya que estos existen de antemano y los flujos y temperaturas de operación deben ajustarse al diseño mecánico que tienen.

c) Existe dentro del proceso un paso de recuperación de energía (cambiadores de calor), lo cual también genera que se hagan cálculos iterativos dentro de cada evaluación de la función objetiva.

En opinión del autor de este ejemplo, el modificar las condiciones de operación de una planta para ajustarlas a nuevos requerimientos es, con mucho, bastante más complejo que el diseño de una nueva planta ya que en este último caso se tienen muchos grados de libertad.

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO OPTIMO DE SISTEMAS DE INGENIERIA

OPTIMACION DE CIRCUITOS ELECTRICOS

DR.MARCO ANTONIO MURRAY LASSO

MARZO, 1982

REFERENCES

1. Scheerer, W. G.: Optimization with Computers, *Proc. Tutorial Symp. Circuit Design by Computers*, New York Univ., Department of Electrical Engineering, Feb. 1, 1967.
2. Fleischer, P. E.: Optimization Techniques, in F. F. Kuo and J. F. Kaiser (eds.), "System Analysis by Digital Computer," pp. 175-217, John Wiley & Sons, Inc., New York, 1966.
3. Linvill, J. G., and J. F. Gibbons: "Transistors and Active Circuits," McGraw-Hill Book Company, New York, 1961.
4. Llewellyn, F. B.: Some Fundamental Properties of Transmission Systems, *Proc. IRE*, vol. 40, pp. 271-282, March, 1952.
5. Rollet, J. M.: Stability and Power Gain Invariants of Linear Two-Ports, *IRE Trans. Circuit Theory*, pp. 22-32, March, 1962.
6. Raisbeck, G.: Definition of Passive Linear Networks in Terms of Time and Energy, *J. Appl. Phys.*, vol. 25, pp. 1510-1514, Dec., 1954.
7. Owens, J. L.: The Discrete Tantalum Nitride Thin Film Resistor on a Flat Embossed Ceramic Substrate, *Proc. Electron. Components Conf.*, Washington, May 5-7, 1965.
8. Wyndrum, R. W., Jr., and W. Pendergast: A Tantalum Film Gigacycle Amplifier, *Proc. Intern. Solid-State Circuits Conf.*, Philadelphia, pp. 20-21, 1965.
9. Kuo, C. Y., J. S. Fischer, and J. C. King: Thermal Processing of Tantalum Nitride Resistors, *Proc. Electron. Components Conf.*, *Washington*, May 5-7, 1965.
10. Owens, J. L., and N. G. Lesh: Further Developments in Discrete Tantalum Nitride Thin Film Resistors, *ibid.*
11. Wilde, D. J.: "Optimum Seeking Methods," Prentice-Hall, Inc., Englewood Cliffs, N. J., 1964.

4 ANALYSIS OF LINEAR INTEGRATED CIRCUITS BY DIGITAL COMPUTER USING BLACK-BOX TECHNIQUES

M. A. Murray-Lasso

Member, Technical Staff
Bell Telephone Laboratories, Inc.
Whippany, N. J.

4-1 INTRODUCTION

This chapter covers some methods of dealing efficiently with linear integrated circuits using black-box techniques and describes some computer programs implementing the methods discussed.

One of the drawbacks of the general-purpose analysis programs which are presently available (ECAP, NET-1, CIRCUS, PREDICT NASAP, SCEPTRE, etc.) [1-6] is that they can handle only lumped-circuit elements. As a result, the models for transistors and integrated circuits that the circuit designer is forced to use have parameters which are difficult to determine and it is a relatively complex matter to achieve good approximations to measured data over broad frequency ranges. For example, the circuit shown in Fig. 4-1 is a hybrid-pi linear model of a high-frequency transistor [7] including header and overlap diode capacitances which, in spite of its complexity, predicts the frequency behavior of y_{ie} with precision up to only a few MHz. For this reason, some designers use different models for each frequency range.

In [8] Carlin emphasizes the fact that one can no longer fall back on the comfortable security of coils, capacitors, resistors,

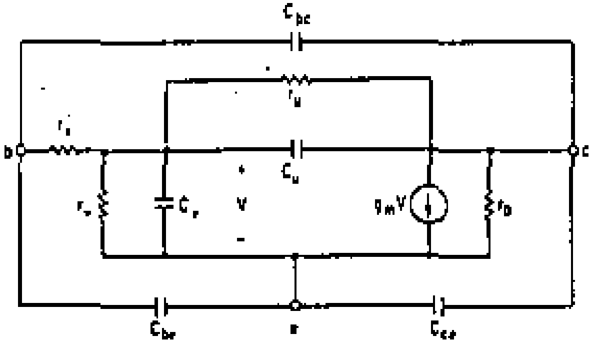


Fig. 4-1 Hybrid-pi model of a transistor with header and overlap diode capacitances.

and transformers in describing the details of construction of complex electrical devices. Independent of the internal construction of a device, if the circuit is linear and time-invariant, frequency response matrices in the complex plane will describe the device accurately. In some cases the frequency response matrix may be obtained experimentally without any knowledge of the internal structure. In other cases integrated circuits may be analyzed and the network may be characterized by a matrix whose entries are functions of frequency. The circuit's internal structure is ignored thereafter. This technique is called the black-box approach. It has the following advantages for integrated circuit analysis

1. It is able to accept experimental as well as analytical data.
2. Complete integrated circuits may be black-box modeled. This is extremely advantageous if the same circuits appear several times in a system.
3. With this approach there is no fundamental difference between lumped, distributed, or ideal networks (such as ideal low-pass filters).
4. Solution of networks by pieces may be done quite simply using this approach. This may become mandatory for analyzing circuits with many nodes on computers with limited memories.

We do not propose to abandon conventional device models since they are quite valuable for giving the designer insight into the interaction of the device with the rest of the circuit. What we propose is that a tool, such as the computer, should be used in ways appropriate to itself. Certain intuitive tools such as circuit schematics and Bode diagrams are appropriate for the pencil-and-paper designers. A computer, on the other hand, handles polynomials better than logarithmic graphs; it is simpler to fit experimental curves with polynomials than with circuit functions having certain pole-zero constellations. This does not include the designer from keeping in front of him a schematic with conventional models for insight.

The black-box curve modeling approach is most appropriate for handling interconnections involving single-chip linear integrated circuits; the only available points for connection, measurement, and characterization are the external terminals.

The four central ideas in this chapter are

1. In analyzing general linear stationary networks it should be possible to handle them as black boxes.
2. The modeling of a circuit with two or more terminals should be flexible enough to admit not only ratios of complex polynomials, but also general curves in the frequency domain.
3. It should be possible to preanalyze pieces of a circuit and eventually interconnect the pieces.
4. The indefinite matrices [12] are ideal vehicles for characterizing multiterminal black-box circuits which are arbitrarily interconnected.

4-2 FITTING FREQUENCY CURVES WITH STANDARD FUNCTIONS

For the analysis of interconnected black boxes, three methods are possible

1. Analytical expressions which give the terminal characteristics of the boxes.
2. Standard functions fitted to discrete data which were either measured or calculated.
3. Calculations performed only at the frequencies for which discrete data is available through either measurement or calculation.

There are some devices which are described accurately enough with analytical expressions over a limited frequency range. For example, smooth uniform transmission lines may be described with matrices whose entries contain hyperbolic functions over the

electrical length of the line. Lumped elements such as resistors, inductors, or capacitors may be characterized with the analytical expressions R, Ls, or $1/Cs$. Multiterminal lumped networks may be analyzed symbolically and characterized at given terminals with ratios of complex polynomials. Ideal elements such as bandpass filters with linear phase may be characterized by analytical phase and magnitude expressions. When such analytical expressions are available without too much additional effort and are accurate enough, they are preferable to other characterizations because of their convenience.

Sometimes analytical expressions are not available without considerable additional effort, but discrete data is. The discrete data may have been measured or calculated. If we know the general shape of the curve from which the discrete data points were sampled, the discrete data may be replaced by standard functions whose parameters are determined by a process of curve fitting [9]. The choice of standard functions is determined by the application. Some possibilities are

1. If the range of frequencies and the range of the values of the functions are small, low-order polynomials will generally be adequate for slowly varying functions.
2. If the range of frequencies is large and the range of values is small, low-order polynomials in the logarithm of the frequency are usually adequate.
3. If both the range of values and the range of frequencies are large, the logarithms of the values versus the logarithms of the frequencies may be fit with low-order polynomials.

If the data varies somewhat erratically, different curves may be fitted over each frequency range, for example, linear interpolation between strategically chosen data points.

The methods mentioned above are the simplest, and library routines are usually available to implement them. A user may, however, choose any functional form and adjust parameters with an optimization program if he so desires for some particular reason.

By having several subroutines available in a program, the user, depending on his problem, may through data cards control what type of standard functions are used for interpolation.

If the discrete data which is available is spaced sufficiently closely the analysis may be done only at the frequency values at which the data is available. This of course assumes that all the components described by discrete data are characterized at the same frequencies (or else that the analysis is done only at the frequencies for which all the elements are characterized). In this case curve fitting may be dispensed with altogether.

When approximating a given curve with a linear combination of a set of functions, there are certain problems that may arise. One of these problems is that calculations performed by computer carry a limited number of significant figures. Subtraction of large numbers to obtain small numbers may cause considerable error due to truncation. The situation will not arise if orthogonal functions are used in performing calculations by computer.

In function space the powers of x: $1, x, x^2, x^3, \ldots, x^n, \ldots$ are not orthonormal functions (orthogonal and normalized). The higher the order of the polynomials the less the projection of one upon another as exhibited by their inner product in the interval [0,1]

$$\langle x^p, x^q \rangle = \int_0^1 x^p \cdot x^q dx = \frac{x^{p+q+1}}{p+q+1}\Bigg]_0^1 = \frac{1}{p+q+1}$$

This means that when one fits a curve with a high-order polynomial, it will generally be necessary to carry a significant number of digits, especially for the higher powers. This is the reason for the following surprising fact: Although experimental data may be accurate to two significant figures, it may be necessary to provide polynomial coefficients which are accurate to seven figures. Unless one is aware of this one may be tempted to round off the polynomial coefficients to the same significant figures as the data. This would, of course, give very inaccurate results.

One way of alleviating this problem is to use functions other than powers of x. There exist several families of orthogonal polynomials which are orthonormal over different intervals [9]. Examples are: Legendre, orthogonal in [-1,1]; Chebyshev, First Kind, orthogonal in [-1,1]; Chebyshev, Second Kind, orthogonal in [-1,1]; Jacobi, orthogonal in [-1,1]; Generalized Laguerre, orthogonal in [0,∞]; and Hermite, orthogonal in [-∞,∞]. These polynomials are solutions to certain differential or difference equations with particular boundary conditions. One may generate one's own orthogonal polynomials over any interval using the Gram-Schmidt process. More generally one may use linear combinations of any set of functions (preferably orthogonal but not necessarily so) to data-fit. The choice will be a compromise between availability of subroutines to do the fitting, size, efficiency, and

accuracy of the subroutines, and (quite important) the willingness of the electronic circuit designer to work with analytical tools which he is not accustomed to using.

One possibility which has been found quite effective and simple to use is piecewise fitting of curves over different intervals. Discrete points may be fed into the computer and different curves may be fit on different intervals. For instance, one approach is to fit a quadratic polynomial through the first three points and to subsequently fit other quadratics through each additional point, matching the previous curve both in value and derivative at the last common point. With the aid of IF statements the routine decides what quadratic to use depending on the interval to which the independent value belongs. In this manner a continuous curve with a continuous derivative passing through the data points is obtained. More sophistication is obviously possible. A simplified version of this approach is simple linear interpolation between data points. This yields a continuous curve between data points although the derivative will be discontinuous. Linear interpolation has been found so effective and simple to use that it is used as the standard option in BELNAP[1] with the other options available on request. It is particularly suitable for the characterization of curves which are used only once in the program and will not become part of the permanent library. When a subcircuit (such as certain transistors) is used very often and becomes part of the permanent library of the program, it is worth the effort to fit its curves with computationally efficient expressions [10]. Optimization programs such as SUPROX [11] have been found very useful in this respect. With an optimization program one assumes an expression with some variable parameters and the program automatically determines the best parameters for the fit, thus providing an analytical model for the device. A particular case of this is to fit lumped circuits to frequency curves.

4-3 CHARACTERIZATION OF CIRCUITS WITH THE INDEFINITE ADMITTANCE MATRIX

There are many matrices that can characterize a multiterminal network. Examples are impedance matrices, hybrid matrices, and scattering matrices. Because of its simplicity when handling interconnected black boxes, the indefinite admittance matrix [12] (IAM) will be one of the characterizing matrices in this chapter.

The IAM is the short-circuit admittance matrix of a multinode network in which ports are formed between each terminal and a datum node which is "floating," i.e., unconnected to the circuit. When connecting two-terminal loads to a multiterminal network it is sufficient to characterize it as a multiport with ports defined at each terminal pair to which a load is connected. However, when an n-terminal network is arbitrarily interconnected with other multiterminal networks it is necessary to characterize the network at a set of n-1 ports corresponding to a complete and independent set of terminal pairs. A sufficient condition for the voltages of the n-1 ports to be Kirchhoff-voltage-law-independent is that a graph made of edges representing the port voltages form a tree [13].

It is trivially simple to go back and forth between the indefinite admittance matrix and a (definite) short-circuit admittance matrix when all the ports have a common node which is connected to the circuit (usually referred to as ground) [14]. If all the ports do not have a common ground, however, the IAM is not obtained quite as simply. This makes it necessary to have a method of going from a given set of ports to a second set having a common node. This can be accomplished by using the following formula

$$Y = C^T \hat{Y} C \tag{4-1}$$

where Y is the admittance matrix with ports having a common node, $\hat{Y}$ is the original admittance matrix whose ports from a tree, C is the transpose of reduced incidence matrix of the graph whose edges represent the ports of $\hat{Y}$, and C^T is the transpose of C [13, 14].

The use of Eq. (4-1) can be illustrated with the following example: The circuit of Fig. 4-2(a) represents a pair of mutually coupled coils. Let port 1 be defined by making node 1 the positive terminal and node 2 the negative one, and port 2 with node 3 positive and node 4 negative. The open-circuit impedance matrix of this two-port is

$$\hat{Z}_t = s \begin{bmatrix} L_{11} & L_{12} \\ L_{12} & L_{22} \end{bmatrix}$$

The inverse of $\hat{Z}_t$ is the admittance matrix with ports 1 and 2 defined as above

[1] A program to analyze the n-bus circuits which will be described below.

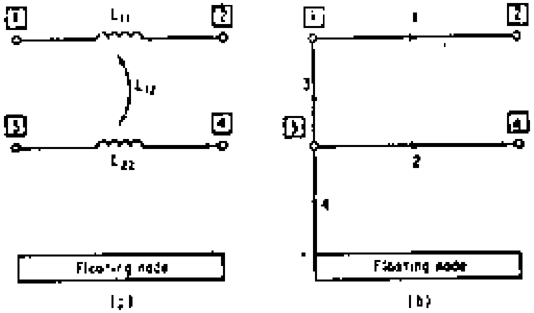


Fig. 4-2 (a) Two mutually-coupled inductors; (b) graph of the original four-port.

$$\tilde{Y}_1 = \tilde{Z}_1^{-1} = \frac{1}{s\left(L_{11}L_{22} - L_{12}^2\right)} \begin{bmatrix} L_{22} & -L_{12} \\ -L_{12} & L_{11} \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} \\ y_{12} & y_{22} \end{bmatrix}$$

To obtain the indefinite admittance matrix of the circuit of Fig. 4-2(a) it is necessary to obtain the short-circuit admittance matrix of the circuit having ports going from each terminal to a floating ground. To do this is is necessary to characterize the circuit at two additional ports. Let a third port be defined from nodes 3 to 1 in Fig. 4-2(a), and a fourth port from node 3 to the floating node. The admittance matrix of the circuit with the four ports as defined is

$$\tilde{Y} = \begin{bmatrix} y_{11} & y_{12} & 0 & 0 \\ y_{12} & y_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Now to apply Eq. (4-1), it is necessary to form the transpose of the reduced incidence matrix C of the graph defining the ports. This graph is shown in Fig. 4-2(b). The result is

$$C = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Thus Eq. (4-1) gives

$$Y = C^T\tilde{Y}C = \begin{bmatrix} 1 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_{11} & y_{12} & 0 & 0 \\ y_{12} & y_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$Y = \begin{bmatrix} y_{11} & -y_{11} & y_{12} & -y_{12} \\ -y_{11} & y_{11} & -y_{12} & y_{12} \\ y_{12} & -y_{12} & y_{22} & -y_{22} \\ -y_{12} & y_{12} & -y_{22} & y_{22} \end{bmatrix}$$

In similar fashion the indefinite admittance matrix of other typical components appearing in electronic circuits may be found. Some typical cases are presented compactly in Table 4-1.

Table 4-1 shows the indefinite admittance matrices of some typical devices when they are connected to the lowest-numbered nodes in a circuit (1, 2, 3, etc.). If a device is connected to nodes $i, j, k, \ldots$ instead of to 1, 2, 3, ..., then a simple replacement of indices i for 1, j for 2, k for 3, etc., gives the location of the entries to the corresponding indefinite admittance matrix. The total indefinite admittance matrix of a complicated circuit is simply obtained by subsequently adding in the proper positions the contributions of each of the devices. This avoids manipulating any topological matrices, since the indefinite admittance matrix is actually obtained by inspection. The entries of Table 4-1 should enable the reader to obtain by inspection the IAM of most linear transistor circuits appearing in practice. This topic is considered in more detail in the next section.

4-3.1 Analysis of Black-box Circuits Through the Indefinite Admittance Matrix

Since the indefinite admittance matrix (IAM) is the n-port short-circuit admittance matrix in which the ports are formed between each terminal and an unconnected node which is floating, the current going into each terminal may always be considered to

Table 4.1. Indefinite Admittance Matrices of Some Typical Subcircuits

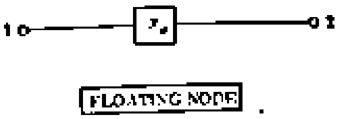
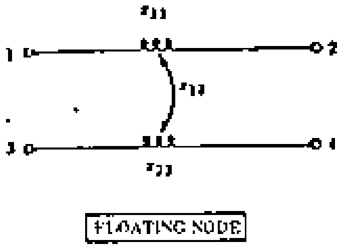
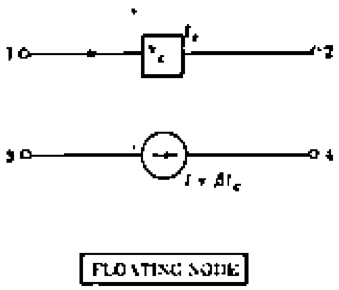
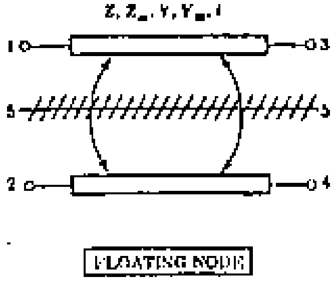
Circuit	Original $\hat{Y}$ matrix where ports form tree $\hat{Y}$	Final Y matrix when ports have common node Y
Element of admittance y_a connected between nodes 1 and 2  	PORT NODES 1 1, 2 2 1, F $\hat{Y} = \begin{bmatrix} y_a & 0 \\ 0 & 0 \end{bmatrix}$ $C = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}$	PORT NODES 1 1, F 2 2, F $Y = \begin{bmatrix} y_a & -y_a \\ -y_a & y_a \end{bmatrix}$
Mutually coupled coils of self-impedances z_{11}, z_{22} and mutual impedance z_{12}  z_{11} 1 2 z_{12} 3 4 z_{22} FLOATING NODE	PORT NODES 1 1, 2 2 3, 4 3 3, 1 4 3, F $\hat{Z}_1 = \begin{bmatrix} z_{11} & z_{12} \\ z_{12} & z_{22} \end{bmatrix}; \hat{Y}_1 = \hat{Z}_1^{-1}$ $\hat{Y} = \begin{bmatrix} y_{11} & y_{12} & 0 & 0 \\ y_{12} & y_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ $C = \begin{bmatrix} 1 & +1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	PORT NODES 1 1, F 2 2, F 3 3, F 4 4, F $Y = \begin{bmatrix} y_{11} & -y_{11} & y_{12} & -y_{12} \\ -y_{11} & y_{11} & -y_{12} & y_{12} \\ y_{12} & -y_{12} & y_{22} & -y_{22} \\ -y_{12} & y_{12} & -y_{22} & y_{22} \end{bmatrix}$
Current controlled current source Gain β controlled by current in y_c  I_c 1 Y_c 2 3 4 $I = \beta I_c$ FLOATING NODE	PORT NODES 1 1, F 2 2, F 3 1, 2 4 3, 4 $\bar{Y} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & y_c & 0 \\ 0 & 0 & \beta y_c & 0 \end{bmatrix}$ $C = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & +1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$	PORT NODES 1 1, F 2 2, F 3 3, F 4 4, F $Y = \begin{bmatrix} y_c & -y_c & 0 & 0 \\ -y_c & y_c & 0 & 0 \\ \beta y_c & -\beta y_c & 0 & 0 \\ -\beta y_c & \beta y_c & 0 & 0 \end{bmatrix}$

Table 1.1. Indefinite Admittance Matrices of Some Typical Subcircuits (Continued)

Circuit	Original $\hat{Y}$ matrix where ports form tree $\hat{Y}$	Final Y matrix when ports have common node Y
Pair of identical, uniform coupled lines of longitudinal impedance Z, mutual impedance Z_m, admittance to ground Y, and admittance between lines Y_m, all per unit length. Length of lines l. [7]  Z, Z_m, Y, Y_m, l 1 3 5 5 2 4 FLOATING NODE	PORT NODES 1 1,5 2 2,5 3 3,5 4 4,5 5 5,F $\hat{Y} = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} & 0 \\ y_{12} & y_{11} & y_{14} & y_{13} & 0 \\ y_{13} & y_{14} & y_{11} & y_{12} & 0 \\ y_{14} & y_{13} & y_{12} & y_{11} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $C = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	PORT NODES 1 1,F 2 2,F 3 3,F 4 4,F 5 5,F $Y = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} & y_{15} \\ y_{12} & y_{11} & y_{14} & y_{13} & y_{25} \\ y_{13} & y_{14} & y_{11} & y_{12} & y_{35} \\ y_{14} & y_{13} & y_{12} & y_{11} & y_{45} \\ y_{51} & y_{52} & y_{53} & y_{54} & y_{55} \end{bmatrix}$

where

$$y_{11} = \frac{1}{2} M \coth(Pl) + \frac{1}{2} N \coth(Ql)$$

$$y_{12} = \frac{1}{2} M \coth(Pl) - \frac{1}{2} N \coth(Ql)$$

$$y_{13} = -\frac{1}{2} M \operatorname{csch}(Pl) - \frac{1}{2} N \operatorname{csch}(Ql)$$

$$y_{14} = -\frac{1}{2} M \operatorname{csch}(Pl) + \frac{1}{2} N \operatorname{csch}(Ql)$$

$$M = \sqrt{\frac{Y}{Z + Z_m}},\quad N = \sqrt{\frac{Y + 2Y_m}{Z - Z_m}},\quad P = \sqrt{(Z + Z_m)Y},\quad Q = \sqrt{(Z - Z_m)(Y + 2Y_m)}$$

$$y_{15} = y_{25} = y_{35} = y_{45} = y_{51} = y_{52} = y_{53} = y_{54} = -(y_{11} + y_{12} + y_{13} + y_{14})$$

$$y_{55} = -(y_{15} + y_{25} + y_{35} + y_{45}) = 4(y_{11} + y_{12} + y_{13} + y_{14})$$

come out of the floating node under any type of interconnection and hence the condition for having a proper port is never lost. In the circuit of Fig. 4-3, the subnetworks N1, N2, and N3 can each be considered as five terminal networks, each one having one node dangling, and each having its own IAM. The IAM of the whole circuit is simply the sum of the indefinite admittance matrices of the subnetworks. This is very simple to program in a digital computer. All that is necessary is to know the contribution of each subnetwork separately. Each subnetwork may range from a simple resistor to a complicated integrated circuit or combinations of several of them.

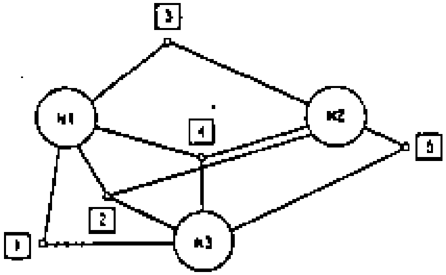


Fig. 4-3 Black-box network.

The principal properties of the indefinite admittance matrix are summarized here for convenience [12]

1. The sum of entries in each row is zero.
2. The sum of entries in each column is zero.
3. If the floating node is connected to the kth terminal of the circuit, the (definite) nodal matrix of the circuit with the kth terminal as datum is the indefinite admittance matrix with the kth row and the kth column deleted.
4. If the first p terminals of an n-terminal network are connected together, the new indefinite admittance matrix of the resultant $n-p+1$ terminal network is an $(n-p+1)\times(n-p+1)$ matrix obtained as follows: first form an intermediate matrix Q by substituting for the first p columns of the indefinite admittance matrix Y one column whose entries are the sums of the first p columns of Y. Next substitute for the first p rows of Q one row whose entries are the sums of the entries of the first p rows of Q. The resultant matrix is the new indefinite admittance matrix of the $n-p+1$ terminal network.

Once the indefinite admittance matrix of the circuit is established, it may happen that not all the nodes are of interest for forming ports. Some ports may be defined to simplify the modification of certain elements [13], or because the network will eventually be connected to other networks to form a still larger network. The nodes where ports will not be defined may be considered as internal nodes and suppressed.

The IAM is partitioned as in the following equation

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \tag{4-2}$$

where I_1, V_1, are p-vectors, I_2, V_2 are $(n-p)$-vectors, Y_{11} is a $p\times p$ matrix, Y_{12} is a $p\times(n-p)$ matrix, Y_{21} is a $(n-p)\times p$ matrix, and Y_{22} is an $(n-p)\times(n-p)$ matrix. Assuming the first p terminals are the external terminals, $I_2 = 0$, since nothing will be connected to the internal terminals. Letting $I_2 = 0$ in Eq. (4-2) and solving the second line of Eq. (4-2) for V_2 in terms of V_1

$$V_2 = -Y_{22}^{-1}Y_{21}V_1 \tag{4-3}$$

which when substituted on the first line of Eq. (4-2) gives

$$I_1 = \left(Y_{11} - Y_{12}Y_{22}^{-1}Y_{21}\right)V_1 \tag{4-4}$$

Thus the suppressed admittance matrix Y_S is

$$Y_S = Y_{11} - Y_{12}Y_{22}^{-1}Y_{21} \tag{4-5}$$

The external node voltages may be solved for using Y_S, which is of smaller size than Y. Once the voltages V_1 of the external terminals are determined, the voltages of the internal terminals V_2 may be determined using Eq. (4-3).

It may happen that the internal nodes are connected to independent current sources. In this case I_2 is not zero when suppressing internal nodes in Eq. (4-2) but it has entries of values equal to the currents injected into each terminal by the independent current sources. In this case

$$V_2 = -Y_{22}^{-1}Y_{21}V_1 + Y_{22}^{-1}I_2 \tag{4-6}$$

which when substituted in the first line of Eq. (4-2) gives

$$I_1 = \left(Y_{11} - Y_{12}Y_{22}^{-1}Y_{21}\right)V_1 + Y_{12}Y_{22}^{-1}I_2 \qquad (4\text{-}7)$$

which may be written

$$I_{eq} = \left(Y_{11} - Y_{12}Y_{22}^{-1}Y_{21}\right)V_1 \qquad (4\text{-}8)$$

with

$$I_{eq} = I_1 - Y_{12}Y_{22}^{-1}I_2. \qquad (4\text{-}9)$$

Equations (4-8) and (4-9) indicate that the internal current sources causing I_2 may be replaced by external currents $-Y_{12}Y_{22}^{-1}I_2$ and added to the original external currents I_1 to form an equivalent external current vector I_{eq}.[1]

In order to analyze very large circuits, one may divide the circuit into pieces which may coincide with functional divisions. The IAM of each piece is obtained and the internal nodes are suppressed. All the previously suppressed black boxes may then be interconnected and a larger IAM including only external nodes is obtained and used to solve for all external terminal voltages. Finally in each subnetwork the internal voltages may be obtained. The saving comes from the fact that the internal nodes are not all in memory at once. This method is akin to Kron's method of tearing [16], though it does not invoke tensors. As presented here, it may include active devices and also distributed elements.

4-3.2 Calculation of Definite Matrices and Terminal Characteristics

Once the total suppressed IAM of a circuit is obtained, the matrix may be made definite by connecting one of the nodes (called ground) to the floating node and deleting the corresponding row and column to obtain the node-to-ground definite admittance matrix Y_{SG}. The node-to-ground matrix may be inverted to obtain the node-to-ground impedance matrix Z_{SG}. Since it may be desired to form ports other than node-to-ground, a transformation from node-to-ground ports to other ports is necessary. The formula to use is

$$Z = DZ_{SG}D^T \qquad (4\text{-}10)$$

[1] In case independent voltage sources exist, Norton-equivalent independent current sources may replace them and the method applied to them.

where Z is the impedance matrix with the desired ports, and D is the transpose of the reduced incidence matrix of the graph corresponding to the desired ports. D^T is the transpose of D and Z_{SG} is the impedance matrix with ports defined from the nodes to ground [17].

Let the Z matrix of Eq. (4-10) have elements Z_{ij}

$$Z = (Z_{ij}) \qquad (4\text{-}11)$$

Then the voltage ratios are given by

$$\frac{V_i}{V_j} = \frac{Z_{ij}}{Z_{jj}} \qquad (4\text{-}12)$$

It is usually convenient to consider the loads as part of the network under analysis. Thus Z is the impedance matrix with the loads connected. Often it is of interest to find the driving-point impedance of a loaded circuit but excluding the generator impedance at the port looked into. To do this one may simply connect to the port in question an additional impedance equal to the negative of the generator impedance. Denoting with Z_{iil} the load impedance of the generator at port i, the driving point impedance z_i at port i is given by

$$z_i = \frac{1}{(1/Z_{ii}) - (1/Z_{iil})} \qquad (4\text{-}13)$$

where Z_{ii} is the ith element on the main diagonal of the matrix Z of Eq. (4-10). The return loss RL_{ii} at port i is given in dB by

$$RL_{ii} = 20\log\left|\frac{z_i + Z_{iil}}{z_i - Z_{iil}}\right| \qquad (4\text{-}14)$$

The insertion voltage gain IG_{ij} between ports i and j is given in dB by

$$IG_{ij} = 20\log\frac{(V_j/V_i)[z_i/(z_i + Z_{iil})]}{[Z_{jjl}/(Z_{jjl} + Z_{iil})]} \qquad (4\text{-}15)$$

4-3.3 Combining the Indefinite Admittance Matrix and Curve Modeling for Analysis of Integrated Circuits

A problem that commonly occurs in practice is the analysis of a circuit in which certain subcircuits recur. For example, a

circuit may contain several identical operational amplifiers. A convenient way of analyzing such circuits is the following: analyze each subcircuit separately and obtain its indefinite admittance matrix at several frequencies suppressing the internal terminals (terminals not connected to the rest of the network). Repeated subcircuits are analyzed only once. Model the subcircuits with polynomials fitting the frequency behavior of the real and imaginary parts of each entry in the admittance matrix. For example, if cubics are used for a circuit with four external terminals, 128 coefficients will model the device (4 (coefficients/cubic) times $4 \times 4 \times 2$ (real and imaginary elements/matrix of 4 terminal network)). Interconnect the subcircuits to form a larger indefinite admittance matrix and again suppress the internal terminals. Repeat the process as required until the whole network is analyzed and the terminals of interest are the only external terminals.

Note that this method is essentially the same method one uses in designing large systems, where each subsystem is designed separately and eventually the different subsystems are connected. Several levels of subsystems may of course be used.

One of the important points to notice here is that a whole network which may have been analyzed in several levels of tearing may still be modeled with polynomials (or other standard interpolation methods) regardless of how complicated the internal structure of the network is. What we are trying to point out is that one can work advantageously with "universal" models which do not have to be the classical R, L, C, controlled-source, schematic models which network designers are accustomed to using.

In handling circuits with admittance matrices, certain degeneracies may occur for pieces of the network even though the whole circuit has a nondegenerate admittance matrix. In many cases these problems can be circumvented with special circuit techniques.

The indefinite admittance method is geared towards imbedded independent current sources rather than voltage sources. If the voltage source has an impedance in series, Norton-equivalent current sources may replace the voltage sources. If there is no impedance in series with the voltage source, one may add a positive and negative impedance of equal value in series and associate the voltage source with either to produce a Norton-equivalent current source. This method will create an additional node. Similar tricks may be employed for handling voltage-controlled voltage sources.

Certain multiterminal elements do not have an indefinite admittance matrix because infinite elements in the matrix are created due to what essentially amounts to short circuits between some terminals. One may avoid the degeneracy by using the trick of adding and subtracting elements in series. For example, perfectly coupled inductances and ideal transformers have no admittance matrix. However, by adding positive and negative resistances in series with each winding and considering the positive resistances as though they were associated with the windings and the negative resistances as elements separate from the device, the degeneracy disappears. This procedure may also be used for black boxes containing more terminals.

4-4 DESCRIPTION OF THE BELNAP PROGRAM

A computer program known as BELNAP (Bell Electronic Linear Network Analysis Program),[1] using the methods discussed above, has been written completely in FORTRAN IV and has been implemented on a GE-635 computer at Bell Telephone Laboratories [18, 19]. (A flowchart of the BELNAP appears in Table 4-2.)

The circuits which BELNAP can handle may include

1. ordinary positive and negative R, L, C, M elements
2. current-controlled current sources
3. distributed elements such as transmission lines, RC lines, coupled transmission lines
4. black boxes with n terminals which have been characterized at $n - 1$ independent ports. The characterization may be done with admittance matrices, impedance matrices, scattering matrices (for n-terminal networks) and h parameters or $ABCD$ parameters (for three-terminal networks)

The characterization of the black boxes may be done with tables of values at discrete frequencies, parameters of standard families of functions, or codes which call subroutines containing analytical models of the devices.

The program is limited to the following maxima: 40 nodes, 20 subnetworks of up to 8 ports each, 20 frequencies, 100 resistors, 100 inductors (25 mutuals), 100 capacitors, and 20 controlled current sources.

[1] BELNAP was written by L. A. Davieau of Bell Telephone Laboratories.

Table 4-2. Flowchart of BELNAP

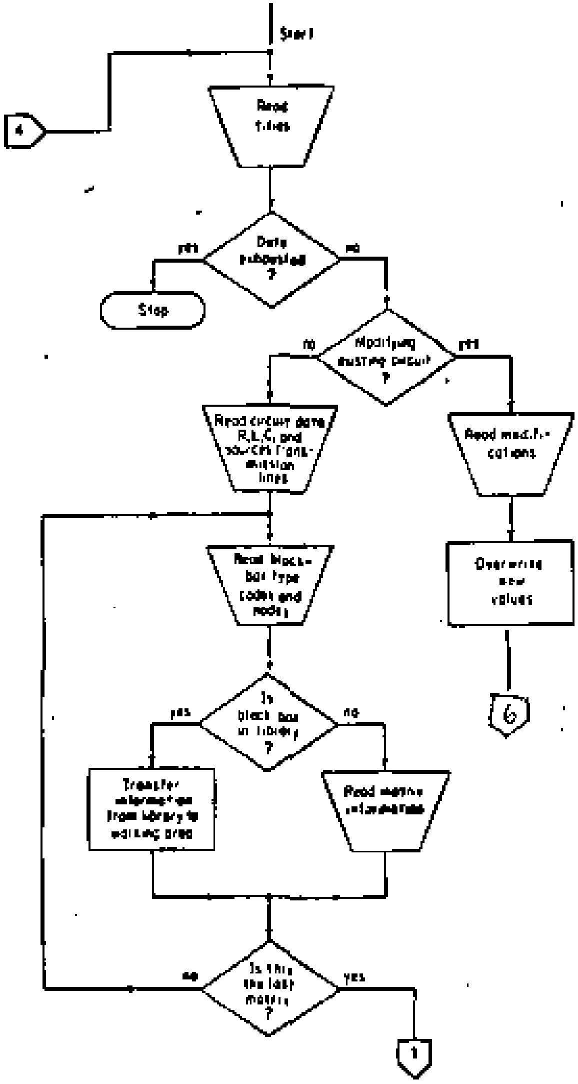

Start
4
Read titles
Data exhausted ?
yes
no
Stop
Modifying existing circuit ?
no
yes
Read circuit data R,L,C, and sources trans-mission lines
Read modifi-cations
Read black-box type codes and nodes
Overwrite new values
6
Is black box in library ?
yes
no
Transfer information from library to working area
Read matrix information
Is this the last matrix ?
no
yes
1

Table 4-2. Flowchart of BELNAP (Continued)

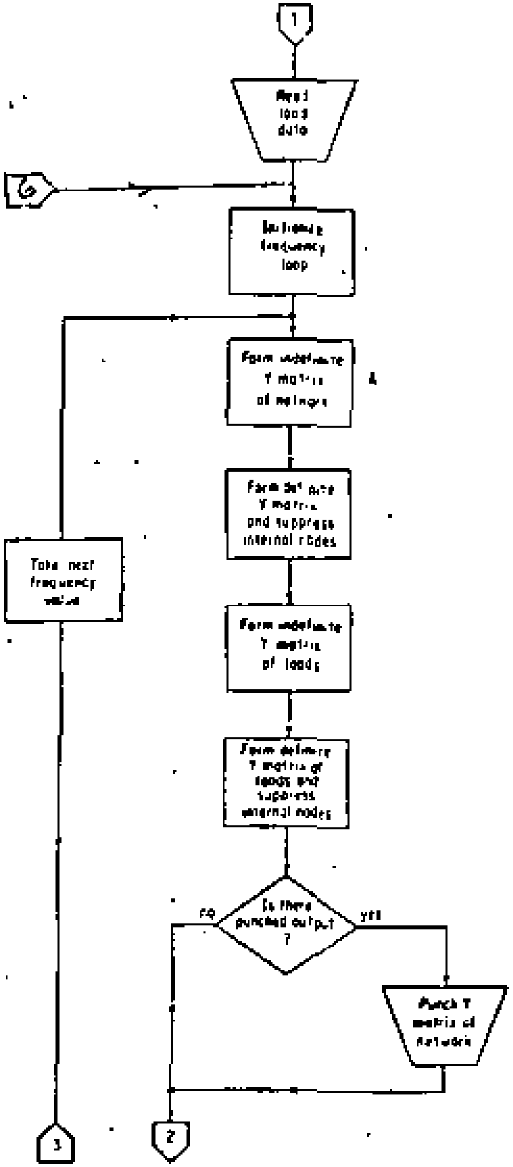

1
Read load data
6
Initialize frequency loop
Form indefinite Y matrix of network
4
Form definite Y matrix and suppress internal nodes
Take next frequency value
Form indefinite Y matrix of loads
Form definite Y matrix of loads and suppress internal nodes
Is there punched output ?
no
yes
Punch Y matrix of Network
3
2

Table 4-2. Flowchart of BELNAP (Continued)

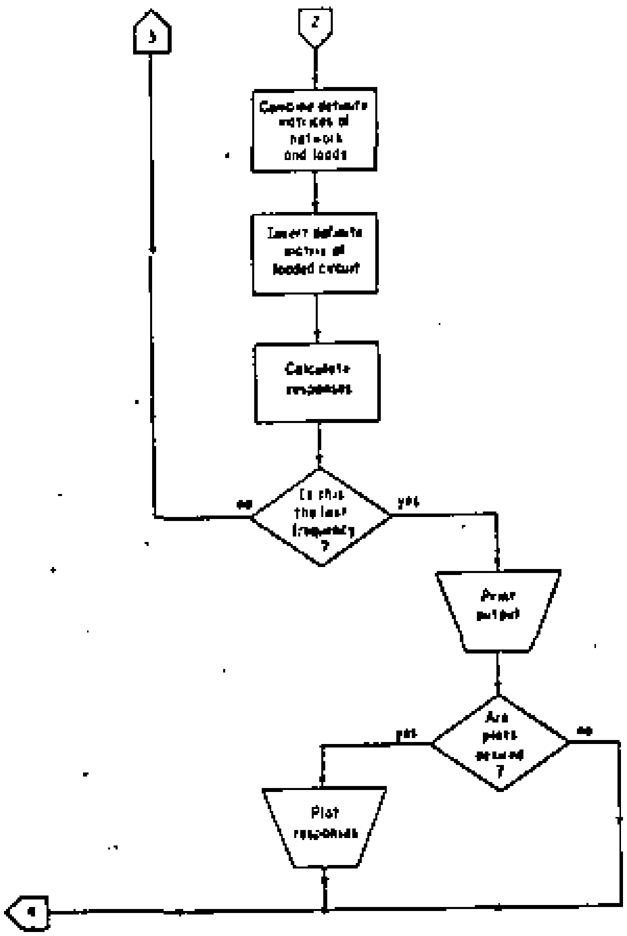
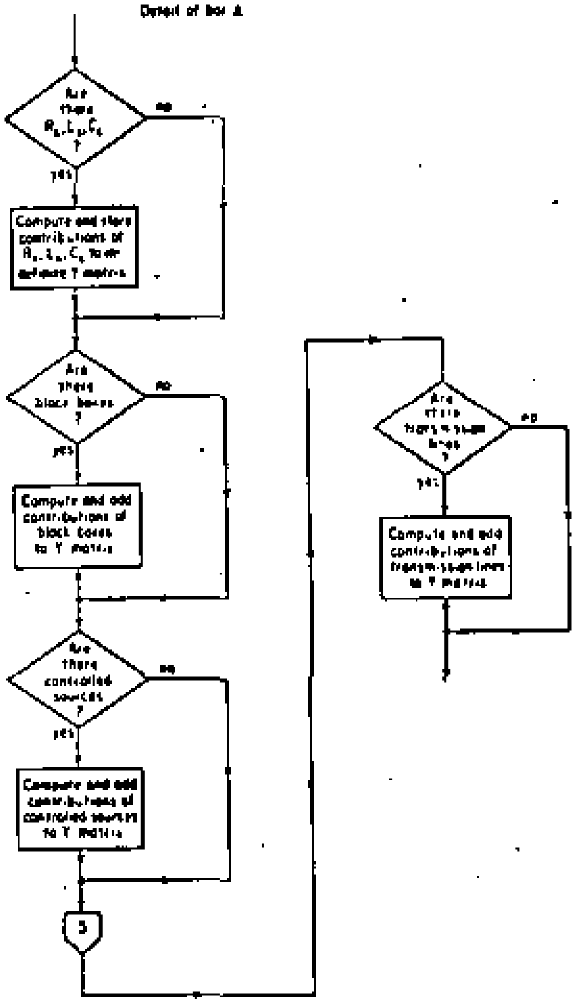

3
2
Combine definite matrices of network and loads
Invert definite matrix of loaded circuit
Calculate responses
no
Is this the last frequency ?
yes
Print output
yes
Are plots desired ?
no
Plot responses
4

Table 4-2. Flowchart of BELNAP (Continued)


Detail of box A
Are there R_k, L_k, C_k ?
no
yes
Compute and store contributions of R_k, L_k, C_k to the definite Y matrix
Are there black boxes ?
no
yes
Compute and add contributions of black boxes to Y matrix
Are there controlled sources ?
no
yes
Compute and add contributions of controlled sources to Y matrix
3
Are there transmission lines ?
no
yes
Compute and add contributions of transmission lines to Y matrix

These limitations depend exclusively on memory requirements and can be changed by changing dimension statements. One of the versions used at Bell Laboratories has lower capabilities in order to be able to run in the express option for which it is required to use less than 40K of memory.

One of the features of the program is a library for commonly used black boxes. If a black box which is in the library appears in a circuit, all that is needed is to input it with a code name and to give the nodes to which it is connected in a given order.

A MODIFY option allows a user to change the values of *R*, *L*, *C*, or controlled-source betas and repeat the analysis without having to feed complete circuit descriptions. Several types of transmission lines can be specified by giving their characteristics. The program accepts discrete measured data in the form of admittance, impedance, or scattering *n* × *n* matrices and *H* parameters or *ABCD* parameters for three-terminal networks. Values at intermediate frequencies are obtained automatically by linear interpolation.

The input to the program is user-oriented and in free format using the NAMELIST feature of FORTRAN IV. For the input of special elements, several choices exist, including, for example, coefficients for frequency power series or coefficients for the log of the function in terms of powers of the log of frequency.

The program computes the indefinite admittance matrix of the network at given discrete frequencies, suppresses the internal nodes indicated by the user, and punches on request the suppressed indefinite admittance matrix. It then inverts the reduced definite admittance matrix to obtain the nodal impedance matrix from which it obtains the desired driving point impedance at each port (excluding generator impedance), insertion voltage gains, voltage ratios, and return losses. The output may be obtained in either tabular form or in the form of frequency plots produced by a SC-4020 microfilm plotter.

4-4.1 Analysis of a Balanced Amplifier Using BELNAP

As an example of the kinds of problems that BELNAP may handle, the analysis of a balanced amplifier that contains distributed and lumped elements is given [20, 21]. The transistors have been characterized by actual measurements.

A general schematic of the amplifier is given in Fig. 4-4. The triangles represent identical transistor amplifiers and the square boxes represent stripline directional couplers. The schematic of a transistor amplifier appears in Fig. 4-5.

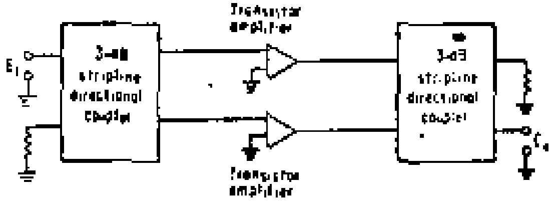

Transistor amplifier
E1
3-db stripline directional coupler
3-db stripline directional coupler
E2
Transistor amplifier

Fig. 4-4 Schematic of balanced amplifier containing lumped and distributed elements.

The analysis was done as follows: Certain portions of the network were preanalyzed or measured at a discrete set of frequencies and the resulting real parts and imaginary parts were fitted with standard functions.

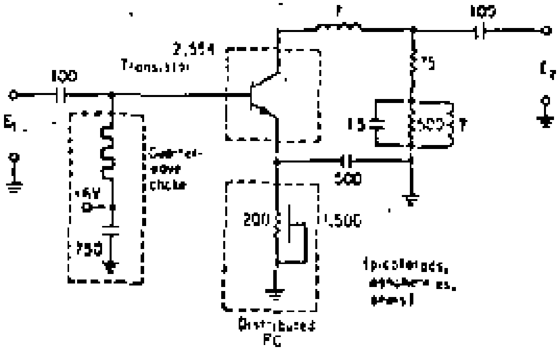


Fig. 4-5 Circuit schematic of a transistor amplifier.

In Fig. 4-5 the portions inside the dotted rectangles were precharacterized. The transistor was characterized by measuring its scattering matrix at discrete frequencies, converting to Y matrix, and fitting the discrete points with polynomials. The quarter-wave meander choke terminated in a capacitor was theoretically characterized at several discrete frequencies by calculations with an analytical expression that includes dissipation effects and the discrete values were fitted with standard functions.

The *RC* distributed circuit was modeled with 20 cascaded lumped *RC* sections which were analyzed and fitted with polynomials. (Although theoretical expressions could have been used, this method was chosen for illustrative purposes.) Once the elements in the dotted rectangles had been characterized as black boxes, the whole amplifier of Fig. 4-5 containing both lumped elements and black boxes was analyzed and characterized as a black box. Finally the whole amplifier of Fig. 4-4 was analyzed (the directional couplers having previously been analyzed with a special subroutine for coupled striplines) and the results printed and plotted [22].

A sample of the intermediate results are shown in Table 4-3 where a list of values of the real and imaginary part of y_{11} of the transistor of the amplifier for 10 discrete frequencies is shown and where the polynomials that were internally fitted to this data appear at the bottom of the table. This is also shown graphically in Figs. 4-10 and 4-11 where the rectangles give the measured data and the continuous curves give the points calculated with the polynomials fitted. Figures 4-6, 4-7, and 4-8 show respectively the magnitude and phase of E_2/E_1 of the transistor amplifier of Fig. 4-5, the return loss at port 4 of the complete balanced amplifier of Fig. 4-4, and the magnitude and phase of the ratio of E_4/E_1 of the complete amplifier of Fig. 4-4 as produced by the microfilm plotter subroutine of BELNAP.

Table 4-3. Polynomial Fit of y_{11} of the CE-2554 Transistor

MHz	Experimental data		Points computed by polynomial	
	Real part	Imaginary part	Real part	Imaginary part
100	0.53×10^{-2}	0.205×10^{-2}	0.526×10^{-2}	0.204×10^{-2}
135	0.55×10^{-2}	0.34×10^{-2}	0.553×10^{-2}	0.344×10^{-2}
180	0.61×10^{-2}	0.40×10^{-2}	0.612×10^{-2}	0.460×10^{-2}
250	0.73×10^{-2}	0.60×10^{-2}	0.725×10^{-2}	0.591×10^{-2}
330	0.86×10^{-2}	0.71×10^{-2}	0.859×10^{-2}	0.707×10^{-2}
430	1.05×10^{-2}	0.825×10^{-2}	1.05×10^{-2}	0.836×10^{-2}
610	1.29×10^{-2}	0.93×10^{-2}	1.28×10^{-2}	0.941×10^{-2}
825	1.54×10^{-2}	1.00×10^{-2}	1.55×10^{-2}	0.978×10^{-2}
1130	1.85×10^{-2}	0.675×10^{-2}	1.84×10^{-2}	0.884×10^{-2}
1500	2.19×10^{-2}	0.34×10^{-2}	2.19×10^{-2}	0.339×10^{-2}

$$\operatorname{Re} y_{11} = 2.587270 \times 10^{-2} - 4.987302 \times 10^{-3}\phi - 4.316561 \times 10^{-4}\phi^2$$
$$- 2.752786 \times 10^{-3}\phi^3 + 4.545855 \times 10^{-3}\phi^4 - 1.049891\phi^5$$
$$\operatorname{Im} y_{11} = -1.670402 \times 10^{-1} + 6.071215 \times 10^{-2}\phi - 2.292190 \times 10^{-3}\phi^2$$
$$- 8.264316 \times 10^{-4}\phi^3 - 3.801567 \times 10^{-4}\phi^4$$
$$+ 1.124385 \times 10^{-4}\phi^5 - 7.414405 \times 10^{-6}\phi^6$$

where $\phi = \log_e(f)$ and f is in MHz.

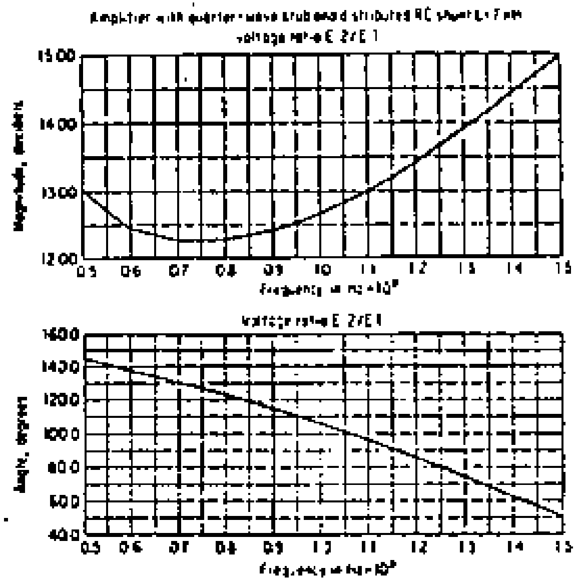


Fig. 4-6 Voltage ratio E_2/E_1 of a transistor amplifier.

4-4.2 BELTIP: Transient Version of BELNAP

Once the desired characteristics of a circuit are obtained in the complex frequency domain, it is possible to obtain them in the time domain by numerical Laplace transform inversion. Let $F(s)$ be the Laplace transform of a time function $f(t)$. The inverse transform formula is

$$f(t) = \frac{1}{2\pi j}\int_{c-j\infty}^{c+j\infty} F(s)e^{st}\,ds \tag{4-16}$$

evaluated along a line to the right of the singularities of $F(s)$.

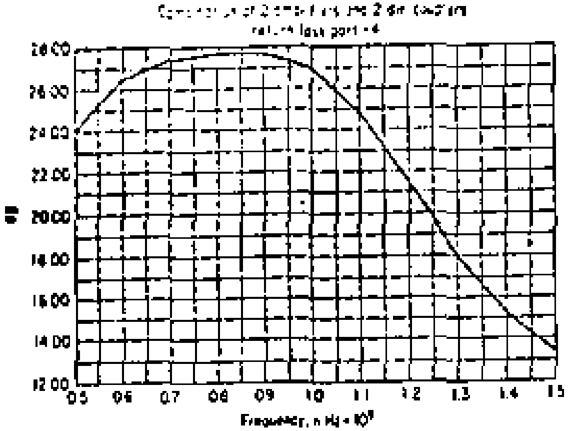


Fig. 4-7 Return loss at port 4 of the balanced amplifier.

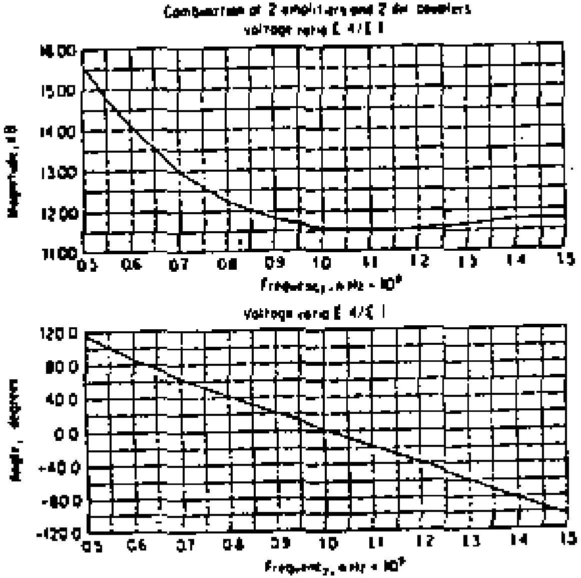


Fig. 4-8 Voltage ratio E_4/E_1 of the balanced amplifier.

The dummy integration variable s may be written $s = \sigma + j\omega$, with σ = constant. With the new integration variable ω, Eq. (4-16) reads

$$f(t) = \frac{1}{2\pi j}\int_{\omega=-\infty}^{+\infty} F(\sigma + j\omega)e^{(\sigma+j\omega)t}j\,d\omega$$

from which

$$e^{-\sigma t}f(t) = \frac{1}{2\pi}\int_{\omega=-\infty}^{+\infty} F(\sigma + j\omega)e^{j\omega t}\,d\omega \tag{4-17}$$

which places in evidence the fact that $F(\sigma + j\omega)$ is the Fourier transform of the function $e^{-\sigma t}f(t)$.

Let us assume that $\int_{\omega=\omega_c}^{\infty}|F(\sigma + j\omega)|\,d\omega$ is negligible for some ω_c. For instance, this will be true if $|F(\sigma + j\omega)|$ goes down as $1/\omega^2$ nears infinity. Let us also assume that $F(\sigma + j\omega)e^{j\omega t}$ varies slowly enough with ω so that it may be approximated by a series of piecewise complex constants of values $F(\sigma + jk\Delta\omega)e^{jk\Delta\omega t}$, $k = 0, \pm 1, \pm 2, \ldots, \pm(N-1)/2$; $\Delta\omega = 2\omega_c/N$; N odd; for equal intervals of length $\Delta\omega$.

The integral of Eq. (4-17) may be approximated with a sum as follows

$$\frac{1}{2\pi}\int_{\omega=-\infty}^{+\infty} F(\sigma + j\omega)e^{j\omega t}\,d\omega = \frac{\Delta\omega}{2\pi}\sum_{k=-(N-1)/2}^{(N-1)/2} F(\sigma + jk\Delta\omega)e^{jk\Delta\omega t} \tag{4-18}$$

Now note that the last sum may be interpreted as a finite exponential Fourier series whose coefficients are $(\Delta\omega/2\pi)F(\sigma + jk\Delta\omega)$. The time function which the sum of Eq. (4-18) adds up to is a periodic function of period $T = 2\pi/\Delta\omega$. If the quantity $\Delta\omega$ is small, the function will repeat itself after a long period of time T. In the limit as $\Delta\omega \to 0$, $N \to \infty$, $T \to \infty$, $\omega_c \to \infty$, the function ceases to repeat itself (since the period is infinite) and the approximate sum of Eq. (4-18) gives the exact value of the integral on the left.

There are three parameters to be chosen, σ, ω_c, and N. Their choice is made according to the following considerations. The larger the σ the better the function $F(\sigma + j\omega)e^{j\omega t}$ can be approximated piecewise by constants since the function will be evaluated far from the singularities. On the other hand, the time function obtained has to be multiplied at the end by $e^{\sigma t}$; any errors at large values of time are thus exaggerated. The cutoff frequency ω_c

should be such that

$$\epsilon = \frac{1}{\pi}\int_{+\omega_c}^{+\infty} |F(\sigma + j\omega)|\,d\omega \qquad (4\text{-}19)$$

is small. The quantity ϵ gives a bound on the maximum error at any time due to truncation of the spectrum in calculating $e^{-\sigma t}f(t)$. The number N is chosen so that the function $F(\sigma + j\omega)e^{j\omega t}$ remains reasonably constant during the intervals of length $\Delta\omega = 2\omega_c/N$. If the function varies rapidly, N will have to be large so that the intervals of length $\Delta\omega$ be small.

A sum of the form of Eq. (4-18) is evaluated using a modification of the Cooley-Tukey algorithm [23, 24] for fast complex Fourier series.[1] The result is a function which approximates $e^{-\sigma t}f(t)$ which when multiplied by $e^{\sigma t}$ gives the desired time function $f(t)$.

Figure 4-9 shows a time plot produced by BELTIP (Bell Electronic Laplace Transform Inversion Program).[2]

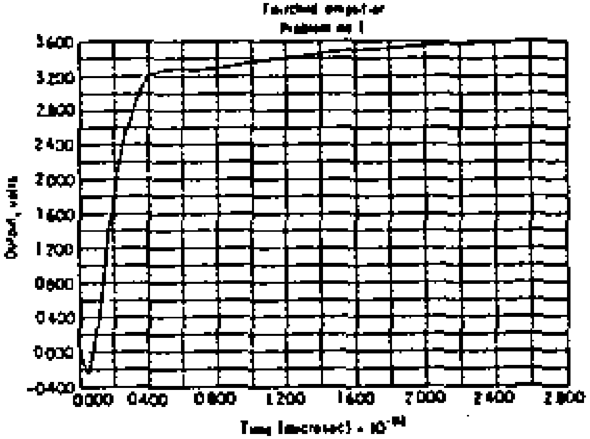


Fig. 4-9 Time response of an amplifier to a step.

When a circuit is being analyzed for time domain response, it is necessary to be able to obtain the frequency response for all the frequencies for which its magnitude is not negligible and it may in some cases be necessary (depending on the location of the singularities) to be able to obtain such frequency information not at the $j\omega$ axis but on the complex plane. Thus, during fitting of standard functions these facts should be borne in mind. It has been noted, for example, that unless the real and imaginary part satisfies the Hilbert transform condition [14], the resultant computations yield highly anticipatory circuits which respond before being excited.

4-5 ANALYSIS OF CIRCUITS USING THE INDEFINITE TRANSFER MATRIX

Given a $2n$-terminal network, the indefinite transfer matrix E is defined by the following equations

$$\left[\begin{array}{c} V_1 \\ \hline I_1 \end{array}\right] = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]\left[\begin{array}{c} V_2 \\ \hline -I_2 \end{array}\right], \quad E = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \qquad (4\text{-}20)$$

where V_1, I_1, V_2, I_2 are n entry columns corresponding to the currents and voltages shown in Fig. 4-11 in which the ports $1, 2, \ldots, n$ are considered input ports and the ports $n+1, n+2, \ldots, 2n$ are considered output ports. The matrices A, B, C, D are $n \times n$ matrices. (Their names result from the fact that they are extensions to $2n$-ports of the familiar A, B, C, D parameters of two-ports.) Like the indefinite admittance matrix, all ports are defined with a common floating terminal. For this reason the port condition is never lost with arbitrary interconnections. It is not difficult to go from the indefinite admittance matrix to the indefinite transfer matrix and vice versa. Let us partition the indefinite admittance matrix of a $2n$-terminal network as follows

$$Y = \left[\begin{array}{c|c} Y_{11} & Y_{12} \\ \hline Y_{21} & Y_{22} \end{array}\right] \qquad (4\text{-}21)$$

The matrices $Y_{11}, Y_{12}, Y_{21}, Y_{22}$ are $n \times n$ matrices. The variables are ordered so that Y_{11} correspond to V_1 and I_1 of Eq. (4-20). The equations relating the Y of Eq. (4-21) with A, B, C, D of Eq. (4-20) are

$$Y_{11} = DB^{-1}, \; Y_{21} = C - DB^{-1}A, \; Y_{21} = -B^{-1}, \; Y_{22} = B^{-1}A \qquad (4\text{-}22)$$

[1] The Cooley-Tukey algorithm requires N to be a power of 2. Thus the discussion is intended as a guide towards understanding the nature of errors of the approximations.

[2] BELTIP was developed by M. Silverberg of Bell Telephone Laboratories.

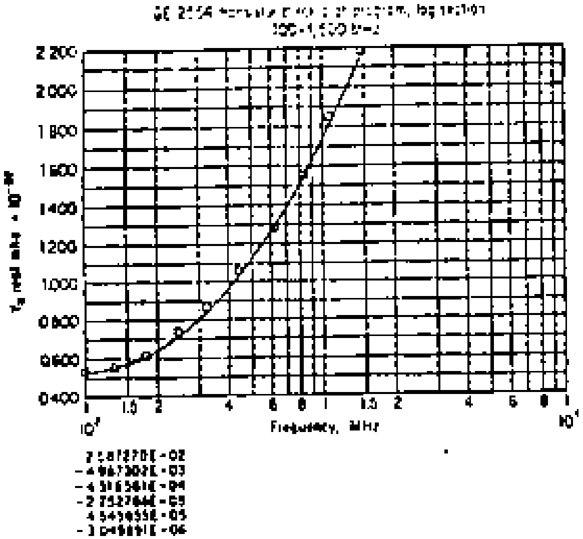


Fig. 4-10 Polynomial fit to Re y_{11} for GE-2554 transistor.

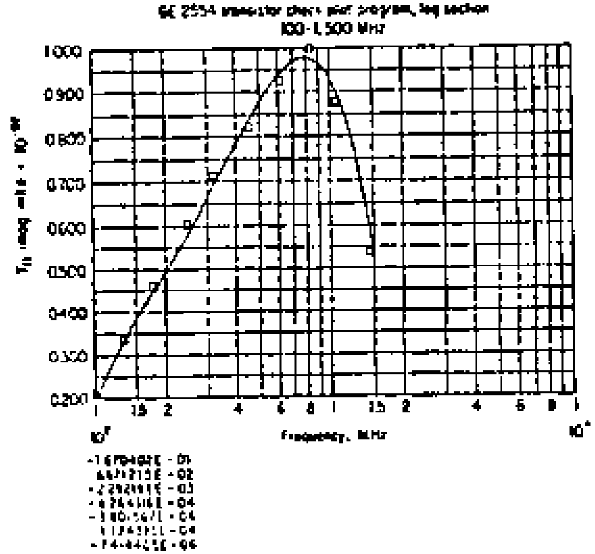


Fig. 4-11 Polynomial fit to Im y_{11} for GE-2554 transistor.

$$A = -Y_{21}^{-1}Y_{22},\quad B = -Y_{21}^{-1},\quad C = Y_{12} - Y_{11}Y_{21}^{-1}Y_{22},\quad D = -Y_{11}Y_{21}^{-1} \tag{4-23}$$

The most useful property of the E matrix is that if several 2n-terminal networks are connected in cascade the E matrix of the network is equal to the product of the individual E matrices of the cascaded sections. Because all the ports have a common terminal the port condition is never lost and no ideal transformers or tests for circulating currents are necessary [12].

In the analysis of electric circuits with a digital computer one often sacrifices computational efficiency to gain generality when using a "general purpose analysis program." When analysis and optimization programs are coupled, or in the making of statistical variability studies, it may be necessary to use the analysis program hundreds of times before the design process is over; thus the computational efficiency becomes increasingly important for such cases.

To obtain absolute maximum computational efficiency it would be necessary to write specific programs (perhaps in basic languages) for each circuit to be analyzed. Each program would be tailored to the peculiarities of the circuit in order to avoid the performance of any unnecessary operations. This is quite laborious, especially for large circuits. If many units of a particular circuit are going to be manufactured and hence exhaustive computer studies are going to be made, the writing of particular programs for circuits may be economically justified. If, on the other hand, many circuits which are members of a family are to be manufactured, although no particular member has a large volume of production, then it is better to write a slightly more general program that is capable of analyzing the whole family and would still be more efficient than a general purpose analysis program.

We will exhibit a family of transmission circuits for which extensive Monte Carlo statistical variability studies were required. It is possible to analyze the family of circuits with the indefinite transfer matrix and thereby gain enormous computational advantage over methods such as the indefinite admittance matrix which require matrix inversion.

Many four-ports which are of interest in radar and other communication systems fall into the class shown schematically in Fig. 4-12. The boxes labeled $N_1, N_2, \ldots, N_n$ represent four-terminal circuits plus a "ground." A port is made from each terminal to ground. Each four-port may be characterized by a 4 × 4 transmission matrix E each of which is partitioned into A, B, C, D matrices which are 2 × 2. All matrices have entries which are complex numbers and depend on frequency.

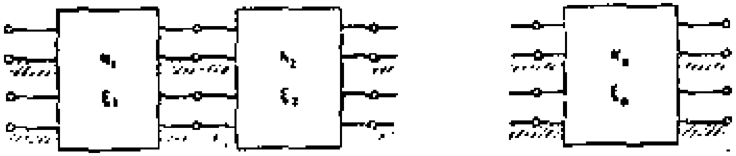


Fig. 4-12 Cascaded four-ports which are of interest in some communication systems.

To obtain the total E matrix of the n cascaded four-ports, the individual E matrices of the sections are multiplied in the order in which they appear. E_T denotes the E matrix of the complete network and E_k the E matrix of the kth section

$$E_T = E_1 \cdot E_2 \ldots E_k \ldots E_n \cdot \tag{4-24}$$

The first and last sections may correspond to the driving and load networks.

Each of the subnetworks of Fig. 4-12 may have an arbitrary internal structure. The characterization of each subnetwork may be done theoretically or experimentally. Curve fitting methods and preanalysis of subcircuits are also applicable here. Certain configurations which have application in the realization of directional couplers will be considered in detail. Figure 4-13 shows two identical lossless coupled transmission lines each with a capacitance to ground per unit length C, capacitance between lines per unit length C_M, self-inductance per unit length L_{11}, and mutual inductance per unit length L_{12}. The length of the lines is l. The E matrix of the four-port of Fig. 4-13 is

$$E = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \left[\begin{array}{c|c} \cosh \Gamma \cdot l & (\sinh \Gamma \cdot l) Z_0 \\ \hline Y_0 \sinh \Gamma \cdot l & \cosh \Gamma \cdot l \end{array}\right] \tag{4-25}$$

where Γ, Z_0, and Y_0 are 2×2 matrices given by

$$\Gamma = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} j\omega \sqrt{(L_{11} + L_{12})C} + \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} j\omega \sqrt{(L_{11} - L_{12})(C + 2C_M)} \tag{4-26}$$

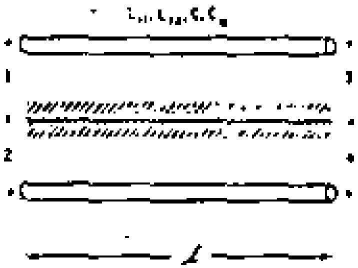


Fig. 4-13 Pair of lossless coupled lines.

$$Z_0 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \sqrt{\frac{L_{11} + L_{12}}{C}} + \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \sqrt{\frac{L_{11} - L_{12}}{C + 2C_M}} \tag{4-27}$$

$$Y_0 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \sqrt{\frac{C}{L_{11} + L_{12}}} + \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \sqrt{\frac{C + 2C_M}{L_{11} - L_{12}}} \tag{4-28}$$

The functions $\Gamma \cdot l$ and $\sinh \Gamma \cdot l$ of the matrix Γ are the following matrices

$$\cosh \Gamma \cdot l = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \cosh \left\{ \sqrt{(L_{11} + L_{12})C} \;\; j\omega l \right\}$$

$$+ \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \cosh \left\{ \sqrt{(L_{11} - L_{12})(C + 2C_M)} \;\; j\omega l \right\} \tag{4-29}$$

$$\sinh \Gamma \cdot l = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \sinh \left\{ \sqrt{(L_{11} - L_{12})C}\; j\omega l \right\}$$

$$+ \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \sinh \left\{ \sqrt{(L_{11} - L_{12})(C + 2C_M)}\; j\omega l \right\} \tag{4-30}$$

Besides the distributed section considered above, two lumped subnetworks will be considered.

The first lumped subnetwork to be considered is the one shown in Fig. 4-14. The E matrix of the circuit of Fig. 4-14 is

$$E = \left[\begin{array}{c|c} 1 & Z \\ \hline 0 & 1 \end{array}\right] \tag{4-31}$$

where 0 is the 2 × 2 zero matrix, 1 is the 2 × 2 unit matrix, and Z is

$$Z = \begin{bmatrix} j\omega L_{11} + R_{11} & j\omega L_{12} \\ j\omega L_{12} & j\omega L_{11} + R_{22} \end{bmatrix} \tag{4-32}$$

The second type of lumped subnetwork to be considered is shown in Fig. 4-15 where Y_a, Y_b, and Y_c are the admittances of the boxes. The E matrix of the subnetwork of Fig. 4-15 is

$$E = \left[\begin{array}{c|c} 1 & 0 \\ \hline Y & 1 \end{array}\right] \tag{4-33}$$

where Y is

$$Y = \begin{bmatrix} Y_a + Y_c & -Y_c \\ -Y_c & Y_b + Y_c \end{bmatrix} \tag{4-34}$$

For the particular cases of Fig. 4-16 and 4-17, the corresponding Y matrices are

$$Y = \begin{bmatrix} (C_1 + C_3)j\omega + G_1 + G_3 & -C_3 j\omega - G_3 \\ -C_3 j\omega - G_3 & (C_2 + C_3)j\omega + G_2 + G_3 \end{bmatrix} \tag{4-35}$$

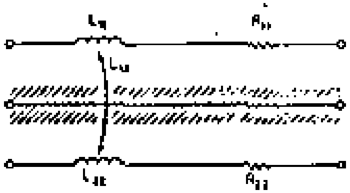


Fig. 4-14 Pair of lossy coupled coils forming a section of the cascade.

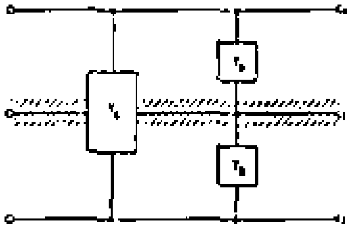


Fig. 4-15 Transversal admittances forming a section of the cascade.

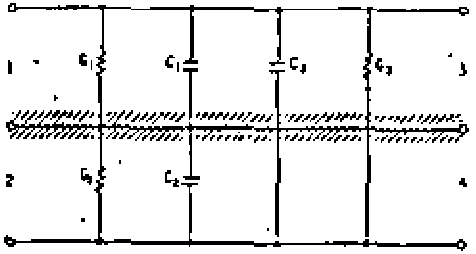


Fig. 4-16 Particular case of the section of Fig. 4-15.

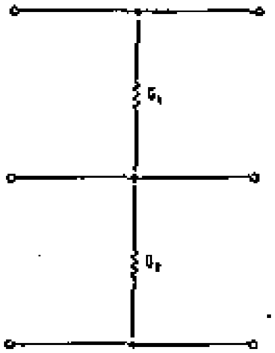


Fig. 4-17 Resistive loads considered as a particular case of Fig. 4-15.

$$Y = \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix}. \tag{4-36}$$

where G_i are conductances in mhos and C_i are capacitances in farads.

Once the E matrix of the complete circuit or of a portion of it is known the voltage ratios are calculated very simply in terms of the entries of E. Partitioning the E matrix into 2 × 2 matrices A, B, C, D and assuming that a four-port is driven at port 1, we have

$$\frac{V_3}{V_1} = \frac{C_{22}}{C_{22}A_{11} - C_{21}A_{12}} \tag{4-37}$$

$$\frac{V_4}{V_1} = \frac{-C_{21}}{C_{22}A_{11} - C_{21}A_{12}} \tag{4-38}$$

$$\frac{V_2}{V_1} = \frac{C_{22}A_{21} - C_{21}A_{22}}{C_{22}A_{11} - C_{21}A_{12}} \tag{4-39}$$

The driving-point impedance at port 1 is

$$z = \frac{A_{11}C_{22} - A_{12}C_{21}}{C_{11}C_{22} - C_{12}C_{21}}. \tag{4-40}$$

The reflection coefficient Γ_R when port 1 is connected as a load to a lossless transmission line of characteristic impedance z_0 is

$$\Gamma_R = \frac{z - z_0}{z + z_0}. \tag{4-41}$$

Γ_R will in general be complex. Denoting its magnitude with $|\Gamma_R|$, the voltage standing wave ratio is given by

$$\text{VSWR} = \frac{1 + |\Gamma_R|}{1 - |\Gamma_R|}. \tag{4-42}$$

With these tools, the analysis of a circuit containing lumped and distributed parameters, such as the one shown in Fig. 4-18, is easily done by matrix multiplication and application of Eqs. (4-37) to (4-42).

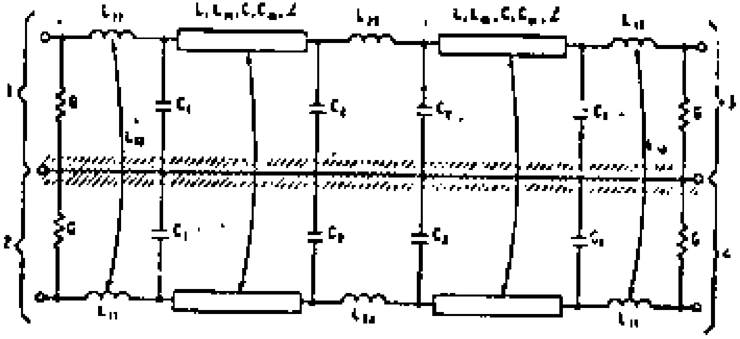
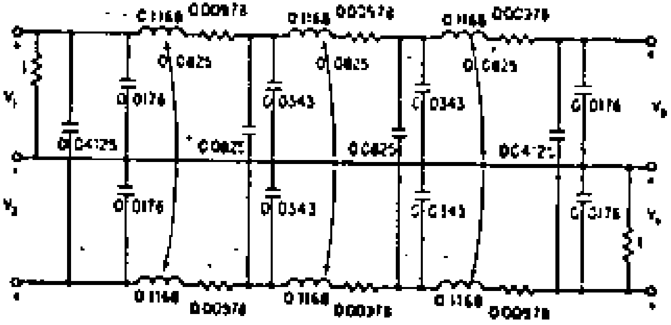


Fig. 4-18 Typical circuit that can be analyzed by COUPLER.

Suppose that a circuit of the form of Fig. 4-12 is made up of sections each of which has the internal structure of the circuit shown in Fig. 4-19. Suppose the complete circuit has 10 sections. Each section has 15 nodes. The complete circuit will have 131 nodes (some of nodes are shared by two sections). A program such as ECAP would not be able to accommodate such a circuit and even



Notes.

Unless otherwise specified
Resistance values are in ohms
Capacitance values are in farads
Inductance values are in henries

Fig. 4-19 Lumped circuit forming a section of the circuit of Fig. 4-12.

If it were, the amount of computation to solve the circuit would be considerable. Using an algorithm such as Gauss' elimination procedure, it takes roughly $(1/3)n^3$ complex multiplication-additions to solve the system of equations at each frequency. Here n stands for the order of the matrix which corresponds to the number of nodes minus one. Thus the circuit we are considering would require roughly $1/3\,(133)^3 = 0.8$ million multiplication-additions to solve for one frequency.

The nodal admittance matrix of the network we are considering is quite sparce (has many zeros); there are more efficient methods of solving it if its sparsity is exploited.

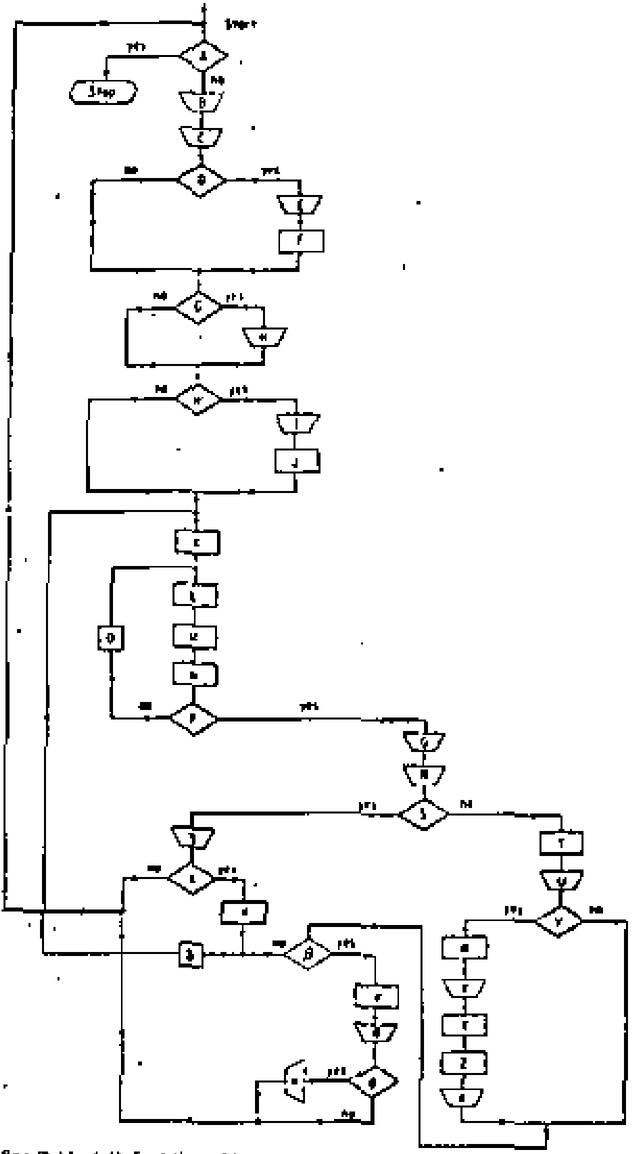
If the transfer matrix method is used and the network is considered as a cascade connection of four-ports of the types discussed above, the number of complex multiplication-additions per frequency is 5760, that is, only 1/130 as many.[1] Some Monte Carlo runs in which each run involved 500 circuits calculated at 10 frequencies were performed. The reader can estimate the enormous savings that were realized by using this method instead of the nodal matrix method. The principal reason for the savings is that with cascade configurations the number of operations to solve the circuit is a linear function of the number of nodes as opposed to a cubic function for general configurations and using the nodal matrix. Even if the transfer matrix were not used, the problem could have been solved much more efficiently by considering each section as a subnetwork, analyzing each subnetwork, and finally connecting all the sections obtaining an admittance matrix of order 20 which is considerably smaller than 133 [25].

4-6 BRIEF DESCRIPTION OF COPLER: A FOUR-PORT SIMULATOR

COPLER is a computer program written at Bell Telephone Laboratories [26] for simulating a class of four-ports. (A flowchart of COPLER is given in Table 4-4(a), and a description of this block appears in Table 4-4(b).) The program has been used mainly to predict the behavior of lumped and/or distributed directional couplers [27]. The program calculates the total E matrix of a circuit of the form of Fig. 4-12 given the values of the lumped R, L, C, M elements, values per unit length of coupled lines, or special

[1] Assuming that all the elements have different values. If the cascaded sections are equal, the savings are even larger.

Table 4-4a. Flowchart of COPLER Program*



*See Table 4-4b for titles of boxes.

Table 4-6. Titles of Boxes for COPLER Program

A	Is data exhausted?
B	Read titles, configuration code, and option flags.
C	Read element values for standard subcircuits.
D	Is statistical study desired?
E	Read statistics of parameters and number of trials.
F	Store nominal parameters.
G	Is optimization desired?
H	Read desired curves and type of criterion of performance.
H'	Are there special black boxes?
I	Read position of special black boxes.
J	Call subroutines for computing E matrices of special black boxes.
K	Initialize frequency loop.
L	Compute E matrices of standard subcircuits.
M	Multiply E matrices of complete cascade.
N	Calculate voltage ratios and VSWR.
O	Take next frequency.
P	Is this the last frequency?
Q	Print table of voltage ratios and VSWR versus frequency.
R	Print present values of parameters and trial.
S	Is this the nominal trial?
T	Calculate and store performance index.
U	Print performance index and trial number.
V	Is optimization desired?
W	Calculate criterion of performance.
X	Print criterion of performance and trial number.
Y	Store best criterion and trial number so far.
Z	Store table of voltage ratios and VSWR of best trial so far.
α	Print best criterion of performance and trial number so far.
β	Have enough statistical trials been calculated?
γ	Calculate moments of performance index.
δ	Print moments of performance index.
ξ	Make random variation of parameters around nominal values.
η	Plot the voltage ratios and VSWR.
λ	Is statistical study or optimization desired?
κ	Store nominal voltage ratios and VSWR.
ϕ	Is optimization desired?
π	Plot voltage ratios and VSWR of nominal and best trials.

subroutines defining arbitrary analytical models of four-ports in the frequency domain. A code vector indicates to the program how the elements are connected. The program also performs a statistical tolerance analysis using a Monte Carlo method and taking advantage of the tolerance analysis optimizes the network through a random walk. In order to do the statistical variability study COPLER has several pseudo random number generating subroutines. Taking advantage of the fact that the circuit is analyzed for the statistical variability study, optimum parameters for the system may be found by calculating a ''criterion of performance'' for each variation of the circuit and using the values of the parameters that give the best criterion of performance.

The output of COPLER under various options includes: printing or printing and plotting the phase and magnitude of the voltage ratios given by Eqs. (4-37)-(4-39) and the VSWR given by Eq. (4-42); printing of a ''performance index'' for each trial, and the first and second moments of the ''performance index'' of a set of random trials around a nominal network (the random parameters have specified statistical distributions). For each trial the program also prints the complete set of parameters of the circuit analyzed. Given a desired frequency behavior the program calculates a ''criterion of performance'' and prints the set of parameters with the best criterion of performance. There are several criteria of performance available and the user may define his own by writing his own subroutine.

The input to the program is user-oriented. The data is fed in formatless form using the NAMELIST feature of FORTRAN IV. If many elements are repeated there are simple ways of feeding them in. The circuit of Fig. 4-19 with the initial normalized values shown was analyzed with the aid of COPLER. Figure 4-20 shows the magnitude of V_3/V and Fig. 4-21 the VSWR versus normalized frequency of three random variations of the circuit of Fig. 4-19. Figure 4-22 shows a plot produced by the SC-4020. The plot corresponds to the circuit of Fig. 4-18 with the following normalized values: $L_{11} = L_{22} = 0.03533$ henries, $L_{12} = 0.025$ henries, $C_1 = C_2 = 0.01033$ farads, $G = 1$ mho, $L = 0.318$ henries/meter, $L_m = 0.225$ henries/meter, $C = 0.093$ farads/meter, $C_m = 0.225$ farads/meter,

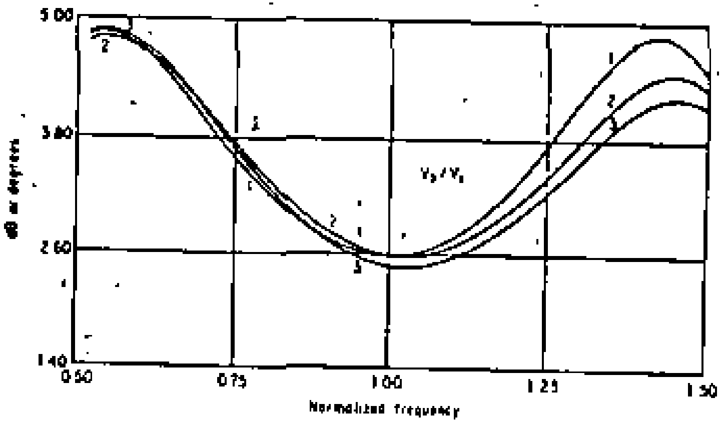


Fig. 4-20 Graph of $|V_3/V_1|$ of three random variations of the circuit of Fig. 4-19.

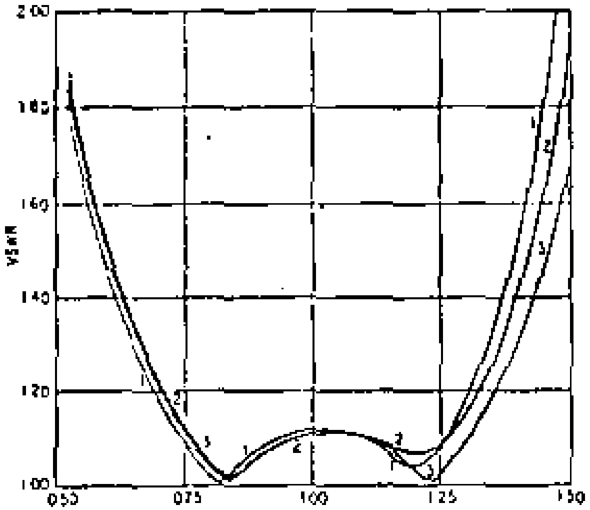


Fig. 4-21 Graph of the VSWR of three random variations of the circuit of Fig. 4-18.

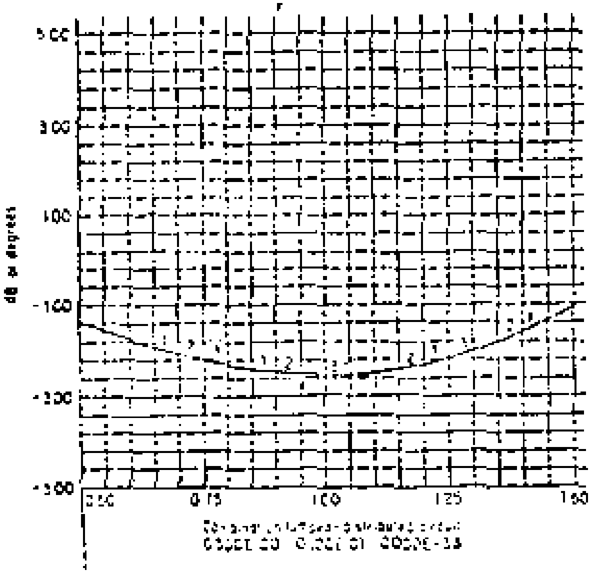


Fig. 4-22 Graph of V_3/V_1 of a [illegible] circuit of the form of the [illegible] Fig. 4-[illegible].

and l = 0.33 meter. The plot shows the magnitude in dB of the voltage ratio V_3/V_1 versus normalized frequency.

4-7 OUTLOOK

In this chapter the analysis of linear circuits consisting of interconnected black boxes was treated. (We consider conventional lumped elements as particular cases of black boxes.) Three programs, BELNAP, BELTIP, and COPLER, were briefly described and examples of their output were given. The black-box approach to the analysis of integrated circuits offers several advantages not available with the existing general-purpose computer programs. Among these advantages are: analysis of large circuits by pieces, modeling of transistors and integrated circuits by experimental measurements, and inclusion of distributed elements. The analysis of circuits by pieces not only alleviates the problem of limited memory, but also gains considerable computational efficiency since it takes advantage of the sparsity of the matrices of the networks.

REFERENCES

1. 1620 Electronic Circuit Analysis Program (ECAP), "Application Program 1620-EE-02X," Data Processing Division, IBM Corporation, White Plains, N. Y.
2. Malmberg, A. F., F. L. Cornwell, and F. N. Hofer: NET-1 Network Analysis Program, *Report LA-3119*, Los Alamos Scientific Laboratory, Los Alamos, N. M., 1964.
3. Happ, W. W.: NASAP: Present Capabilities of a Maintained Program, *Proc. Conf. Analysis of Circuits with Digital Computer*, Nat. Univ. Mexico, Mexico City, June, 1967.
4. Dickhaut, R. H.: CIRCUS, a Digital Computer Program for Transient Analysis of Electronic Circuits, Computer-Aided Circuit Design Seminar, M.I.T., Cambridge, Mass., seminar proc. published by NASA/ERC, pp. 4-143, April, 1967.
5. Sedore, S. R.: SCEPTRE: A Second Generation Transient Analysis Program, Computer-aided Circuit Design Seminar, M.I.T., Cambridge, Mass., seminar proc. published by NASA/ERC, pp. 57-61, April, 1967.
6. Automated Digital Computer Program for Determining Responses of Electronic Systems to Transient Nuclear Radiation, vol. II, File No. 64-521-5, IBM Corporation, Owego, N. Y., July, 1964.

7. Searle, C. L., et al.: "Elementary Circuit Properties of Transistors," pp. 102–120, John Wiley & Sons, Inc., New York, 1964.
8. Carlin, H. J.: Network Theory without Circuit Elements, *Proc. IEEE*, vol. 55, no. 4, pp. 482–496, April, 1967.
9. Davis, P. J.: "Interpolation and Approximation," Blaisdell Publishing Co., Waltham, Mass., 1963.
10. Hastings, C., Jr.: "Approximation for Digital Computers," Princeton University Press, Princeton, N. J., 1955.
11. Fleischer, P. E.: Optimization Techniques in System Design, in F. F. Kuo and J. F. Kaiser (eds.), "System Analysis by Digital Computer," John Wiley & Sons, Inc., New York, 1966.
12. Huelsman, L.: "Circuits, Matrices, and Linear Vector Spaces," pp. 76–91, McGraw-Hill Book Company, New York, 1963.
13. Seshu, S., and M. B. Reed: "Linear Graphs and Electrical Networks," pp. 19–154, Addison-Wesley Publishing Co., Inc., Reading, Mass., 1961.
14. Guillemin, E. A.: "Theory of Linear Physical Systems," pp. 144–158, John Wiley & Sons, Inc., New York, 1963.
15. So, H. C.: Analysis and Iterative Design of Networks Using On-Line Simulation, in F. F. Kuo and J. F. Kaiser (eds.), "System Analysis by Digital Computer," pp. 34–58, John Wiley & Sons, Inc., New York, 1966.
16. Kron, G.: A Set of Principles to Interconnect the Solution of Physical Systems, *J. Appl. Phys.*, vol. 24, pp. 965–980, 1953.
17. Murray-Lasso, M. A.: The Use of the Indefinite Admittance Matrix for Computer Analysis of Circuits, *Proc. Conf. Analysis of Circuits with Digital Computer*, Nat. Univ. Mexico, Mexico City, June, 1967.
18. Davies, L. A.: BELNAP – A Computer Program for the AC Analysis of Linear Active and Passive Networks, unpublished work, Bell Telephone Laboratories, May, 1967.
19. *Ibid.*, BELNAP II – The Second Generation of a Computer Program for the AC Analysis of Linear Active and Passive Networks, unpublished work, Bell Telephone Laboratories, Sept., 1967.
20. Kurokawa, K.: Design Theory of Balanced Amplifiers, *Bell System Tech. J.*, vol. 44, no. 8, pp. 1675–1698, Oct., 1965.
21. Eisele, K. M., R. S. Engelbrecht, and K. Kurokawa: Balanced Transistor Amplifiers for Precise Wideband Microwave Applications, *Dig. Tech. Pap.*, Intern. Solid-State Circuits Conf., Philadelphia, pp. 18–19, Feb., 1965.

22. Murray-Lasso, M. A.: "Report on a Theoretical Investigation on Multiple Coupled Transmission Lines," unpublished work, Bell Telephone Laboratories, Sept., 1965.
23. Silverberg, M.: "Numerical Solution of Distributed Networks Containing Nonlinear Elements," doctoral dissertation, Department of Electrical Engineering, Columbia Univ., New York, 1967.
24. Cooley, J. W., and J. W. Tukey: An Algorithm for the Machine Calculation of Complex Fourier Series, *Math. of Computation*, vol. 19, pp. 297–301, April, 1965.
25. Murray-Lasso, M. A.: "Analisis y Optimizacion de Circuitos de Comunicaciones con Computadora" Memoria Segundo Congreso Panamericano de Ingenieria Mecanica, Electrica y de Ramas Afines, Caracas, Venezuela, Sept., 1967.
26. *Ibid.*, A Digital Computer Simulation of a Class of Lumped and/or Distributed Four-Ports, *Proc. 1967 ACM-SHARE Design Automation Workshop*, Los Angeles, June, 1967.
27. *Ibid.*, Unified Matrix Theory of Lumped and Distributed Directional Couplers, *Bell System Tech. J.*, vol. 47, pp. 39–71, Jan., 1968.

A GENERAL TRANSFORMATION WITH APPLICATIONS TO CIRCUIT THEORY

By
M. A. MURRAY-LASSO

FRANKLIN INSTITUTE OF THE STATE OF PENNSYLVANIA
1824

Reprinted from JOURNAL OF THE FRANKLIN INSTITUTE
VOL. 285, NO. 2, FEBRUARY, 1968

Reprinted from JOURNAL OF THE FRANKLIN INSTITUTE, Vol. 285, No. 2, February, 1968
Printed in U. S. A.

A General Transformation With Applications to Circuit Theory

by M. A. MURRAY-LASSO
Bell Telephone Laboratories, Inc.
Whippany, New Jersey

ABSTRACT: *A general functional transformation is introduced and some applications to circuit theory are given. The transformation is most useful in the area of active, nonbilateral circuits. This functional transformation introduces new voltages and currents which are linear combinations of the original voltages and currents plus their derivatives, integrals or more complicated implicit integrodifferential relationships. The transformation applied to the currents may be completely unconnected to the transformation applied to the voltages.*

The transformation may be used for obtaining equivalent circuits of some driving point impedance. Another possibility is obtaining equivalent circuits of some transfer impedance but with different driving point impedances. The transformation handles with equal ease bilateral and non-bilateral circuits as well as circuits containing controlled sources. It may also handle circuits containing non-real elements. Simple numerical examples are given in active, non-bilateral, and passive synthesis, as well as in analysis.

Introduction

The transformation we introduce is most useful in the area of active, nonbilateral circuits, but may also be applied to passivo, bilateral ones. The transformation is initially formulated in abstract terms, so that it can be applied to different situations. Both the dependent and independent variables are transformed, leaving, for instance, the possibility of applying one transformation to the voltages of a circuit and a completely different one to the currents. The form of the functional equation relating independent and dependent variables is left invariant. This determines the transformation of the impedance operator. After the transformation the new variables are interpreted using the same interpretation employed for the original variables.

Great liberty is allowed for the transformation operators so that when applied to the generation of equivalent circuits the transformation may go from a bilateral network to a non-bilateral one and *vice versa*. It may also leave a transfer impedance invariant while all the driving point impedances change. These problems cannot be handled with the congruent transformation which has received a good deal of attention from researchers concerned with equivalent networks through linear transformations.

Most of the applications presented in the paper are in the field of synthesis. However, to demonstrate the versatility of the functional aspect of the transformation an application to analysis is also given.

Transformation Equations

Consider the functional equation

$$y = Fx \tag{1}$$

representing the mapping by the *operator F* of the *points* (or vectors) x in the linear vector space X—called the *domain* of F—onto the *points* y in the linear vector space Y—called the *range* of F. The operator F is linear, that is, $F(\alpha x + \beta y) = \alpha Fx + \beta Fy$; α, β complex numbers.

Now make the following non-singular transformations:

$$x = P\hat{x}, \tag{2}$$

$$y = Q\hat{y}. \tag{3}$$

The points $\hat{x}$ and $\hat{y}$ and the operators P, Q in Eqs. (2) and (3) have a meaning analogous to the entities in Eq. (1). Substitution of Eqs. (2) and (3) into (1) gives $Q\hat{y} = FP\hat{x}$. Since the operation represented by Q is non-singular

$$\hat{y} = Q^{-1}FP\hat{x}, \tag{4}$$

where Q^{-1} is defined in such a way that $Q^{-1}Q\hat{y} = \hat{y}$, $QQ^{-1}y = y$, $\hat{y}$ being in the domain of Q and y being in its range.

If a new operator $\hat{F}$ is defined according to

$$\hat{F} = Q^{-1}FP, \tag{5}$$

Eq. (4) may be written

$$\hat{y} = \hat{F}\hat{x}. \tag{6}$$

The operator $\hat{F}$ given by Eq. (5) maps the points $\hat{x}$ of the space $\hat{X}$ onto the points $\hat{y}$ of the space $\hat{Y}$. The spaces to be considered in this paper are of various natures, for this reason the transformation is formulated in abstract terms borrowing the nomenclature of functional analysis. Equations (2) and (3) may be interpreted in two different ways:

a) Invariant-Object Transformations.

Consider x and $\hat{x}$ as giving the coordinates of *the same object* (point) measured in two *different frames of reference*. The operator P gives a rule for translating from one frame to the other, or

b) Invariant-Coordinate-Frame Transformations.

Consider x and $\hat{x}$ as *different objects*. The quantities x and $\hat{x}$ being the coordinates of the different points in the *same reference frame*. The operator P gives the rule for going from one point to the other. We shall consider applications of both interpretations.

A functional equation of the form of Eq. (1) which arises in circuit theory is

$$i = \mathcal{Y}v. \tag{7}$$

The voltage v and the current i are functions of the time t. The operational admittance $\mathcal{Y}$ or its inverse are often given in the form of a differential equation or in the form of the kernel of a superposition integral. The functions v and i may be considered as points in a nondenumerably infinite dimensional linear vector space and $\mathcal{Y}$ as an operator which maps the points v of the space V onto the points i of the space I. Several currents may be of interest in a circuit; hence, in Eq. (7), v and i may represent columns

$$v = \begin{bmatrix} v_1(t) \\ v_2(t) \\ \cdot \\ \cdot \\ \cdot \\ v_N(t) \end{bmatrix}, \qquad i = \begin{bmatrix} i_1(t) \\ i_2(t) \\ \cdot \\ \cdot \\ \cdot \\ i_N(t) \end{bmatrix}, \tag{8}$$

and $\mathcal{Y}$ may be specified by a set of simultaneous differential equations or as a matrix of kernels of a superposition integral. In this case v and i may still be considered as points in an infinite dimensional space and $\mathcal{Y}$ as the mapping connecting these points. With these preliminary concepts some applications of the transformation given by Eq. (4) are considered.

Application to the Synthesis of Equivalent Circuits

The circuit designer is interested in equivalent networks because some forms of circuits may be more suitable than others as regards cost, sensitivity, spread of element values, absorbing stray values, etc. Some equivalent networks are trivial variations of each other (for example, replacing a resistor by two in parallel). Others may contain elements with negative values, or non-reciprocal elements, or very different topological structures.

One way of producing equivalent networks is through linear transformations. A powerful method was proposed by Cauer (1) in 1929. Writing loop equations

in matrix form, the loop currents are transformed keeping the energy functions invariant. This results in a congruent transformation of the loop impedance matrix. If the current in a particular loop is held invariant by suitable restrictions on the transformation matrix it follows from Lagrange's equations that the impedance of that loop must remain the same. The transformation may also be applied to a node-to-datum admittance matrix. In the Cauer transformation the elements of the transformation matrix are real constants. The transformation has received the attention of numerous investigators (2, 3, 4, 5, 8, 9, 12). Cauer showed that the positive definite or semidefinite character of the parameter matrices is both necessary and sufficient for realizability with positive circuit elements as long as ideal transformers are allowed. This property is preserved by a congruent transformation. This implies that if the node-to-datum admittance matrix of a passive network is Cauer transformed, the resultant network will be passive as far as the node-to-datum ports are concerned, even though some elements may be active. However, it is clear that given two circuits, one of which is passive and the other active, as seen from a set of node-to-datum ports, both may have the same driving point impedance at one port. But the Cauer transformation cannot produce the active circuit from the passive one. Furthermore, with the Cauer transformation the final circuits will be bilateral if the original circuit was bilateral. This is a consequence of the fact that a congruent transformation does not destroy the symmetry of a matrix. For these reasons, in synthesis in which passive, active, bilateral and nonbilateral networks are of interest congruent transformations have restricted applicability.

In this paper the circuit elements assumed to be at the disposal of the designer are: constant resistors, inductors, capacitors (all both positive and negative), gyrators and controlled sources. For these elements the complex frequency domain representation is convenient. However if the transformation is applied to time varying circuits it might be more convenient to work in the time domain.

The attitude of allowing for greater generality is further motivated by the fact that with the recent developments in monolithic and other integrated circuits, more than before, the circuit designer is interested in equivalent circuits containing active non-bilateral elements. The microminiature transistor technology has advanced to the point that often it is more desirable to introduce transistors than passive elements. This situation implies that methods which a few years ago would not have received the attention of circuit designers have become relevant given the present state of the art.

Transformation for Invariant Driving Point Impedance

Suppose a circuit is analyzed on the node basis using ground as the datum node. The equilibrium equations may be written symbolically in the time domain as in Eq. (7); v and i are the vectors given by Eqs. (8) and $\mathcal{Y}$ represents the differential equations linking v and i. The complex frequency domain representation of the circuit is already an application of Eq. (4), if the variables of Eqs. (2), (3) and (4) are identified in the following way:

$$x = v(t), \qquad \hat{x} = V(s), \qquad P = E\int_{-\infty}^{\infty}[\cdot]e^{-st}\,dt, \tag{9}$$

$$y = i(t), \qquad \hat{y} = I(s), \qquad Q = E\int_{-\infty}^{\infty}[\cdot]e^{-st}\,dt. \tag{10}$$

(In the definition of the integral operators P and Q, E stands for the $N \times N$ unit matrix.) Equation (6) reads, for this case

$$I(s) = Y(s)V(s), \tag{11}$$

where

$$I(s) = \begin{bmatrix} I_1(s) \\ I_2(s) \\ \cdot \\ \cdot \\ I_N(s) \end{bmatrix}, \qquad V(s) = \begin{bmatrix} V_1(s) \\ V_2(s) \\ \cdot \\ \cdot \\ \cdot \\ V_N(s) \end{bmatrix}, \tag{12}$$

are the double-sided Laplace transforms of the current and voltage vectors; $Y(s)$ is an $N \times N$ matrix whose entries are, for RLC circuits, ratios of polynomials of the complex variable s of the Laplace transformation.

Equation (11) is an example of the invariant-object interpretation of a transformation, since the new variables represent the same currents and the same voltages in a different frame of reference known as the complex frequency domain. We assume that this transformation has been done and Eq. (11) is the starting point. No transformers or mutual inductances are considered to be present in the circuits discussed. In Eq. (12) V_j is the voltage of the jth node with respect to the datum; I_j is the sum of the currents injected into node j by the current sources; Y_{jj} is the sum of all the admittances connected to node j; Y_{jk} is the negative of the admittance connected between nodes j and k.

Consider a network with only one current source connected to the two terminals forming a port whose driving point impedance is of interest. Make the node into which the current is entering from the source node 1, and the other the datum. Hence,

$$I(s) = \begin{bmatrix} I_1(s) \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \tag{12'}$$

Now subject the voltage and current vectors to non-singular transformations, defined by

$$I(s) = A\hat{I}(s), \tag{13}$$

$$V(s) = B\hat{V}(s). \tag{14}$$

Let the operators A and B be $N \times N$ matrices whose entries are ratios of polynomials of s. If the form of the functional equation is kept invariant, Eq. (11) is transformed into

$$\hat{I} = \hat{Y}\hat{V}, \tag{15}$$

where

$$\hat{Y} = A^{-1}YB. \tag{16}$$

The transformation given by Eqs. (13) and (14) is an invariant-coordinate-frame transformation, that is, the new current and voltage vectors will be different physically. They will be measured in the same reference frame as the old currents and voltages, which means that after the transformation the interpretation of $\hat{I}$, $\hat{V}$, $\hat{Y}$ is based on the same criterion that was used for the interpretation of I, V, Y.

If it is desired to keep the driving point impedance between node 1 and the datum invariant, the operators A and B must be restricted somewhat. For the original circuit, the driving point impedance from node 1 to ground is

$$z = V_1/I_1. \tag{17}$$

In the new circuit the driving point impedance from node 1 to ground is

$$\hat{z} = \hat{V}_1/\hat{I}_1. \tag{18}$$

If Eqs. (18) and (17) are equated, the operators are to be restricted in such a way that

$$V_1/I_1 = \hat{V}_1/\hat{I}_1 \tag{19}$$

is satisfied. Hence, a possibility is to choose A and B such that

$$\hat{V}_1 = \beta V_1, \tag{20}$$

$$\hat{I}_1 = \beta I_1, \tag{21}$$

where β is an arbitrary ratio of polynomials of s. If the matrix B is of the form

$$B = \begin{bmatrix} 1/\beta & 0 & \cdots & 0 \\ B_{21} & B_{22} & \cdots & B_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ B_{N1} & B_{N2} & \cdots & B_{NN} \end{bmatrix} \tag{22}$$

and the matrix $A^{-1} = C$ is of the form

$$A^{-1} = C = \begin{bmatrix} \beta & C_{12} & \cdots & C_{1N} \\ C_{21} & C_{22} & \cdots & C_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ C_{N1} & C_{N2} & \cdots & C_{NN} \end{bmatrix}, \tag{23}$$

Eq. (19) will be satisfied and the transformed circuit will have the same driving point impedance between node 1 and the datum as the original circuit. The transformed circuit will, in general, contain controlled sources. The short-circuit admittance matrix $\hat{Y}$ of the new circuit is given by Eq. (16). For further clarification at this point we introduce a simple numerical example.

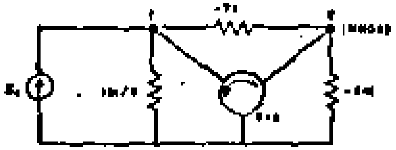


Fig. 1. Circuit corresponding to Y of Eq. 24.

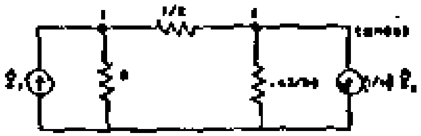


Fig. 2. Circuit corresponding to $\hat{Y}$ of Eq. 30 and $\hat{I}$ of Eq. 31.

Consider the circuit of Fig. 1. The driving point impedance from node 1 to the datum is one ohm. The short-circuit node-to-datum admittance matrix is

$$Y = \begin{bmatrix} -11/2 & 71 \\ 71 & -770 \end{bmatrix} + \begin{bmatrix} 0 & -6 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} -11/2 & 65 \\ 77 & -770 \end{bmatrix}. \tag{24}$$

By applying the transformation of Eq. (16) it is desired to obtain a *bilateral* circuit that has the same driving point impedance from node 1 to the datum. Using Eqs. (16), (22) and (23) yield

$$\hat{Y} = \begin{bmatrix} \beta & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} -11/2 & 65 \\ 77 & -770 \end{bmatrix} \begin{bmatrix} 1/\beta & 0 \\ B_{21} & B_{22} \end{bmatrix}. \tag{25}$$

Performing the multiplications indicated

$$\hat{Y} = \begin{bmatrix} -\frac{11}{2} + 77(C_{12}/\beta) + (65\beta - 770C_{12})B_{21} & (65\beta - 770C_{12})B_{22} \\ (-\frac{11}{2}C_{21} + 77C_{22})(1/\beta) + (65C_{21} - 770C_{22})B_{21} & (65C_{21} - 770C_{22})B_{22} \end{bmatrix}. \tag{26}$$

Since it is desired to obtain a bilateral circuit, the matrix $\hat{Y}$ must be symmetric.

This imposes the following constraint:

$$\left(-\tfrac{1}{7}C_{11} + 77C_{12}\right)(1/\beta) + (65C_{12} - 770C_{22})B_{12} = (65\beta - 770C_{21})B_{22}. \quad (27)$$

from which one of the arbitrary parameters β, C_{12}, C_{21}, C_{22}, B_{12}, B_{22} may be eliminated. All the rest of the parameters remain arbitrary. The following choice

$$\beta = 1, \qquad C_{12} = \tfrac{1}{7}, \qquad C_{21} = \tfrac{1}{11}, \qquad C_{22} = 0, \qquad B_{12} = 0, \quad (28)$$

gives, from Eq. (27)

$$B_{22} = \tfrac{1}{11}, \quad (29)$$

and yields the following $\hat{P}$ matrix from Eq. (26):

$$\hat{P} = \begin{bmatrix} 11/2 & -1/2 \\ -1/2 & 13/198 \end{bmatrix}. \quad (30)$$

According to the choice made

$$\begin{bmatrix} \hat{I}_1 \\ \hat{I}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1/7 \\ 1/11 & 0 \end{bmatrix} \begin{bmatrix} I_1 \\ 0 \end{bmatrix}.$$

That is,

$$\begin{aligned} \hat{I}_1 &= I_1, \\ \hat{I}_2 &= \tfrac{1}{11}I_1 = \tfrac{1}{11}\hat{I}_1. \end{aligned} \quad (31)$$

Hence, a controlled source of the value shown by Eq. (31) must be connected from node 2 to the datum. The circuit realizing $\hat{P}$ of Eq. (30) and with the controlled current source indicated by Eq. (31) appears in Fig. 2. Analysis of the circuit of Fig. 2 shows that the driving point impedance from node 1 to the datum is one ohm, the same as that of Fig. 1. With this transformation the gyrator was eliminated at the expense of having a controlled source. In the next section we show how to avoid the controlled sources.

Invariant Driving Point Impedance Without Introducing Controlled Sources

In the circuit of Fig. 2 a controlled current source appeared due to the fact that the matrix A^{-1} has non-zero elements in the first column. If the matrix C of Eq. (22) is restricted to be of the form

$$A^{-1} = C = \begin{bmatrix} \beta & C_{12} & \cdots & C_{1N} \\ 0 & C_{22} & \cdots & C_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & C_{N2} & \cdots & C_{NN} \end{bmatrix}. \quad (32)$$

No controlled sources appear in the new circuit. The rest of the procedure is identical to the previous one. Returning to Eq. (25), if $C_{21} = 0$ so that Eq. (32) be satisfied, and the following choice is made

$$\beta = 77, \qquad C_{12} = 13, \qquad C_{22} = -13, \qquad B_{12} = \text{[illegible]}, \qquad B_{22} = \text{[illegible]},$$

the resulting $\hat{P}$ is

$$\hat{P} = \begin{bmatrix} 3/2 & -1 \\ -1 & 2 \end{bmatrix}.$$

The corresponding circuit is shown in Fig. 3, where the driving point impedance from node 1 to the datum is unity, as in Figs. 1 and 2. However, no gyrators or controlled sources appear. It is possible also to go from a transformed circuit to the original one by applying another transformation. From Eq. (16), solving for Y, $Y = A\hat{P}B^{-1}$, which expresses the same transformation as Eq. (16) since A and A^{-1} are of the same form and likewise B and B^{-1}. The fact is obvious if A^{-1} is of the form of Eq. (23) since all the elements are arbitrary. For matrices of the form given by Eq. (22) note that, except the first, all cofactors of the elements in the first column of the transpose of B are zero. Hence, B^{-1} will have zeros in the elements of the first row (except the first element). A similar statement can be made for matrices of the form given in Eq. (32). This means that we may start with a bilateral circuit and end with a non-bilateral circuit and vice versa.

Invariant Transfer Impedance

Instead of keeping a driving point impedance invariant it may be desired to keep a transfer impedance invariant, say, the transfer impedance between the ports formed of node 1 and ground and node 2 and ground. Then, following a procedure similar to the one followed in Eqs. (17) through (23), it is sufficient to restrict B and C to be of the form

$$B = \begin{bmatrix} B_{11} & B_{12} & B_{13} & \cdots & B_{1N} \\ 0 & 1/\beta & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ B_{N1} & B_{N2} & B_{N3} & \cdots & B_{NN} \end{bmatrix}. \quad (33)$$

$$C = A^{-1} = \begin{bmatrix} \beta & C_{12} & \cdots & C_{1N} \\ C_{21} & C_{22} & \cdots & C_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ C_{N1} & C_{N2} & \cdots & C_{NN} \end{bmatrix}. \quad (34)$$

This gives a circuit in which z_{21} is kept invariant and controlled sources appear. If no controlled sources are desired, the matrix C should be of the form of Eq. (32)

so that the currents entering nodes 2, 3, $\cdots$, N are not related to the current entering node 1, as indicated by Eq. (13). If we desire to keep z_{11} invariant, rather than starting with a circuit in which $I(s)$ is of the form of (12') we should have

$$I(s) = \begin{bmatrix} 0 \\ I_1(s) \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}. \tag{35}$$

The matrix B should be of the form of B of Eq. (22) and $C = A^{-1}$ should be of the form

$$C = A^{-1} = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1N} \\ C_{21} & \beta & \cdots & C_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ C_{N1} & C_{N2} & \cdots & C_{NN} \end{bmatrix}. \tag{36}$$

The transformed circuit will have controlled sources. If no controlled sources are desired, C should be of the form

$$C = \begin{bmatrix} C_{11} & 0 & \cdots & C_{1N} \\ C_{21} & \beta & \cdots & C_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ C_{N1} & 0 & \cdots & C_{NN} \end{bmatrix}. \tag{37}$$

Note that Cauer's transformation theory cannot handle this problem (10). It should be obvious how to extend these concepts to maintain invariant several driving point and/or transfer impedances.

Impedance Driving Point or Transfer Impedance to Yield a Bilateral Circuit

If the original Y is symmetric and

$$A^{-1} = B^T, \tag{38}$$

where the superscript T denotes transposition; then according to Eq. (16)

$$\hat{Y} = B^T Y B = \hat{Y}^T. \tag{39}$$

Hence $\hat{Y}$ is symmetric. E ion (38) is a *sufficient* condition for symmetry of $\hat{Y}$ if Y is symmetric. If Eq. (39) is used, β of Eqs. (20) and (21) is forced to be unity.

$$\hat{Y} = \begin{bmatrix} 1 & B_{21} & \cdots & B_{N1} \\ 0 & B_{22} & \cdots & B_{N2} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & B_{2N} & \cdots & B_{NN} \end{bmatrix} \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1N} \\ Y_{12} & Y_{22} & \cdots & Y_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ Y_{1N} & Y_{2N} & \cdots & Y_{NN} \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ B_{21} & B_{22} & \cdots & B_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ B_{N1} & B_{N2} & \cdots & B_{NN} \end{bmatrix}. \tag{40}$$

So far, the entries B_{ij} are still arbitrary rational functions of s. If they are restricted to be constants the transformation reduces to the Cauer transformation. Although condition (38) is sufficient for ending with a symmetric $\hat{Y}$, if the original Y is symmetric, it is not necessary (7). All that is necessary is that

$$\hat{Y} = \hat{Y}^T; \tag{40'}$$

this gives, using Eq. (16),

$$A^{-1} Y B = B^T Y^T (A^{-1})^T. \tag{41}$$

Equation (41) represents $(N^2 - N)/2$ equations with which we may eliminate as many variables from the matrices B and A^{-1}, leaving the rest of the entries arbitrary. Equation (41) is especially useful if the original Y is not symmetric, and when keeping transfer impedances invariant while changing all the driving point impedances. Its use is illustrated for the example of Fig. 1. For that case, the matrix Eq. (41) reduces to the single scalar Eq. (27).

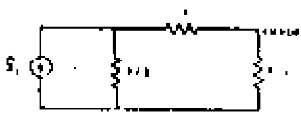


FIG. 3. Circuit equivalent to those of Figs. 1 and 2.

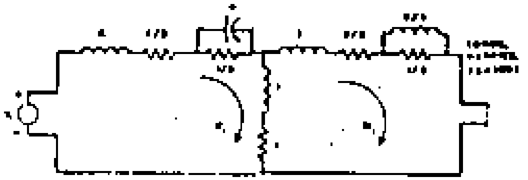


FIG. 4. Circuit to be transformed using the loop basis.

Passive Transformerless Synthesis by Transformation

If the driving point impedance of interest is realizable and we wish to obtain an RLC circuit (no transformers or mutual inductances) with positive elements, besides wanting $\hat{Y}$ to be symmetric, it is sufficient that the negative of the terms off the main diagonal as well as the sum of the entries of each row are positive real functions. Few general results exist to determine A^{-1} and B which guarantees that $\hat{Y}$ will satisfy these conditions.

Pantell (19), Guillemin (13), Duda (5), Darlington (12), Schneider (17), and Coderbaum (16) have some results for two element networks. Schoeffler

(15) has done work in the RLC problem. However, the problem is far from solved. Newcomb (18) treats the problem allowing the presence of transformers.

Invariant Driving Point Impedance by Transforming Circuits on the Loop Basis

What has been said for the nodes basis can be extended to the loop basis. The loop basis is appropriate for keeping short-circuit driving point and transfer admittances invariant on ports formed by making plier-type of entries into the loops. (In the node basis soldering-iron-type of entries at node pairs were made.) As an example of a transformation on the loop basis, we consider a simple problem in passive synthesis. This example illustrates the use of transformation matrices whose entries are functions of s, thus exploiting the functional character of the transformation. Consider the circuit of Fig. 4 whose loop impedance matrix is

$$Z = \left[\begin{array}{c|c} 3s + 3 + \dfrac{s+2}{2s+3} & -(s+3) \\ \hline -(s+3) & s + 3 + \dfrac{(s+1)(s+3)}{s+2} \end{array}\right]$$

The matrix Z may be transformed into a new matrix $\hat{Z}$ according to

$$\hat{Z} = CZB. \tag{42}$$

In order to keep the short-circuit driving point admittance of a plier entry at loop 1 invariant, C and B will be of the forms

$$A^{-1} = C = \begin{bmatrix} \beta & p \\ 0 & r \end{bmatrix}, \quad B = \begin{bmatrix} 1/\beta & 0 \\ m & n \end{bmatrix}. \tag{43}$$

Since C and B are of the forms indicated by Eqs. (22) and (32) the new circuit will not contain controlled sources. If, besides, Eq. (41) is satisfied the resulting circuit will be bilateral. If the multiplications indicated by Eq. (42) are performed with the aid of Eq. (43), an equation similar to Eq. (27) is written and m eliminated. The new matrix $\hat{Z}$ may be written as follows:

$$\hat{Z} = \left[\begin{array}{c|c} Z_{11} + \dfrac{np^2}{r} Z_{22} + Z_{12}\left[\dfrac{2np}{\beta r} + \dfrac{Z_{11}}{Z_{12}}\left(\dfrac{n}{\beta r} - 1\right)\right] & \dfrac{nZ_{12}}{\beta} + npZ_{22} \\ \hline \dfrac{nZ_{12}}{\beta} + npZ_{22} & nrZ_{22} \end{array}\right]. \tag{44}$$

In Eq. (44) the elements β, p, r, n are still arbitrary and may be manipulated to make the final circuit realizable.

The following choice in the parameters in Eq. (43)

$$\beta = 1, \quad p = 0, \quad r = 1, \quad n = (s+2)/(s+3),$$

and

$$m = s + 2/(2s+3)(s+3),$$

gives the new loop impedance matrix

$$\hat{Z} = \begin{bmatrix} 3s + 3 & -(s+2) \\ -(s+2) & 2s+3 \end{bmatrix},$$

whose realization is shown in Fig. 5. The short-circuit admittance at loop 1 of both circuits of Figs. 4 and 5 is

$$y = (2s+3)/(5s^2 + 11s + 6).$$

The transformation eliminates the capacitance in the circuit which is redundant. We emphasize that in general it is very difficult to choose the transformation coefficients such that the resulting network is realizable with positive elements. Furthermore, with respect to transformations in the loop basis it is well known that circuits with more than three independent meshes must meet complicated constraints unless transformers or some other non-conductive couplings are employed (14).

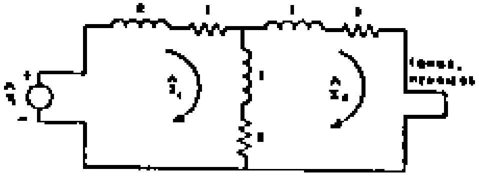


FIG. 5. Circuit equivalent to that of Fig. 4.

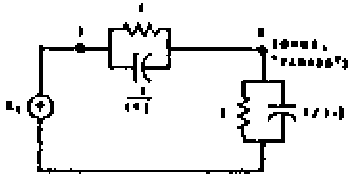


FIG. 6. "Foster Realization" of a circuit with complex poles.

Non-real Transformations

In the previous example a realizable circuit was transformed into a realizable one. It is possible also to transform a realizable circuit into a non-realizable one and *vice versa*. An extreme case is provided by the following example:

The impedance

$$z = \frac{2s+4}{s^2+2s+2} = \frac{1+j}{s+1+j} + \frac{1-j}{s+1-j},$$

can be realized in a Foster form, see Fig. 6. The Y of the circuit of Fig. 6 is

$$Y = \left[\begin{array}{c|c} 1+\dfrac{s}{1+j} & -\left(1+\dfrac{s}{1+j}\right) \\ \hline -\left(1+\dfrac{s}{1+j}\right) & \left(\dfrac{1}{1+j}+\dfrac{1}{1-j}\right)s+2 \end{array}\right]$$

Using Eq. (44) (interpreted for a $\hat{P}$ matrix instead of a $\hat{Z}$ matrix) and choosing $\beta = 1, p = 0, r = 1, n = 0,$

$$\hat{P} = \left[\begin{array}{c|c} (s/2) + 1/(s+2) & 0 \\ \hline 0 & 0 \end{array}\right],$$

whose realization is shown in Fig. 7. Analysis shows that

$$z = \frac{2s+4}{s^2+2s+2}.$$

Altering the Driving Point Impedance Through Transformations

The driving point impedance need not remain invariant after the transformation. It can be scaled (6) by a factor a, where a is a ratio of polynomials. For this transformation the operator B of Eq. (22) is of the form

$$B = \begin{bmatrix} 1/a\beta & 0 & \cdots & 0 \\ B_{21} & B_{22} & \cdots & B_{2N} \\ B_{N1} & B_{N2} & \cdots & B_{NN} \end{bmatrix}. \tag{45}$$

Hence, if the driving point impedance of the original circuit is z, the one of the transformed circuit will be az.

Changing the Order of the Matrices

With the transformations described so far the order of the final admittance matrix is the same as the one of the original matrix. Although it is possible to introduce new nodes, the added nodes will not be connected to all the others. If it is desired to increase the complexity of the networks it is possible to do so by introducing a larger matrix Y_A partitioned in the following diagonal form:

$$Y_A = \left[\begin{array}{c|c} Y & 0 \\ \hline 0 & \Gamma \end{array}\right],$$

where Y is the original $N \times N$ admittance matrix, 0 is the zero matrix, and Γ is an $M \times M$ matrix. The matrix Y_A, which is now $(N+M) \times (N+M)$ may be transformed with transformation matrices of the same order. (See 12).

Application of the Transformation to Analysis

So far the transformation has been applied to the synthesis of equivalent circuits. An application to analysis is now considered which illustrates the advantage of having formulated the transformation in abstract terms. The transformation is of the *invariant-object* type.

In power systems one deals with a single frequency, steady state problem. Kenelly introduced in 1893 the idea of handling the problem with complex numbers. An alternate treatment using only real matrices is possible (11).

Define the operator L as follows:

Domain of L: all 2-vectors with real components;
Range of L: all sinusoids of arbitrary phase and fixed frequency ω;

$$L\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = V_1 \sin \omega t + V_2 \cos \omega t. \tag{46}$$

The operator L has an inverse L^{-1} whose domain is the range of L and whose range is the domain of L and

$$L^{-1}(V_1 \sin \omega t + V_2 \cos \omega t) = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}. \tag{47}$$

It can be shown that if V_a and V_b are 2-vectors L satisfies

$$L(\alpha V_a + \beta V_b) = \alpha L V_a + \beta L V_b.$$

Let

$$y = (d/dt)x \tag{48}$$

be transformed according to

$$y = L\hat{y}, \tag{49}$$

$$x = L\hat{x}. \tag{50}$$

Hence,

$$\hat{y} = L^{-1}(d/dt)L\hat{x}. \tag{51}$$

Now let x and y be sinusoids given by

$$x = X_1 \sin \omega t + X_2 \cos \omega t, \tag{52}$$

$$y = Y_1 \sin \omega t + Y_2 \cos \omega t. \tag{53}$$

Using Eqs. (49) and (50)

$$\hat{x} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, \tag{54}$$

$$\hat{y} = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}. \tag{55}$$

Substitution of (54) and (55) gives

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = L^{-1}\frac{d}{dt}L\begin{bmatrix} X_1 \\ X_2 \end{bmatrix}. \tag{56}$$

Using definition Eq. (46)

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} \begin{matrix} = L^{-1}(d/dt)\,(X_1 \sin \omega t + X_2 \cos \omega t) \\ = L^{-1}(-\omega X_1 \cos \omega t + \omega X_2 \sin \omega t). \end{matrix} \tag{57}$$

Using Eq. (47) in Eq. (57)

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \omega X_2 \\ -\omega X_1 \end{bmatrix} = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \tag{58}$$

Hence comparing (58) and (51)

$$L^{-1}(d/dt)L = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix}. \tag{59}$$

In a similar manner it can be shown that for

$$F_1 = \int [\cdot]\,dt$$

$$L^{-1}F_1L = \begin{bmatrix} 0 & -1/\omega \\ 1/\omega & 0 \end{bmatrix} \tag{61}$$

and for

$$F_2 = K[\cdot] \tag{62}$$

(K a multiplicative constant)

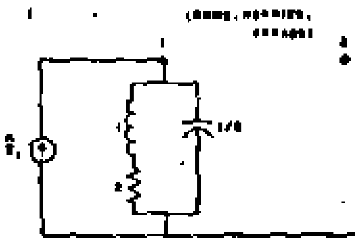


FIG. 7. Circuit equivalent to that of Fig. 6.

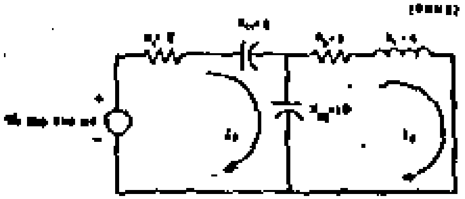


FIG. 8. Circuit to be analyzed without using complex numbers.

$$L^{-1}F_2L = \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix}. \tag{63}$$

If x represents current and y represents voltage then a series RLC circuit of operational impedance

$$z = R + L(d/dt) + C^{-1}\int dt$$

in the time domain has an operational impedance in the new coordinate system of

$$\hat{z} = \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} + \begin{bmatrix} 0 & \omega L \\ -\omega L & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1/\omega C \\ 1/\omega C & 0 \end{bmatrix} \tag{64}$$

$$\hat{z} = \begin{bmatrix} R & \omega L - 1/\omega C \\ 1/\omega C - \omega L & R \end{bmatrix} = \begin{bmatrix} R & X \\ -X & R \end{bmatrix} \tag{65}$$

where $X = \omega L - 1/\omega C$. For a multiloop system, such as that shown in Fig. 8, transforming separately each v_i, z_{ij}, and i_j in

$$\begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{bmatrix}\begin{bmatrix} i_1(t) \\ i_2(t) \end{bmatrix} \tag{66}$$

and writing the resultant equation in partitioned form, the following equation is obtained in the new coordinate system:

$$\begin{bmatrix} \begin{bmatrix} \hat{V}_1 \\ \hat{V}_2 \end{bmatrix}_1 \\ \begin{bmatrix} \hat{V}_1 \\ \hat{V}_2 \end{bmatrix}_2 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} R & X \\ -X & R \end{bmatrix}_{11} & \begin{bmatrix} R & X \\ -X & R \end{bmatrix}_{12} \\ \begin{bmatrix} R & X \\ -X & R \end{bmatrix}_{21} & \begin{bmatrix} R & X \\ -X & R \end{bmatrix}_{22} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \hat{I}_1 \\ \hat{I}_2 \end{bmatrix}_1 \\ \begin{bmatrix} \hat{I}_1 \\ \hat{I}_2 \end{bmatrix}_2 \end{bmatrix} \tag{67}$$

which may be written, using the values of Fig. 8

$$\begin{bmatrix}\begin{bmatrix}100\\0\end{bmatrix}\\ \begin{bmatrix}0\\0\end{bmatrix}\end{bmatrix}=\begin{bmatrix}\begin{bmatrix}2 & -12\\12 & 2\end{bmatrix} & \begin{bmatrix}0 & 10\\-10 & 0\end{bmatrix}\\ \begin{bmatrix}0 & 10\\-10 & 0\end{bmatrix} & \begin{bmatrix}3 & -6\\6 & 3\end{bmatrix}\end{bmatrix}\begin{bmatrix}\begin{bmatrix}I_1\\I_2\end{bmatrix}_1\\ \begin{bmatrix}I_1\\I_2\end{bmatrix}_2\end{bmatrix}. \qquad (68)$$

Solving for the currents,

$$\mathbf{I}=\begin{bmatrix}\begin{bmatrix}11.75\\1.73\end{bmatrix}\\ \begin{bmatrix}13.9\\0.85\end{bmatrix}\end{bmatrix}.$$

Using Eq. (49) this implies that in Fig. 8

$$i_1 = 11.75 \sin \omega t + 1.73 \cos \omega t$$
$$i_2 = 13.9 \sin \omega t + 0.85 \cos \omega t.$$

Here, the use of complex numbers is avoided. This method is useful for digital computer calculations in which a FORTRAN subroutine for inverting real matrices in double precision is available. In this case the subroutine as it stood could not handle double precision complex matrices.

Conclusions

The paper has introduced a functional transformation of a very general character and illustrated some of the applications to circuit analysis and synthesis. Alternatively, one may apply the transformation to *A*, *B*, *C*, *D* parameter matrices or other parameter representations of circuits. Other possible applications include distributed circuits and time varying circuits. This author feels that transformation theory applied to circuit theory is a wide open field of research that has thus far remained largely untapped.

Acknowledgment

The author is indebted to Prof. P. Moon and Prof. E. A. Guillemin for helpful discussions.

References

(1) W. Cauer, "Vierpole," *Elek. Nachr. Tech.*, Vol. 6, pp. 272–282, 1929.
(2) N. Howitt, "Equivalent Electrical Networks," *Proc. IRE*, Vol. 20, pp. 1042–1051, 1932.
(3) R. S. Darlington, "*R*-Matrices and Equivalent Networks," *Jour. Math. Phys.*, Vol. 16, pp. 85–103, 1937.

(4) E. A. Guillemin, "Transformation Theory Applied to Linear Active and/or Non-bilateral Networks," *IRE Trans. on Circ. Thy.*, pp. 106–111, Sept. 1957.
(5) R. Duda, "Equivalent and Optimal Equivalent Electrical Networks," Ph.D. Diss., M.I.T., 1962.
(6) M. A. Murray-Lasso, "Generalized Impedance Leveling in Network Synthesis," *Proc. 2nd Annual Allerton Conf. on Circ. and Syst. Thy.*, pp. 820–840, 1964.
(7) M. A. Murray-Lasso, "Matrix Transformations in Circuit Theory," M.S. Thesis M.I.T., 1962.
(8) L. P. Huelsman, "Circuits, Matrices, and Linear Vector Spaces," New York, McGraw-Hill, 1963.
(9) E. Guillemin, "Synthesis of Passive Networks," New York, John Wiley, 1957.
(10) S. Darlington, "A Survey of Network Realization Techniques," *IRE Trans. on Circ. Thy.*, Vol. CT-2, 1955.
(11) P. Moon and D. E. Spencer, "A New Mathematical Representation of Alternating Currents," *J. Tensor Soc. of Japan*, 1964.
(12) S. Darlington, "On Three Terminal Circuits Without Transformers," *Proc. Third Annual Allerton Conf. on Circ. and Syst. Thy.*, Univ. of Ill., Oct. 1965.
(13) E. Guillemin, "Theory of Linear Physical Systems," New York, John Wiley, 1963.
(14) R. M. Foster, "Topologic and Algebraic Considerations in Network Synthesis," *Proc. Symp. on Modern Network Synthesis*, Brooklyn Polytech. Inst., 1952.
(15) D. A. Calahan, "Modern Network Synthesis," Vol. 2, New York, Hayden Book Co., 1964.
(16) I. Cederbaum, "Dominant Matrices and Their Application to Network Synthesis Under Topological Constraints," *Franklin Inst.*, Dec. 1964.
(17) A. J. Schneider, "RC Driving-Point Impedance Realization by Linear Transformations," *IEEE Trans. on Circ. Thy.*, Vol. CT-13, No. 3, pp. 265–271, Sept. 1966.
(18) R. W. Newcomb, "Linear Multiport Synthesis," New York, McGraw-Hill, 1966.
(19) R. H. Pantell, "New Methods of Driving-Point and Transfer Function Synthesis," Tech. Rept. No. 76, Electronics Res. Lab. Stanford Univ., Stanford, Calif., July 19, 1954.

INSTITUTO
DE
INGENIERIA

ANALISIS DE CIRCUITOS CON COMPUTADORA DIGITAL

M A MURRAY LASSO
L P MC NAMEE

281

ENERO 1971

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

Análisis de circuitos con computadora digital

M. A. Murray Lasso*
L. P. McNamee**

RESUMEN

Se presenta una vista panorámica con apuntes bibliográficos sobre diseño de circuitos con computadora digital. Se ilustran sus posibilidades con un ejemplo específico de un circuito electrónico que se analiza con un programa general.

ABSTRACT

The field of computer-aided circuit design is briefly surveyed and a bibliography is given. Its posibilities are illustrated with a specific example by analizing an electronic circuit with a general purpose program.

1. INTRODUCCION

La práctica del análisis de circuitos con computadora digital es tan vieja como la computadora misma. Inmediatamente después de la Segunda Guerra Mundial se comenzaron a analizar filtros en el dominio de la frecuencia en una computadora de relevadores en los Laboratorios *Bell Telephone*. El primero y más entusiasta investigador de la teoría de análisis de circuitos de estructura general es Gabriel Kron (ref 1), quien redujo el análisis de circuitos a una serie de operaciones rutinarias que eliminaban el tener que "pensar" y, por lo tanto, se podían automatizar. Desgraciadamente expresó sus trabajos en términos de conceptos matemáticos no muy comunes en la rama de ingeniería, por lo que pocos entienden sus métodos. Fue necesario que otros investigadores, como Le Corbeiller y Branin (refs 2 y 3), trascribieran sus trabajos a términos mas comunes en ingeniería.

Desde 1958, veinte años después de los esfuerzos iniciales de Kron, la rama de análisis de circuitos con computadora ha crecido a velocidad vertiginosa. Hoy existen cientos de programas para analizar circuitos lineales y no lineales en el dominio del tiempo y la frecuencia. Muchos de ellos dan información

*Instituto de Ingeniería, UNAM, México, D. F.
**Universidad de California, Los Angeles, E.U.A.

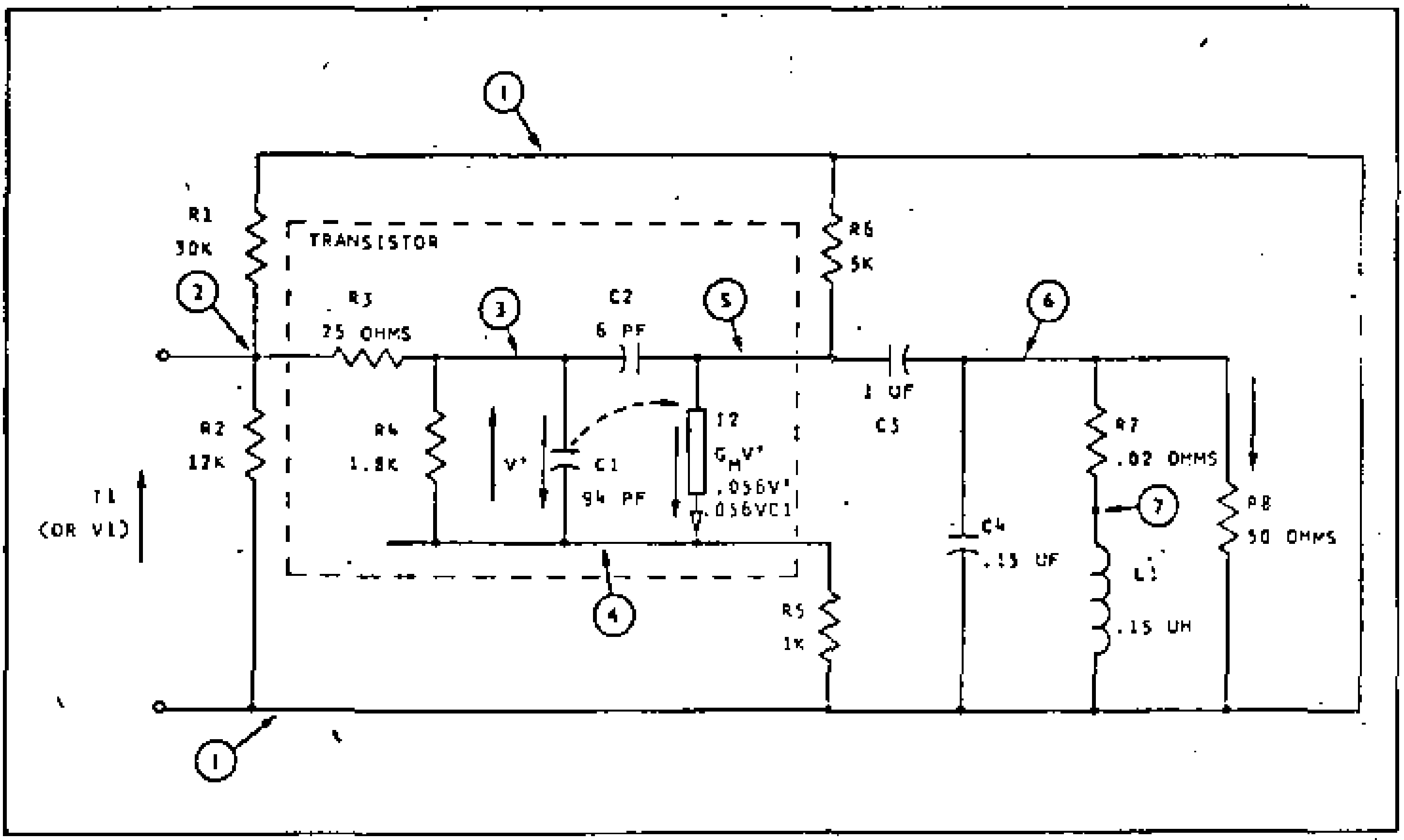


Fig 3.

```
NASAP PROBLEM 41N01   CODING BOOK 011769

I1 1 2 1.
R1 1 2 30K
R2 2 1 12K
R3 2 3 25
R4 3 4 1.8K
C1 3 4 94PF
C2 3 5 6PF
I2 5 4 0.056 VC1
R5 4 1 1K
R6 5 1 5K
C3 5 6 1UF
C4 6 1 0.15UF
R7 6 7 0.02
L1 7 1 0.15UH
R8 6 1 50
OUTPUT
IR8/II1/C4
FREQ  5.8  6.2  0.01
TIME 0.000001
EXECUTION
```

Fig 4.

La primera tarjeta indica la presencia de una fuente de corriente llamada I1, conectada a los nudos 1 y 2 (el orden en que se dan los nudos indica la dirección de referencia), cuyo valor es 1 ampere.

La segunda tarjeta indica la presencia de una resistencia llamada *R1*, conectada a los nudos 1 y 2, cuyo valor es de 30 kilohms.

La tercera, cuarta y quinta dan información similar que el lector puede deducir fácilmente.

La sexta tarjeta indica la presencia de un capacitor *C1* conectado a los nudos 3 y 4, cuyo valor es de 94 picofarads.

La octava tarjeta indica la presencia de una fuente de corriente *I2* que está conectada a los nudos 5 y 4, cuyo valor es 0.056 veces el valor del voltaje a través del capacitor *C1* (indicado por la sigla *VC1*) que la controla.

La decimocuarta tarjeta indica la presencia de un inductor *L1* conectado entre los nudos 7 y 1, cuyo valor es de 0.15 microhenries.

El lector puede comparar el resto de las tarjetas no descritas con el esquema de la fig 3, para darse cuenta cómo se le puede describir un circuito a una máquina "ciega".

Las últimas cinco tarjetas contienen información sobre los cálculos que desea el usuario que haga la computadora. La tarjeta OUTPUT indica a la computadora que ha terminado la descripción del circuito y que a continuación aparecerán los resultados deseados. La siguiente tarjeta indica que se desea calcular la ganancia *IR8/II1* y conocer la sensibilidad de dicha ganancia a variaciones en la capacitancia *C4*. La antepenúltima tarjeta indica que se desea evaluar dichas cantidades en el intervalo de frecuencias entre 10 a la potencia 5.8, y 10 a la potencia 6.2 en 100 (1/.01) intervalos logarítmicos por década. La penúltima indica que se desea la respuesta en el tiempo correspondiente a la función de transferencia indicada en las anteriores tarjetas, y que esta función se debe evaluar hasta .000001 segundos. (El programa da automáticamente 100 puntos intercalados.) La última tarjeta indica que toda la información ha sido dada y ordena a la computadora que ejecute el análisis.

Ejecución del programa. El programa genera en la impresora los siguientes resultados:

1. Una copia de los datos contenidos en las tarjetas que alimentaron a la computadora. (Esto sirve para asegurarse de que la computadora ha aceptado los datos correctamente, y para facilitar la localización de errores en la codificación, en caso de haberlos.)

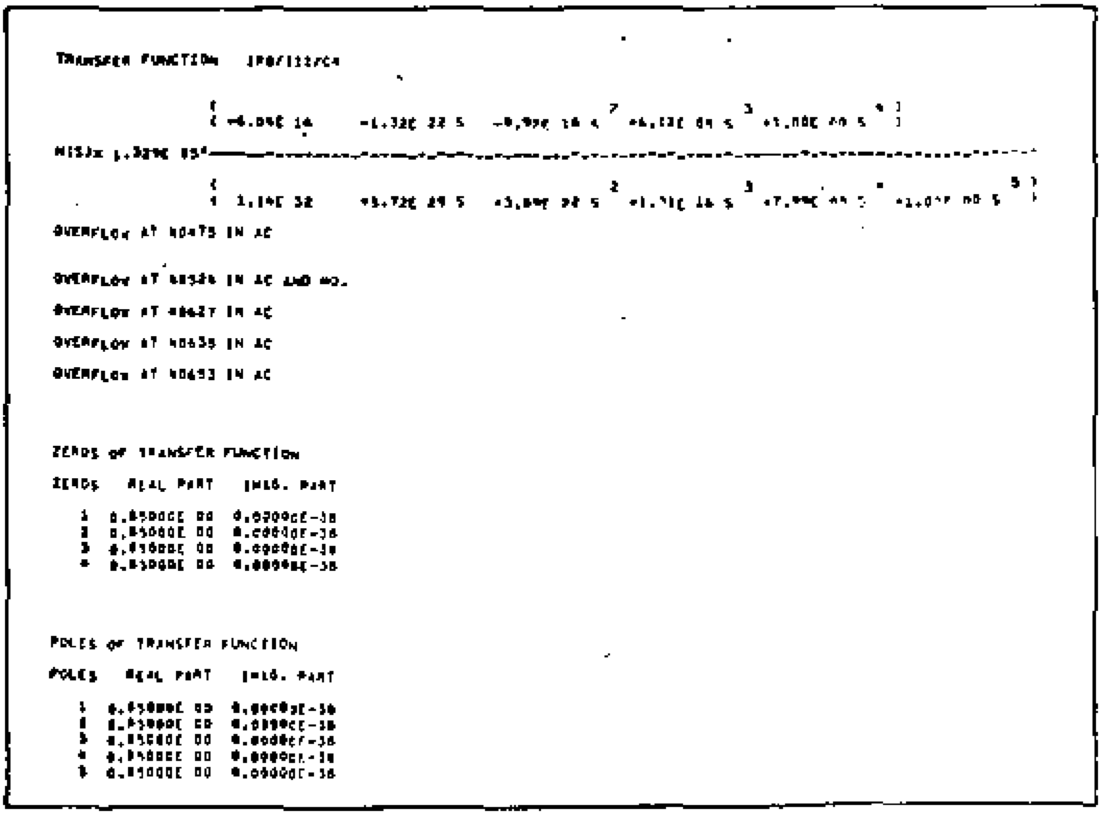
2. Una lista de los números de mallas de diferentes órdenes en la solución del problema por métodos de reogramas. Esto es de interés para el teórico en reogramas. Los números pueden servir de guía al usuario para darse una idea de la complejidad computacional del problema.

En vista de que lo mencionado en el punto 1 es una copia idéntica de la fig 4, y que los resultados de 2 son de poco interés para el ingeniero diseñador, no se muestran esos resultados.

3. La función de transferencia (o varias de ellas si así se pidió), la cual aparece impresa como una razón de polinomios en la frecuencia compleja.

4. Los ceros y polos de la función de transferencia.

Estos dos últimos aparecen en la fig 5. (Debe hacerse notar que los valores de los polos y los ceros son muy extraños, y están equivocados; se aclara esta situación más adelante. Obsérvense también los mensajes producidos por la computadora sobre desborde numérico)



```
TRANSFER FUNCTION   IR8/II1/C4

                 (                                         7             3             4 )
                 ( -4.09E 14    -1.32E 22 S   -9.97E 14 S   +6.13E 09 S   +1.00E 00 S   )
H(S)= 1.329E 05*-----------------------------------------------------------------------------
                 (                                         2             3             4             5 )
                 (  1.14E 32    +5.72E 29 S   +3.84E 24 S   +1.71E 16 S   +7.99E 05 S   +1.07E 00 S   )
OVERFLOW AT 40475 IN AC

OVERFLOW AT 40526 IN AC AND MQ.

OVERFLOW AT 40627 IN AC

OVERFLOW AT 40635 IN AC

OVERFLOW AT 40653 IN AC



ZEROS OF TRANSFER FUNCTION

ZEROS    REAL PART    IMAG. PART

   1   0.850000E 00  0.000000E-38
   2   0.850000E 00  0.000000E-38
   3   0.850000E 00  0.000000E-38
   4   0.850000E 00  0.000000E-38



POLES OF TRANSFER FUNCTION

POLES    REAL PART    IMAG. PART

   1   0.850000E 00  0.000000E-38
   2   0.850000E 00  0.000000E-38
   3   0.850000E 00  0.000000E-38
   4   0.850000E 00  0.000000E-38
   5   0.850000E 00  0.000000E-38
```

Fig.5.

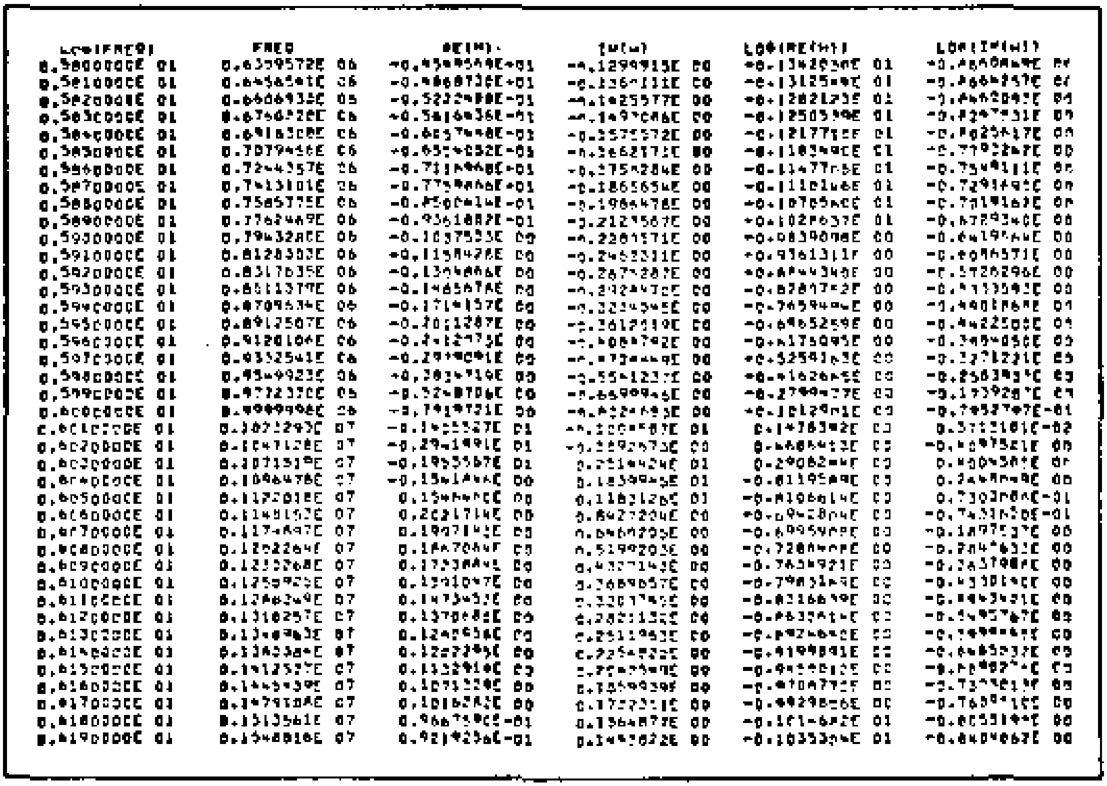
5. Varias listas en las que aparecen el logaritmo de la frecuencia, la frecuencia, la parte real e imaginaria de la función de transferencia *H(S)*, los logaritmos de las anteriores, la magnitud y fase, el logaritmo de la magnitud, y la magnitud en decibeles de la función *H(S)*. En la fig 6 aparece una lista parcial que contiene algunas de las cantidades mencionadas. Además de obtener estas cantidades en forma de listas, el programa genera los resultados en gráficas hechas con la impresora. Dos de dichas gráficas en las que aparecen la magnitud de la función de transferencia en decibeles y su fase se muestran en las figs 7 y 8.

6. Se generan varias clases de sensibilidad: logarítmica, que se representa con SENS (H), incremental de 1 por ciento de la parte real, parte imaginaria, magnitud y fase de la función de transferencia, que se representan con SENS (RE(H)), SENS (IM(H)), SENS (ABS(H)), y SENS (PHI(H)), respectivamente. Se generan también sensibilidades de las posiciones de los polos y ceros de la función de transferencia. Estas sensibilidades aparecen en una multitud de formas análogas a las mencionadas en el inciso 5. Asimismo se generan con la impresora las gráficas correspondientes. En las figs 9 a 12 aparecen una tabla parcial y tres gráficas con información referente a las sensibilidades del ejemplo que se está ilustrando.

7. Respuesta transitoria al impulso correspondiente a la función de transferencia, la cual se calcula desde el tiempo cero hasta el instante elegido por el usuario.

Para obtener la respuesta al impulso, el programa hace uso de los polos de la función de transferencia. Como se indicó anteriormente, debido a los desbordes numéricos ocurridos, dichos polos (y ceros) están incorrectamente calculados. La razón por la cual se presenta este problema numérico es la enorme diferencia entre las magnitudes de los coeficientes de los polinomios del numerador y denominador de la función de transferencia que aparece en la fig 5. La dificultad se puede vencer en parte si se hace un cambio de variables s' = 100 000s. El efecto de este cambio en los valores de los elementos del circuito es el de multiplicar por 100 000 los valores de las inductancias y capacitancias, dejando todos los demás parámetros del circuito sin cambio. Con respecto al tiempo, el efecto es multiplicarlo por 100 000. Finalmente, habrá también que multiplicar el valor del fenómeno transitorio por el mismo factor de 100 000; todas estas equivalencias son bien conocidas en teoría de sistemas lineales (ref 63).

En la fig 13 se muestra la descripción del circuito tras el cambio de variables. Obsérvese que las resistencias y transconductancias en el modelo del transistor permanecen iguales mientras que las inductancias y capacitancias han sido multiplicadas por 100 000. Se hace notar, igualmente, el cambio en los datos



```
   LOG(FREQ)          FREQ              RE(H)              IM(H)            LOG(RE(H))          LOG(IM(H))
0.5800000E 01    0.6309572E 06    -0.4549549E-01    -0.1299915E 00    -0.1342030E 01    -0.8860849E 00
0.5810000E 01    0.6456541E 06    -0.4668736E-01    -0.1361111E 00    -0.1312549E 01    -0.8664257E 00
0.5820000E 01    0.6606932E 06    -0.5222498E-01    -0.1425577E 00    -0.1282123E 01    -0.8460209E 00
0.5830000E 01    0.6760829E 06    -0.5416436E-01    -0.1497086E 00    -0.1250359E 01    -0.8247731E 00
0.5840000E 01    0.6918310E 06    -0.6657448E-01    -0.1575572E 00    -0.1217715E 01    -0.8025617E 00
0.5850000E 01    0.7079456E 06    -0.6514032E-01    -0.1662171E 00    -0.1183490E 01    -0.7792287E 00
0.5860000E 01    0.7244357E 06    -0.7119460E-01    -0.1754284E 00    -0.1147785E 01    -0.7559111E 00
0.5870000E 01    0.7413101E 06    -0.7759466E-01    -0.1865654E 00    -0.1110146E 01    -0.7291491E 00
0.5880000E 01    0.7585775E 06    -0.8500414E-01    -0.1986478E 00    -0.1070540E 01    -0.7019162E 00
0.5890000E 01    0.7762469E 06    -0.9361882E-01    -0.2121567E 00    -0.1028637E 01    -0.6732934E 00
0.5900000E 01    0.7943282E 06    -0.1037553E 00    -0.2201571E 00    -0.9839098E 00    -0.6419564E 00
0.5910000E 01    0.8128303E 06    -0.1158478E 00    -0.2452311E 00    -0.9361311E 00    -0.6096571E 00
0.5920000E 01    0.8317635E 06    -0.1304864E 00    -0.2677287E 00    -0.8844349E 00    -0.5726296E 00
0.5930000E 01    0.8511377E 06    -0.1465878E 00    -0.2924472E 00    -0.8287752E 00    -0.5339592E 00
0.5940000E 01    0.8709634E 06    -0.1714137E 00    -0.3224345E 00    -0.7659445E 00    -0.4900868E 00
0.5950000E 01    0.8912507E 06    -0.2011287E 00    -0.3612919E 00    -0.6965259E 00    -0.4422509E 00
0.5960000E 01    0.9120104E 06    -0.2412773E 00    -0.4086792E 00    -0.6175095E 00    -0.3894050E 00
0.5970000E 01    0.9332541E 06    -0.2974041E 00    -0.4734449E 00    -0.5259163E 00    -0.3271221E 00
0.5980000E 01    0.9549923E 06    -0.3834719E 00    -0.5541227E 00    -0.4162645E 00    -0.2563953E 00
0.5990000E 01    0.9772372E 06    -0.5240706E 00    -0.6599945E 00    -0.2799477E 00    -0.1739287E 00
0.6000000E 01    0.9999996E 06    -0.7919721E 00    -0.8522493E 00    -0.1012971E 00    -0.6932797E-01
0.6010000E 01    0.1023293E 07    -0.1425527E 01    -0.1200587E 01     0.1547632E 00     0.7935101E-01
0.6020000E 01    0.1047128E 07    -0.2941991E 01    -0.2292673E 00     0.4686541E 00    -0.6397521E 00
0.6030000E 01    0.1071519E 07    -0.1955567E 01     0.2514424E 01     0.2906284E 00     0.4004307E 00
0.6040000E 01    0.1096478E 07    -0.1541446E 00     0.1839945E 01    -0.8119589E 00     0.2648849E 00
0.6050000E 01    0.1122018E 07     0.1544645E 00     0.1182126E 01    -0.8106614E 00     0.7302806E-01
0.6060000E 01    0.1148153E 07     0.2621714E 00     0.8427224E 00    -0.5814280E 00    -0.7431620E-01
0.6070000E 01    0.1174897E 07     0.1997142E 00     0.6460295E 00    -0.6995902E 00    -0.1897537E 00
0.6080000E 01    0.1202264E 07     0.1867064E 00     0.5199203E 00    -0.7288442E 00    -0.2840632E 00
0.6090000E 01    0.1230269E 07     0.1723844E 00     0.4327143E 00    -0.7635972E 00    -0.3637981E 00
0.6100000E 01    0.1258925E 07     0.1591047E 00     0.3680857E 00    -0.7983169E 00    -0.4330147E 00
0.6110000E 01    0.1288249E 07     0.1473432E 00     0.3201745E 00    -0.8316699E 00    -0.4945421E 00
0.6120000E 01    0.1318257E 07     0.1370682E 00     0.2821135E 00    -0.8631741E 00    -0.5495767E 00
0.6130000E 01    0.1348963E 07     0.1248534E 00     0.2511945E 00    -0.8924642E 00    -0.5999441E 00
0.6140000E 01    0.1380384E 07     0.1222295E 00     0.2254722E 00    -0.9129891E 00    -0.6469237E 00
0.6150000E 01    0.1412537E 07     0.1132910E 00     0.2026394E 00    -0.9458012E 00    -0.6882724E 00
0.6160000E 01    0.1445439E 07     0.1073328E 00     0.1859939E 00    -0.9694772E 00    -0.7305013E 00
0.6170000E 01    0.1479108E 07     0.1016262E 00     0.1722311E 00    -0.9929866E 00    -0.7639116E 00
0.6180000E 01    0.1513561E 07     0.9667590E-01     0.1564877E 00    -0.1014682E 01    -0.8055194E 00
0.6190000E 01    0.1548816E 07     0.9219236E-01     0.1449622E 00    -0.1035334E 01    -0.8389067E 00
```

Fig 6.

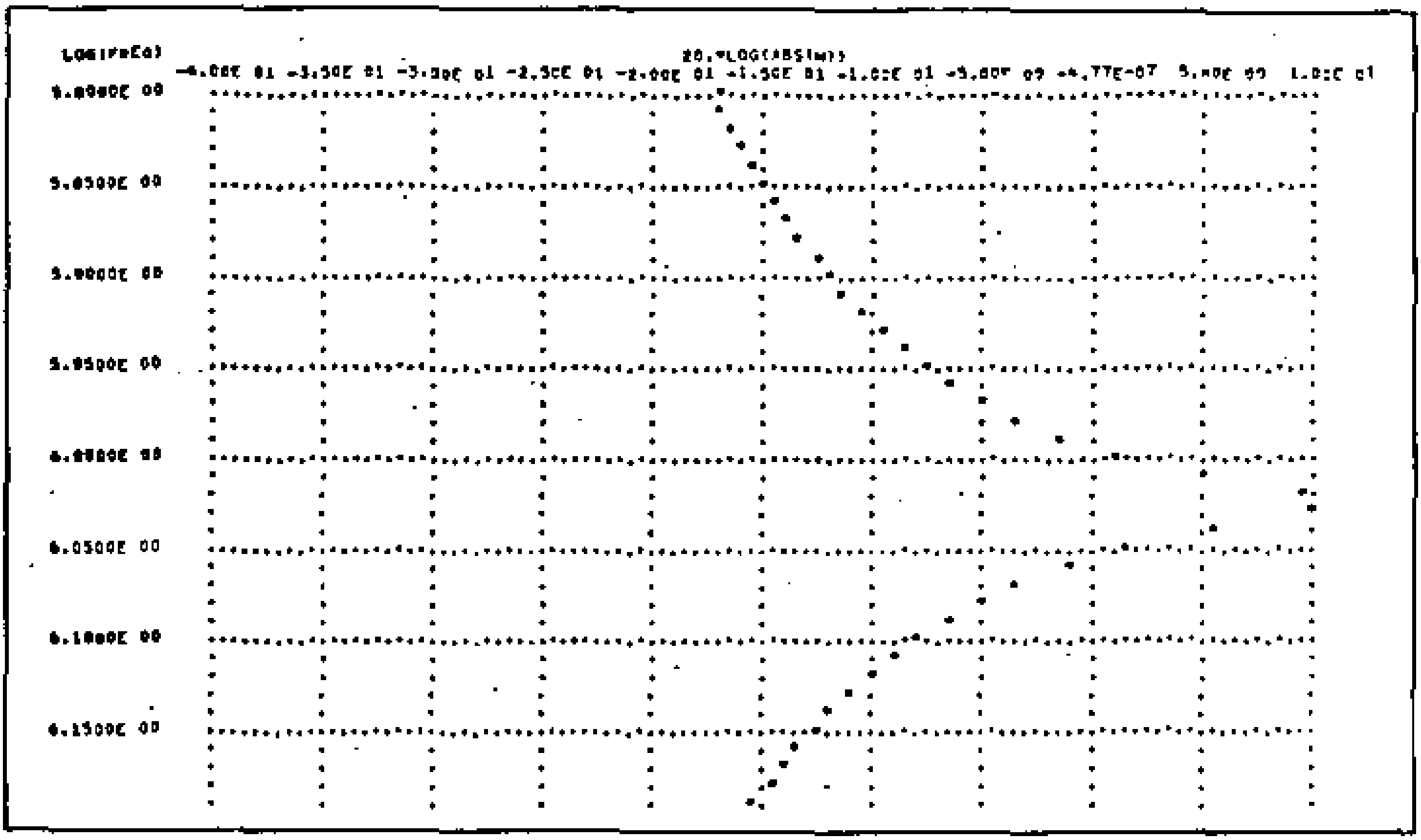

LOG(FREQ)
20.*LOG(ABS(W))
-4.00E 01
-3.50E 01
-3.00E 01
-2.50E 01
-2.00E 01
-1.50E 01
-1.00E 01
-5.00E 00
-4.77E-07
5.00E 00
1.00E 01
5.8000E 00
5.8500E 00
5.9000E 00
5.9500E 00
6.0000E 00
6.0500E 00
6.1000E 00
6.1500E 00

Fig 7.

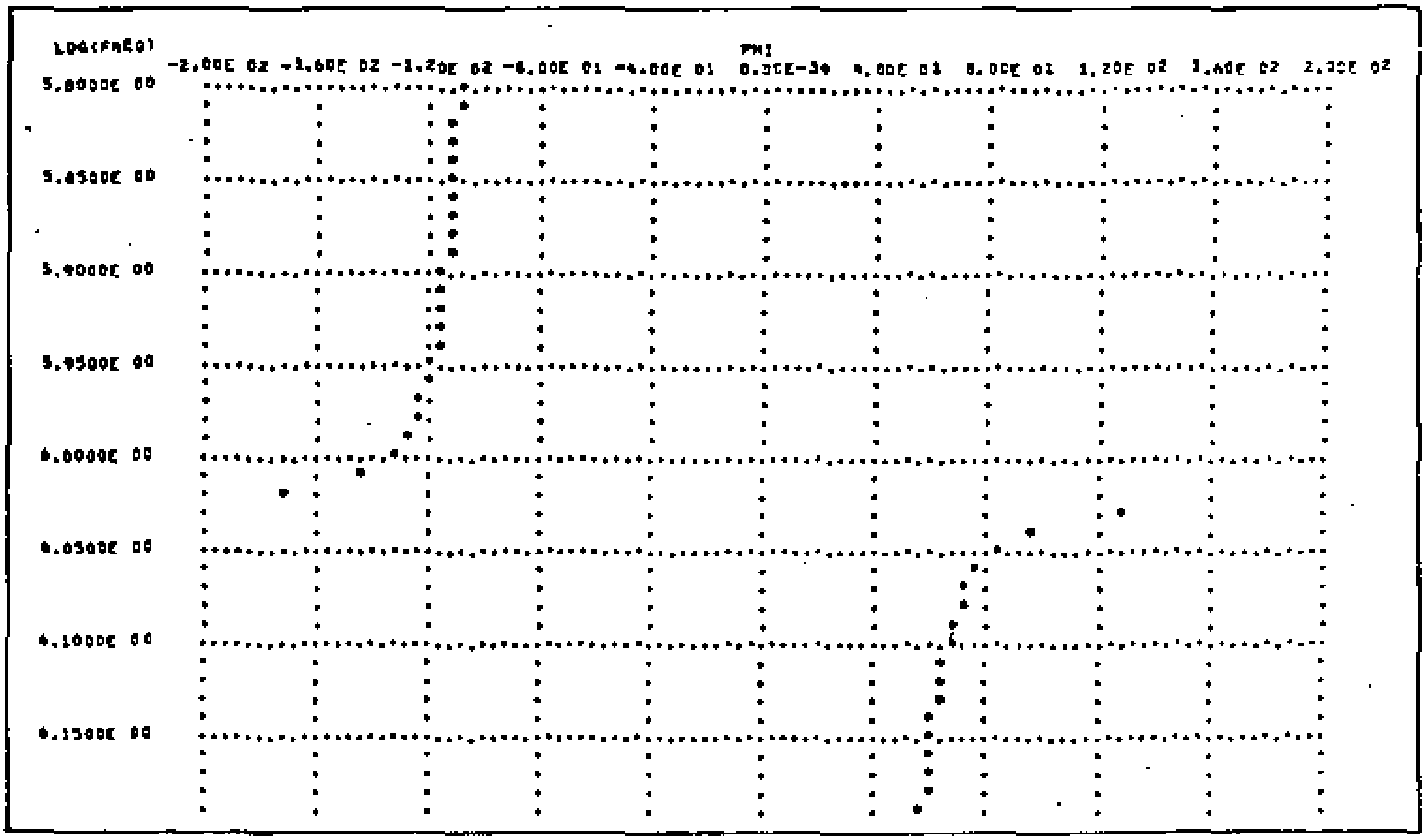

LOG(FREQ)
PHI
-2.00E 02
-1.60E 02
-1.20E 02
-8.00E 01
-4.00E 01
0.00E-34
4.00E 01
8.00E 01
1.20E 02
1.60E 02
2.00E 02
5.8000E 00
5.8500E 00
5.9000E 00
5.9500E 00
6.0000E 00
6.0500E 00
6.1000E 00
6.1500E 00

Fig 8.

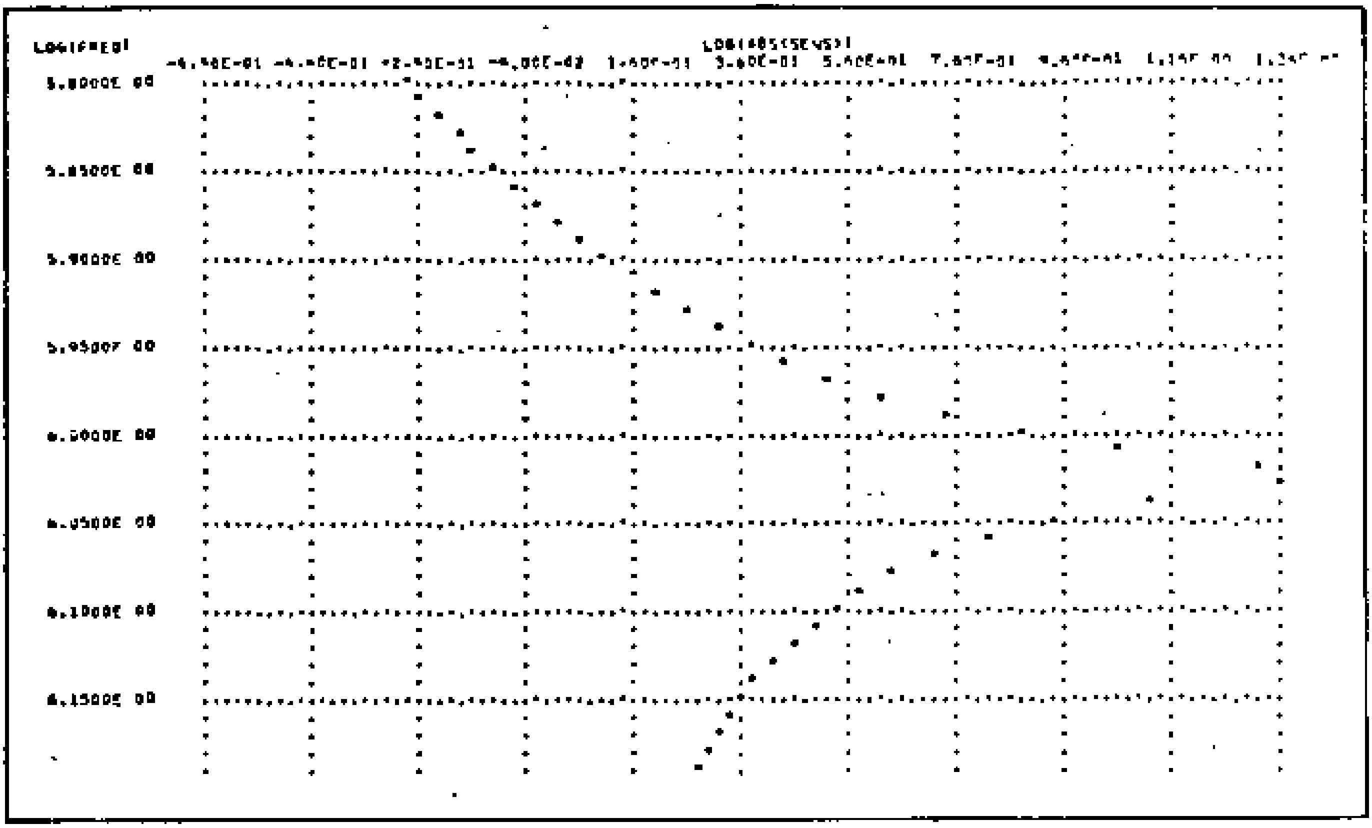
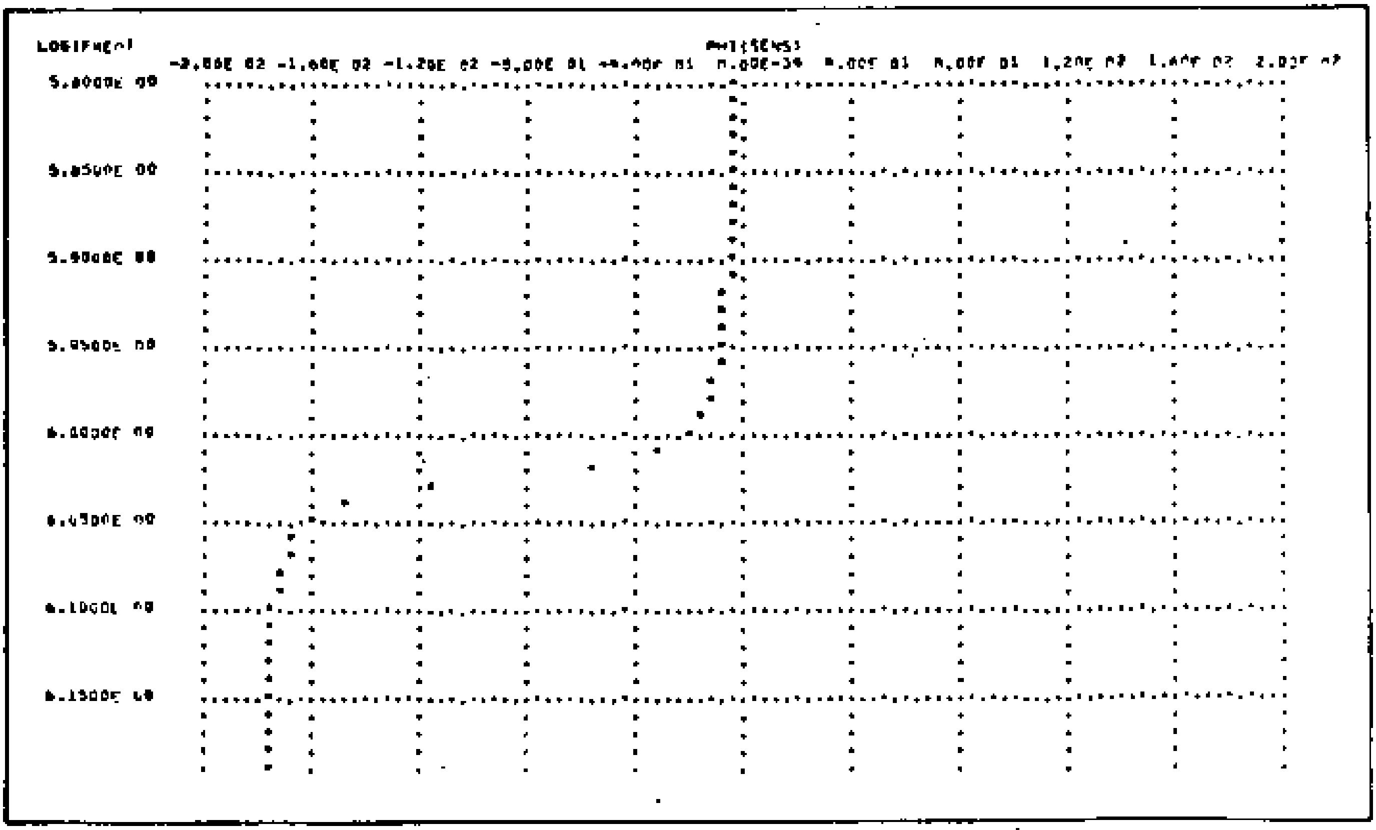

LOG(FREQ)
LOG(ABS(SENS))
-6.40E-01 -4.40E-01 -2.40E-01 -4.00E-02 1.60E-01 3.60E-01 5.60E-01 7.60E-01 9.60E-01 1.16E 00 1.36E 00
5.8000E 00
5.8500E 00
5.9000E 00
5.9500E 00
6.0000E 00
6.0500E 00
6.1000E 00
6.1500E 00

Fig 11.

Fig 12.


LOG(FREQ)
PHI(SENS)
-2.00E 02 -1.60E 02 -1.20E 02 -8.00E 01 -4.00E 01 0.00E-39 4.00E 01 8.00E 01 1.20E 02 1.60E 02 2.00E 02
5.8000E 00
5.8500E 00
5.9000E 00
5.9500E 00
6.0000E 00
6.0500E 00
6.1000E 00
6.1500E 00

sobre las frecuencias deseadas. Como deben dividirse entre 100 000 y están expresadas en logaritmos base 10, se les ha restado 5 (pues 10^5 = 100 000). Se ve que también aumentó el tiempo de solución a 4 seg (aun cuando el estrictamente equivalente sería 1 seg).

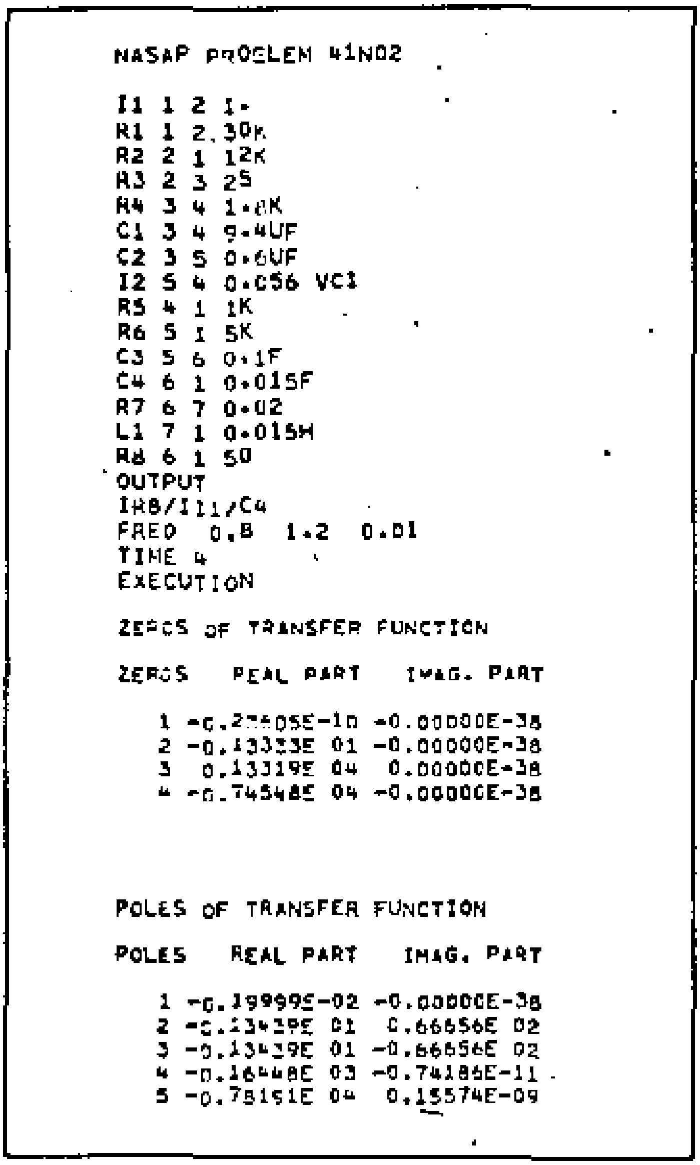
La nueva función de transferencia se muestra en la fig 14. Obsérvese que los coeficientes de los polinomios difieren en magnitud menos que los originales. En la nueva solución no aparecen mensajes de desborde numérico, por lo tanto, puede suponerse que los valores de los polos y ceros son correctos.

En la fig 15 se muestra la sensibilidad de las posiciones de dichos polos y ceros a cambios en *C4*. En la fig 16 se muestra en forma simbólica la respuesta transitoria cuando la excitación es un impulso. Las funciones complejas se pueden combinar con sus conjugadas para producir senoides amortiguadas reales. En la fig 17 la respuesta ha sido evaluada e impresa para 50 puntos intermedios entre cero y 4 seg. En la fig 18 se muestra una gráfica de la respuesta transitoria producida por el programa. (Las líneas rectas han sido trazadas a mano.)

3. DISCUSION

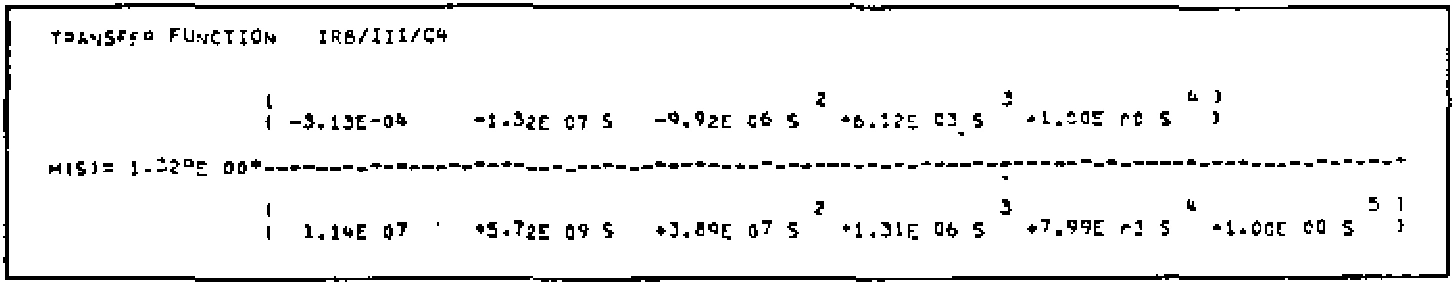
El programa aquí presentado puede ser muy útil en el diseño de circuitos. Para dicho diseño se comienza con un circuito inicial con valores aproximados para los parámetros. Subsecuentemente, con la intuición y experiencia del diseñador, se cambian dichos parámetros analizando los resultados en cada cambio con un programa como NASAP, hasta que el circuito satisfaga los requerimientos del diseño.

Aunque solamente una función de transferencia ha sido exhibida en el ejemplo dado, se pueden obtener tantas como se desee. Por lo tanto, a base de aplicar el programa repetidas veces, se pueden diseñar circuitos multiterminales con caracte-



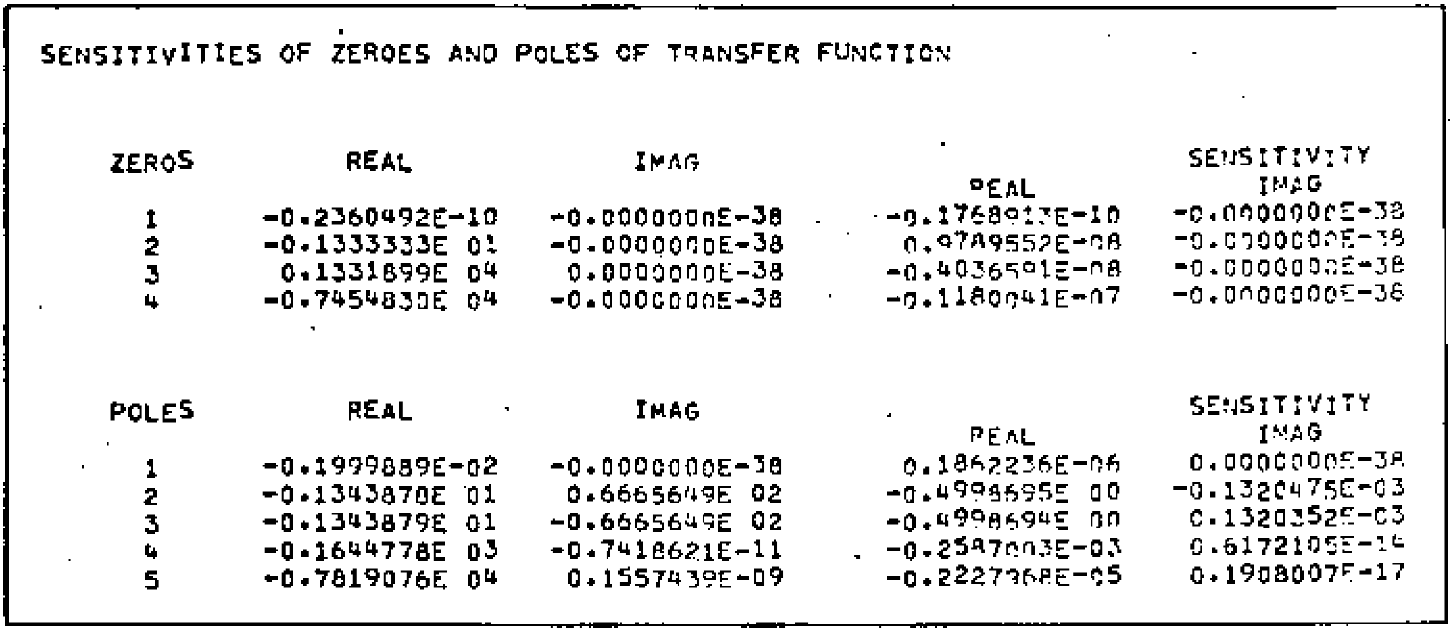
```
NASAP PROGLEM 41N02

I1 1 2 1.
R1 1 2.30K
R2 2 1 12K
R3 2 3 25
R4 3 4 1.8K
C1 3 4 9.4UF
C2 3 5 0.6UF
I2 5 4 0.C56 VC1
R5 4 1 1K
R6 5 1 5K
C3 5 6 0.1F
C4 6 1 0.015F
R7 6 7 0.02
L1 7 1 0.015H
R8 6 1 50
OUTPUT
IR8/I11/C4
FREQ  0.8  1.2  0.01
TIME 4
EXECUTION

ZEROS OF TRANSFER FUNCTION

ZEROS    REAL PART    IMAG. PART

   1 -0.27405E-10 -0.00000E-38
   2 -0.13333E 01 -0.00000E-38
   3  0.13319E 04  0.00000E-38
   4 -0.74548E 04 -0.00000E-38



POLES OF TRANSFER FUNCTION

POLES    REAL PART    IMAG. PART

   1 -0.19999E-02 -0.00000E-38
   2 -0.13439E 01  0.66656E 02
   3 -0.13439E 01 -0.66656E 02
   4 -0.16448E 03 -0.74186E-11
   5 -0.78151E 04  0.15574E-09
```

Fig 13.



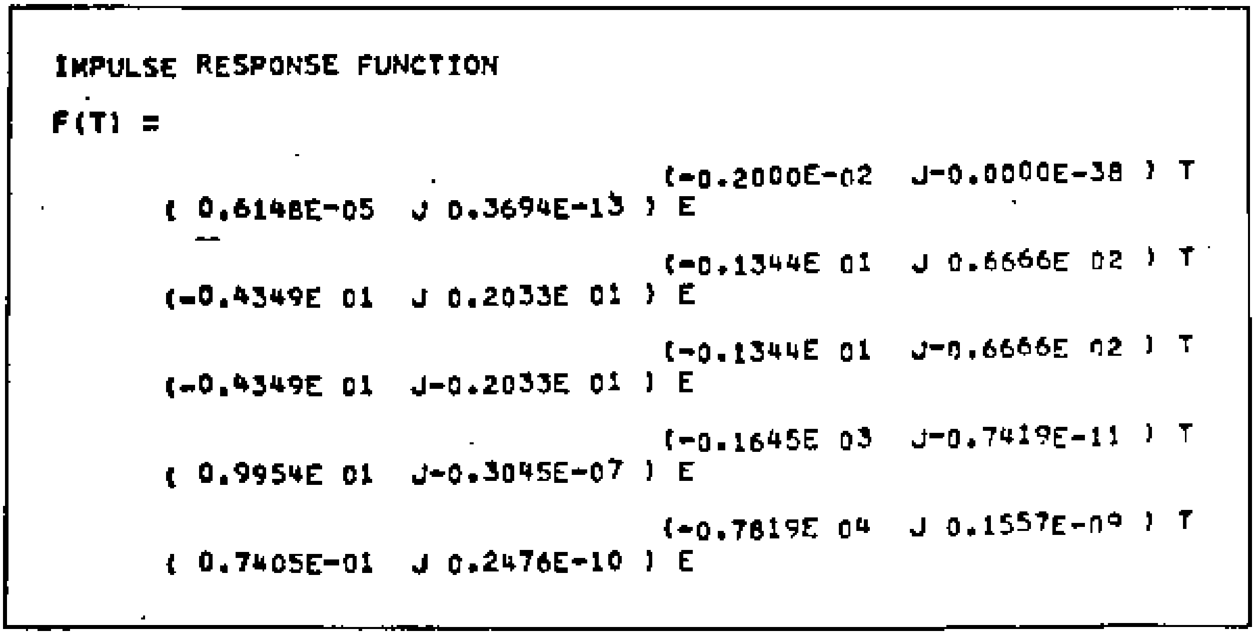
```
TRANSFER FUNCTION   IR8/I11/C4

                   (                                          2                3              4 )
                   ( -3.13E-04     +1.32E 07 S   -9.92E 06 S   +6.12E 03 S   +1.00E 00 S   )
H(S)= 1.329E 00*------------------------------------------------------------------------------------
                   (                                          2                3              4       5 )
                   (  1.14E 07     +5.72E 09 S   +3.89E 07 S   +1.31E 06 S   +7.99E 03 S   +1.00E 00 S   )
```

Fig 14.



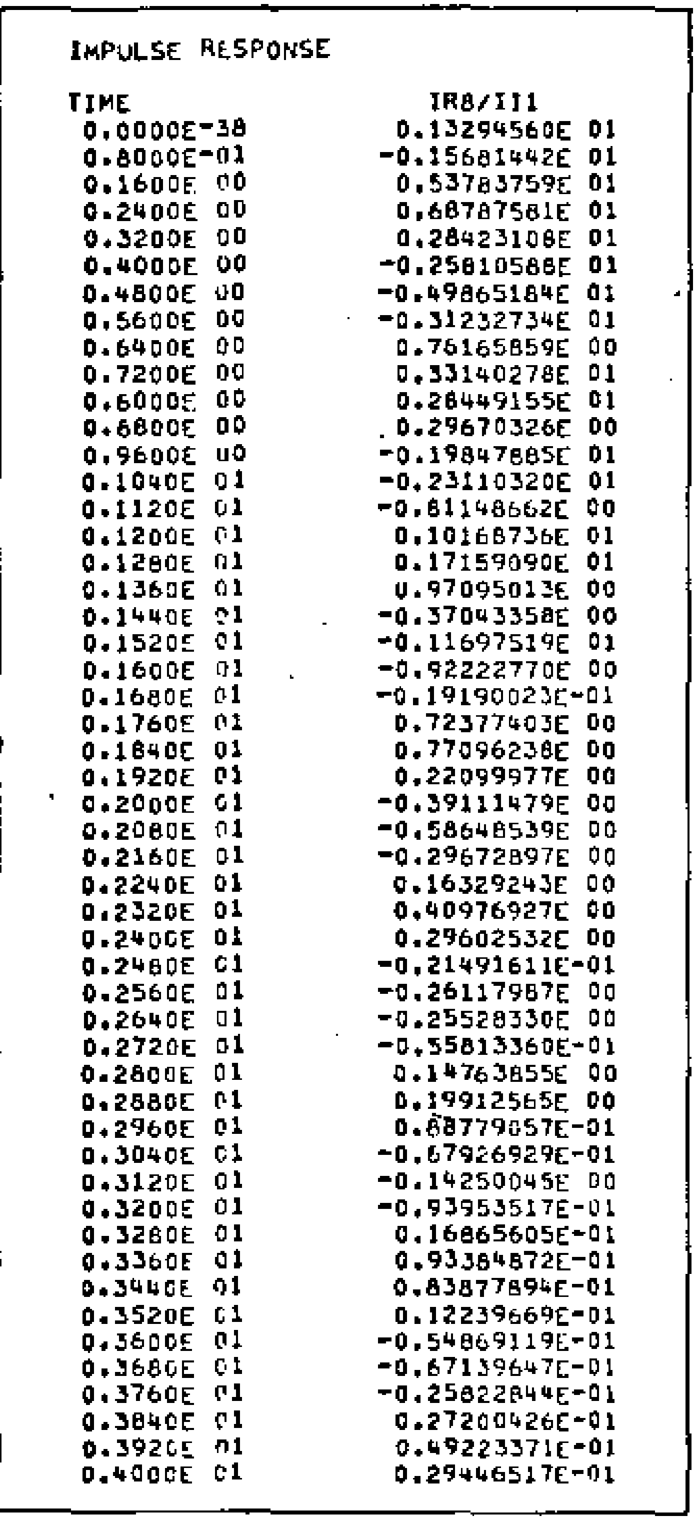
```
SENSITIVITIES OF ZEROES AND POLES OF TRANSFER FUNCTION


    ZEROS          REAL              IMAG                               SENSITIVITY
                                                          REAL              IMAG
      1       -0.2360492E-10    -0.0000000E-38      -0.1768913E-10    -0.0000000E-38
      2       -0.1333333E 01    -0.0000000E-38       0.9789552E-08    -0.0000000E-38
      3        0.1331899E 04     0.0000000E-38      -0.4036591E-08    -0.0000000E-38
      4       -0.7454830E 04    -0.0000000E-38      -0.1180041E-07    -0.0000000E-36


    POLES          REAL              IMAG                               SENSITIVITY
                                                          REAL              IMAG
      1       -0.1999889E-02    -0.0000000E-38       0.1862236E-06     0.0000000E-38
      2       -0.1343878E 01     0.6665649E 02      -0.4998695E 00    -0.1320475E-03
      3       -0.1343879E 01    -0.6665649E 02      -0.4998694E 00     0.1320352E-03
      4       -0.1644778E 03    -0.7418621E-11      -0.2587003E-03     0.6172105E-14
      5       -0.7819076E 04     0.1557439E-09      -0.2227368E-05     0.1908007E-17
```

Fig 15.



```
IMPULSE RESPONSE FUNCTION

F(T) =
                                            (-0.2000E-02   J-0.0000E-38 ) T
        ( 0.6148E-05   J 0.3694E-13 ) E
                                            (-0.1344E 01   J 0.6666E 02 ) T
        (-0.4349E 01   J 0.2033E 01 ) E
                                            (-0.1344E 01   J-0.6666E 02 ) T
        (-0.4349E 01   J-0.2033E 01 ) E
                                            (-0.1645E 03   J-0.7419E-11 ) T
        ( 0.9954E 01   J-0.3045E-07 ) E
                                            (-0.7819E 04   J 0.1557E-09 ) T
        ( 0.7405E-01   J 0.2476E-10 ) E
```

Fig 16



```
IMPULSE RESPONSE

TIME                        IR8/II1
 0.0000E-38                  0.13294560E 01
 0.8000E-01                 -0.15681442E 01
 0.1600E 00                  0.53783759E 01
 0.2400E 00                  0.68787581E 01
 0.3200E 00                  0.28423108E 01
 0.4000E 00                 -0.25810588E 01
 0.4800E 00                 -0.49865184E 01
 0.5600E 00                 -0.31232734E 01
 0.6400E 00                  0.76165859E 00
 0.7200E 00                  0.33140278E 01
 0.6000E 00                  0.28449155E 01
 0.6800E 00                  0.29670326E 00
 0.9600E 00                 -0.19847885E 01
 0.1040E 01                 -0.23110320E 01
 0.1120E 01                 -0.81148662E 00
 0.1200E 01                  0.10168736E 01
 0.1280E 01                  0.17159090E 01
 0.1360E 01                  0.97095013E 00
 0.1440E 01                 -0.37043358E 00
 0.1520E 01                 -0.11697519E 01
 0.1600E 01                 -0.92222770E 00
 0.1680E 01                 -0.19190023E-01
 0.1760E 01                  0.72377403E 00
 0.1840E 01                  0.77096238E 00
 0.1920E 01                  0.22099977E 00
 0.2000E 01                 -0.39111479E 00
 0.2080E 01                 -0.58648539E 00
 0.2160E 01                 -0.29672897E 00
 0.2240E 01                  0.16329243E 00
 0.2320E 01                  0.40976927E 00
 0.2400E 01                  0.29602532E 00
 0.2480E 01                 -0.21491611E-01
 0.2560E 01                 -0.26117987E 00
 0.2640E 01                 -0.25528330E 00
 0.2720E 01                 -0.55813360E-01
 0.2800E 01                  0.14763855E 00
 0.2880E 01                  0.19912565E 00
 0.2960E 01                  0.68779057E-01
 0.3040E 01                 -0.67926929E-01
 0.3120E 01                 -0.14250045E 00
 0.3200E 01                 -0.93953517E-01
 0.3280E 01                  0.16865605E-01
 0.3360E 01                  0.93384872E-01
 0.3440E 01                  0.83877894E-01
 0.3520E 01                  0.12239669E-01
 0.3600E 01                 -0.54869119E-01
 0.3680E 01                 -0.67139647E-01
 0.3760E 01                 -0.25822844E-01
 0.3840E 01                  0.27200426E-01
 0.3920E 01                  0.49223371E-01
 0.4000E 01                  0.29446517E-01
```

Fig 17.

rísticas en los puertos especificadas por el diseñador. Si estas características se escogen adecuadamente, se pueden obtener, para un circuito cualquiera, las matrices de admitancia o impedancia indefinidas. Una vez que se tienen estas matrices, es posible iniciar el diseño de circuitos microelectrónicos formados por varios subcircuitos multiterminales interconectados. Un programa que hace todo esto automáticamente es el BELNAP (ref 23); con él se pueden analizar circuitos microelectrónicos muy extensos, a condición de hacerlo por partes (ref 64).

El mismo programa es útil para el diseño de circuitos de microondas (ref 67). También se puede usar dicho programa, haciendo combinaciones, para determinar fallas internas en un circuito multiterminal. Para este propósito se han desarrollado métodos matemáticos adecuados (ref 65). Cuando se trata de circuitos con relativamente pocas terminales, es más adecuado el método del cálculo de las desviaciones en lugar de las matrices o los métodos topológicos (ref 66).

No obstante que NASAP y BELNAP son muy poderosos y útiles, no es posible hacer con ellos ciertos tipos de análisis (por ejemplo análisis estadístico o transitorio de circuitos no lineales). Para ello es necesario usar otros programas (refs 24 y 26 a 28), lo cual ocasiona una labor adicional, pues habrá necesidad de codificar el circuito de acuerdo con las reglas especiales de cada programa.

Aunque sería muy conveniente desarrollar un programa para todos los tipos de análisis posibles, esto es sumamente difícil, pues los métodos matemáticos que se aplican en un caso no son válidos para otro. Es mejor desarrollar un compilador especial que sirva de puente entre el usuario y cada programa diferente de análisis. Así, un circuito no tendrá que ser codificado varias veces, aun cuando se requieran diversos tipos de análisis, con lo que se elimina la fuente de muchos errores. El compilador tendrá un idioma universal y traducirá a los idiomas de los programas de análisis. La presentación de resultados se hará en el lenguaje universal tras una traducción de los lenguajes particulares. Dicho compilador lo está desarrollando, en la Universidad de California, en Los Angeles, el coautor de este artículo.

4. FUTURO

El futuro de la rama de diseño de circuitos con computadora y áreas relacionadas, tales como localización de fallas, documentación y producción automática, es sumamente brillante. Con respecto al diseño se nota una fuerte tendencia a utilizar más y

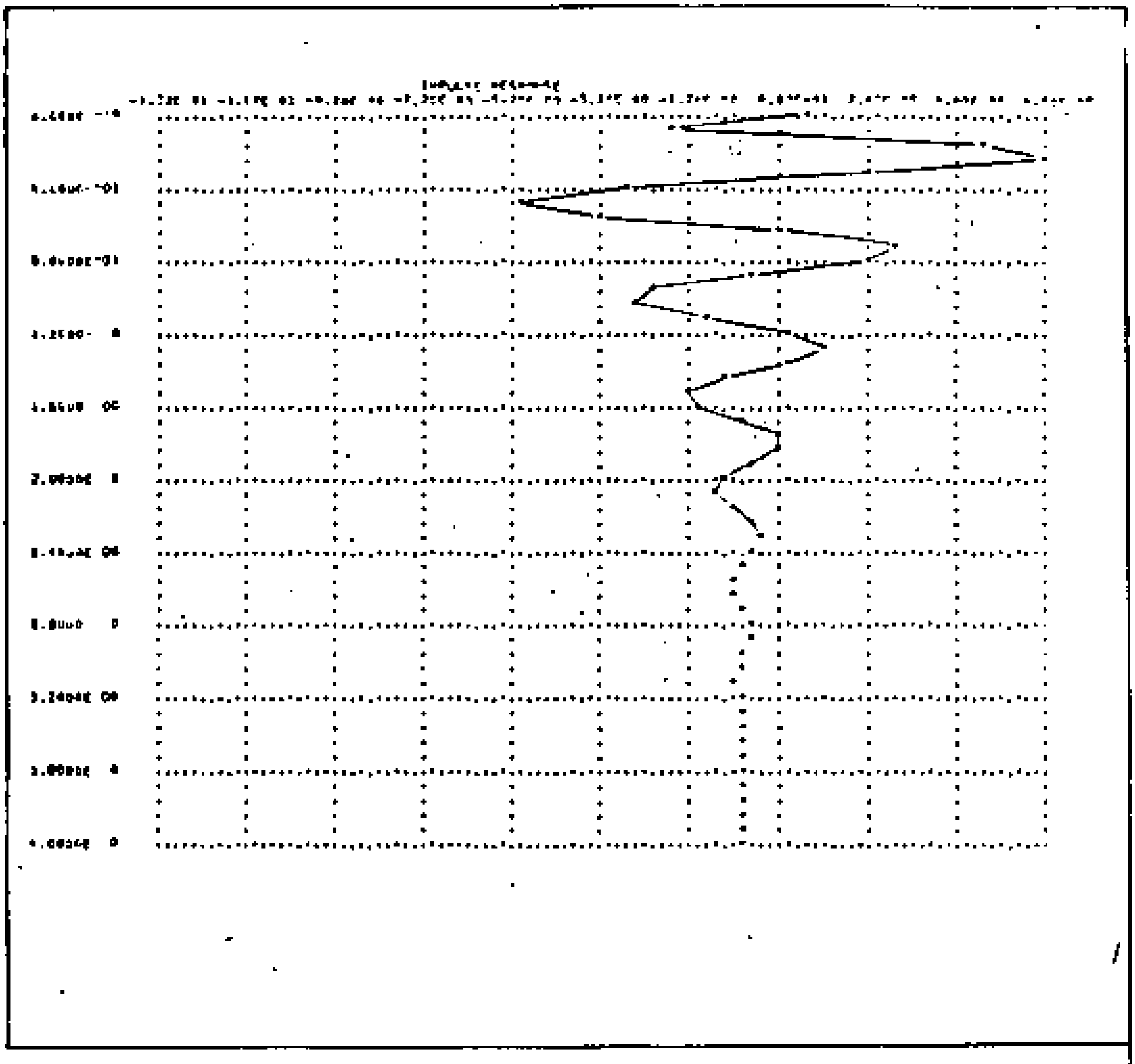


Fig 18.

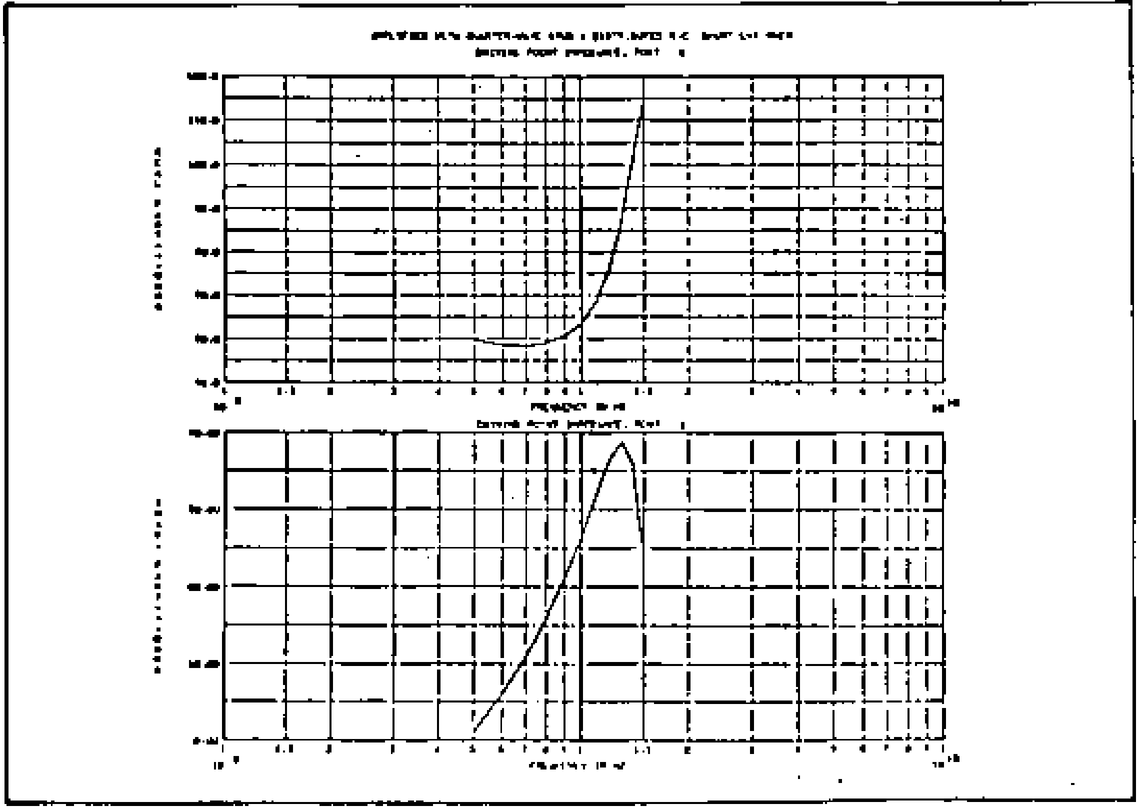


Fig 19.

más el tiempo compartido de operación en lugar de la operación normal. También se ve claro que la comunicación con la computadora será cada vez más humana y menos orientada a los especialistas en computadoras y programación. En este sentido se verán más programas en que gran parte de la comunicación se hará a base de tubos de rayos catódicos con "lápices luminosos". Así, el usuario, en lugar de codificar, dibujará su circuito sobre una pantalla de televisión y dará las órdenes a la computadora "apretando botones luminosos". Obtendrá las respuestas en forma gráfica y fácil de interpretar. Apretando un botón luminoso puede ordenar al programa que efectúe dibujos (fig 19), gráficas y tablas finales del diseño. También podrá pedir a la máquina que determine listas de materiales y precios para la fabricación del diseño. Con esto se eliminará mucho trabajo rutinario. Todo lo aquí mencionado ya está en operación en muchas plantas de fabricación de microcircuitos.

Se vislumbra ya que, en un futuro muy cercano, se podrá automatizar desde la concepción del circuito hasta el sistema final. Hasta ahora se han automatizado la mayor parte de los pasos, por ejemplo, el diseño lógico y eléctrico, la documentación y los dibujos, así como la generación de mascarillas para la deposición por métodos ópticos y químicos, y el alambrado prueba de los circuitos y sistemas. Cuando se logre conquistar un par de eslabones más, se podrá realizar la completa automatización del proceso. En dicha automatización, la computadora digital desempeña el papel principal.

Como gran parte de los microcircuitos que se fabrican hoy en día son precisamente para la construcción de computadoras, en un futuro cercano será realidad aquel sueño de los escritores de ciencia ficción: máquinas que se reproducen.

Ese sueño hoy en día no es una aberración, pues es cierto, en el sentido estricto de la frase, que las computadoras modernas se diseñan y construyen, en gran parte, ellas mismas.

5. REFERENCIAS

1. G. Kron, *Tensor Analysis of Networks*, John Wiley & Sons, Inc., Nueva York (1939)

2. P. Le Corbeiller, *Matrix Analysis of Electrical Networks*, Harvard University Press, Cambridge, Massachusetts (1950)

3. F. H. Branin Jr., *The Relation Between Kron's Method and the Classical Methods of Network Analysis*, IRE WESCON Convention Record, parte 2 (ago 1959), pp. 3 a 28

4. T. R. Bashkow, *The A Matrix, New Network Concept*, IRE Trans. on Circuit Theory, Vol CT-4 (1957), pp. 117 a 120

5. P. R. Bryant, *The Explicit Form of Bashkow's A-Matrix*, IRE Trans. on Circuit Theory, Vol. CT-9 (1962), pp. 303 a 306

6. F. H. Branin Jr., *DC and Transient Analysis of Networks Using a Digital Computer*, IRE International Convention Record, Vol 10, parte 2 (1962) pp. 236 a 256

7. A. Dervisoglu, *Bashkow's A-Matrix for Active RLC Networks*, IEEE Trans. on Circuit Theory, Vol CT-11 (1964), pp. 404 a 406

8. R. J. Duffin, *Nonlinear Networks*, Boletin American Mathematical Society, Vol 2a (oct 1947), pp. 963 a 971

9. G. J. Minty, *Solving Steady-State Nonlinear Networks of Monotone Elements*, Trans. IRE, Vol 8 (1961), pp. 99 a 104

10. J. Katzenelson, *An Algorithm for Solving Nonlinear Resistive Networks*, Bell System Technical Journal, Vol 44 (nov 1965), pp. 1605 a 1620

11. T. E. Stern, *The Theory of Nonlinear Networks and Systems*, Addison-Wesley Publishing Co., Inc., Reading, Massachusetts (1965)

12. C. A. Desoer y J. Katzenelson, *Nonlinear RLC Networks*, Bell System Technical Journal, Vol 54, No 1 (1965), pp. 161 a 198

13. J. Katzenelson, *AEDNET: A Simulator for Nonlinear Networks*, Proc. IEEE, Vol 54 (nov 1966), pp. 1536 a 1552

14. W. S. Percival, *Solution of Passive Electrical Networks by Means of Mathematical Trees*, Journal of the Institution of Electrical Engineers, Vol 100, parte 2, Londres (1953), pp. 143 a 150

15. W. S. Percival, *Graphs of Active Networks,* Institution of Electrical Engineers, Monografia No 129, Londres (1955)

16. S. J. Mason, *Topological Analysis of Linear Non-Reciprocal Networks,* Proc. IRE, Vol 45 (jun 1957), pp. 829 a 838

17. C. L. Coates, *General Topological Formulas for Linear Network Functions,* Trans. IRE, Vol CT-5 (mar 1958), pp. 30 a 42

18. W. Mayeda y M. E. Van Valkenburg, *Analysis of Non-Reciprocal Networks by Digital Computers,* IRE National Convention Record, Vol 6, parte 2 (1958), pp. 70 a 75

19. W. W. Happ, *Flowgraph Techniques for Closed Systems,* IEEE Trans. on Aerospace and Electronic Systems, Vol 3 (1966), pp. 252 a 264

20. T. J. Kobylarz, *Computer Determination of Symbolic State Equations for Nonlinear Circuits,* Proc. International Conference on Computer-Aided Design, Southampton, Inglaterra (abr 1969), pp. 415 a 425

21. T. Roska, *Generating Network Functions in Literal Form by Digital Computers,* Summer School on Circuit Theory, Praga (1968)

22. J. W. Cooley y J. W. Tukey, *An Algorithm for the Machine Calculation of Complex Fourier Series,* Math. of Computation, Vol 19 (1965), pp. 297 a 301

23. M. A. Murray Lasso, *Analysis of Linear Integrated Circuits by Digital Computer Using Black-Box Techniques,* Chapter 4 in Computer-Aided Integrated Circuit Design, G. J. Herskowitz, McGraw-Hill Book Co., Nueva York (1968)

24. *1620 Electronic Circuit Analysis Program (ECAP), Application Program 1620-EE-02X,* Data Processing Division, IBM Corp., White Plains, Nueva York

25. L. P. McNamee y P. Potash, *A User's Guide and Programmer's Manual for NASAP,* Dept. of Engineering, informe No. 68-38, Universidad de California, Los Angeles (ago 1968)

26. A. F. Malmberg, F. L. Cornwell y F. N. Hofer, *NET-1 Network Analysis Program,* Report LA-3119, Los Alamos Scientific Laboratory, Los Alamos, N. M. (1964)

27. L. D. Milliman, W. A. Massena, R. H. Rickhaut y A. C. Mong, *CIRCUS: A Digital Computer Program for Transient Analysis of Electronic Circuits,* Vol 1 y 2, The Boeing Co., Wáshington (ene 1967)

28. H. W. Mathers, S. R. Sedore y J. R. Senta, *Automated Digital Computer Program for Determining Responses of Electronic Circuits to Transient Nuclear Radiation (SCEPTRE),* Vol 1 y 2, IBM Technical Report No AFWL-TR 66-126, Air Force Weapons Lab. (feb 1967)

29. G. K. Pritchard, *A Survey of Transient Analysis Programs,* Proc. Computer Aided Circuit Design Seminar, NASA/ERC, Cambridge, Massachusetts (1967)

30. F. F. Kuo, *Network Analysis by Digital Computer,* Proc. IEEE, Vol 54, No 6 (jun 1966), pp. 820 a 829

31. O. E. Meyerhoff y L. P. McNamee, *Considerations for Optimum General Circuit Analysis Program Application,* Proc. Sixth Annual Allerton Conference on Circuit and Systems Theory, Monticell, 3 (oct 1968), pp. 396 a 405

32. H. A. Spang, *A Review of Minimization Techniques for Non-Linear Functions,* SIAM Review, Vol 4 (1962), pp. 343 a 365

33. G. C. Temes y D. A. Calahan, *Computer-Aided Network Optimization: The State of the Art,* Proc. IEEE, Vol 55, No 11 (nov 1967), pp. 1832 a 1863

34. E. B. Kozemchak y M. A. Murray Lasso, *Computer Aided Circuit Design by Singular Imbedding,* Bell System Technical Journal, Vol 48, No 1 (ene 1969), pp. 275 a 315

35. D. T. Ross y J. E. Rodríguez, *Theoretical Foundations for the Computer-Aided Design System,* 1963 Spring Joint Computer Conference, Proc. AFIPS, Vol 23, Spartan Books, Inc., Wáshington (1963)

36. D. S. Evans, y J. Katzenelson, *Data Structure and Man-Machine Communication for Network Problems,* Proc. IEEE, Vol 55, No 7 (jul 1967), pp. 1135 a 1144

37. H. C. So, *Analysis and Iterative Design of Networks Using On-Line Simulation,* en la ref 38

38. F. F. Kuo y J. F. Kaiser, *System Analysis by Digital Computer,* John Wiley & Sons, Inc., Nueva York (1966)

39. G. J. Herskowitz, *Computer-Aided Integrated Circuit Design,* McGraw-Hill Book Co., Nueva York (1968)

40. D. A. Calahan, *Computer-Aided Network Design,* edición preliminar, McGraw-Hill Book Co., Nueva York (1968)

41. R. W. Jensen y M. D. Lieberman, *IBM Electronics Circuit Analysis Program: Techniques and Applications,* Prentice-Hall Inc., Englewood Cliffs, Nueva Jersey (1968)

42. G. W. Zobrist, *Network Computer Analysis - ECAP, NASAP, NET-1, SCEPTRE and Modeling*, Boston Technical Publishers, Inc., Cambridge, Massachusetts (1968)

43. *Special Issue on Computer-Aided Design*, Proceedings IEEE, Vol 55, No 11 (nov 1967)

44. *Special Issue on Computer Oriented Microwave Practices*, IEEE Transactions on Microwave Theory and Techniques, Vol MTT-17, No 8 (ago 1969)

45. *Special Issue on Educational Aspects of Circuit Design by Computer*, IEEE Transactions on Education, Vol E-12, parte 1, No 3 (sept 1969), parte 2, No 4 (dic 1969)

46. *Circuit Design By Computer Symposium*, Tutorial Symposium, Universidad de Nueva York, Nueva York (ene 31, feb 2, 1967)

47. *Computer-Aided Circuit Design Seminar*, NASA Electronics Research Center, Cambridge, Massachusetts (abr 1967)

48. *Conferencia sobre análisis de circuitos con computadora digital*, UNAM, México (jun 1967)

49. *Conferencia sobre análisis de circuitos con computadora digital*, Instituto Tecnológico y de Estudios Superiores, Monterrey, México (jul 4 y 5, 1967)

50. P. R. Mayaguez, *Conferencia sobre el uso de las computadoras en el diseño eléctrico*, Universidad de Puerto Rico (sep 1967)

51. *Computer-Aided Integrated Circuit Design Course*, Instituto Tecnológico de Stevens (sep 11 a 15, 1967)

52. *Automated Circuit Analysis Course*, Universidad de California, Los Angeles (abr 3 a 7, 1967)

53. *Computer-Aided Design Workshop*, Universidad de Missouri, Columbia, Missouri (ago 7 a 18, 1967)

54. *Computer Oriented Circuit Design: A Simbolic Approach*, Universidad de California, Los Angeles (ene 22 a 26, 1968)

55. *Design Automation Workshops*, Annual Meetings on Computer Aided Design, ACM, SHARE y IEEE

56. *Computer Aids for Large Circuit Arrays*, Universidad de Wisconsin, Madison, Wisconsin (sep 27 a 28, 1967)

57. *Computer-Aided Circuit Design*, A Compiler Oriented Approach, Universidad de California, Los Angeles (feb 3 a 7, 1969)

58. *International Conference on Computer Aided Design*, IEEE (Londres) Southampton, Inglaterra (abr 15 a 18, 1969)

59. *Computer Aided Integrated Circuit Design*, IEEE New Technical and Scientific Activities Committee, Hoboken, Nueva Jersey (jun 5, 1967)

60. *Annual Summer Institute on Computer-Aided Circuit Analysis and Design*, Universidad de Missouri, Columbia, Missouri (ago 5 a 16, 1968)

61. *Computerized Electronics*, Universidad Cornell, Ithaca, Nueva York (ago 26 a 28, 1969)

62 *CODING INSTRUCTIONS FOR NASAP 69/I (Network Analysis for Systems Applications Program)* Revision No 1, Gaertner Research Incorporated, Stamford, Connecticut (ene 8, 1969)

63. E. A, Guillemin, *Theory of Linear Physical Systems*, John Wiley and Sons, Inc., Nueva York (1963)

64. M. A. Murray Lasso, *Black-Box Models for Linear Integrated Circuits*, IEEE Transactions on Education, Vol E-12, No 3 (sep 1969), pp. 170 a 180

65. W. W. Happ, *Combinatorial Analysis of Multi-terminal Devices*, IEEE Transactions on Systems Science and Cybernetics, Vol SSC-3, No 1 (jun 1967), pp. 21 a 27

66. T. R. Nisbit y W. W. Happ, *The Calculus of Deviations Applied to Transistor and Network Analysis*, J. British IRE, Vol 21 (1961), pp. 437 a 450

67. M. A. Murray Lasso y E. B. Kozemchak, *Microwave Circuit Design by Digital Computer*, IEEE Transactions on Microwave Theory and Techniques, Vol MTT-17, No 8 (ago 1969), pp 514 a 526

68. M. A. Murray Lasso y E. B. Kozemchak, *Foundations of Computer-Aided Design by Singular Imbedding*, IEEE Transactions on Circuit Theory, Vol CT-17, No 1 (feb 1970), pp 105 a 112

MICROWAVE CIRCUIT DESIGN BY DIGITAL COMPUTER

BY

M. A. MURRAY-LASSO AND E. B. KOZEMCHAK

Reprinted from IEEE TRANSACTIONS
ON *MICROWAVE THEORY AND TECHNIQUES*
Volume MTT-17, Number 8, August, 1969
pp. 514-526

Microwave Circuit Design by Digital Computer

MARCO A. MURRAY-LASSO, MEMBER, IEEE, AND EDWARD B. KOZEMCHAK

Abstract—Methods for the automatic analysis and design of microwave circuits using a digital computer in batch mode are given. The methods are capable of handling microwave components modeled by ordinary R, L, C, M, CS elements plus transmission lines and multiterminal black-boxes whose characteristics have been determined theoretically or experimentally. The analysis-optimization program, IMPROVE (Integrated and Microwave Program for Optimizing Variable Elements), implementing the methods presented in this paper is described and its use illustrated with a practical design problem.

I. Introduction

CONSIDERABLE progress has been made in the field of analysis of electronic circuits by digital computer in the last five years. Presently, several general purpose computer programs for dc, ac, and transient analysis of linear and nonlinear circuits are widely available [1]-[6]. The design of a circuit is a trial and error process generally accomplished by repeated analysis and parameter or structure changes done by the designer until the response of a circuit is sufficiently close to the desired response. This requires several approaches to the computer and careful evaluation of the results of each run. The procedure can be very time consuming.

A portion of this process can be automated so that the computer does the evaluation and parameter changes and the designer decides only on the structure and the values of a set of fixed parameters. Several researchers have reported various degrees of success in applying these techniques to lumped circuits [7]-[12]. The automatic parameter changes are accomplished by converting the design problem into the problem of minimizing a function of several variables subject to a set of inequality constraints. To this problem a number of mathematical techniques may be applied [13]-[15]. By automating the analysis, evaluation, and parameter changes portion of the design process, the time to complete a design can be shortened considerably.

The methods mentioned above have not been widely applied to microwave circuits due to the inability of the analysis programs to conveniently handle distributed circuits or circuits that have been characterized empirically. Hence, automatic circuit design has been largely restricted to low frequency applications.

In this paper some methods appropriate for microwave circuits are presented and a program implementing them is

Manuscript received March 10, 1969; revised May 8, 1969. This work was partially supported by NASA Electronics Research Center under Grant NAS 12-2101.

M. A. Murray-Lasso is with Systems Research Center, Case Western Reserve University, Cleveland, Ohio.

E. B. Kozemchak is with Bell Telephone Laboratories, Whippany, N. J.

described. IMPROVE (Integrated and Microwave PRogram for Optimizing Variable Elements) is a batch program capable of automatically designing integrated and microwave networks in the frequency domain. It couples a general purpose analysis program BELNAP [16] with a direct pattern search optimization scheme. The user inputs a network description specifying variable parameters and their limits plus a desired performance. The optimization seeks new parameter values to minimize a weighted mean-squared error criterion. The output of the program is a final set of parameter values which produces a performance as "close as possible" to the desired one. The use of the program is illustrated with an example.

II. Automatic Design

For some design requirements synthesis procedures exist to determine both topology and parameter values [17]–[22]. For many requirements, however, no straightforward synthesis exists; this is true particularly in networks containing active devices at sufficiently high frequencies so that simple low frequency models give poor approximations to real behavior. For these cases the designer's experience and intuition are brought to bear in choosing a circuit to meet the requirements. A cut and try procedure is usually adopted in order to determine the appropriate parameter values to meet the requirements.

The design process using the cut and try scheme is indicated in Fig. 1. The operations done best by man are marked M, those done best by computer are marked C and those done by either or both combined are marked MC. The problem is created when a system need is recognized and defined. (For example, it is decided to amplify signals in a specified frequency band.) The need is then expressed as a circuit. (It is decided to build a 20 dB narrow-band amplifier.) Next the objectives are expressed in circuit terms. (The transducer gain must be a constant of value 10 over the band of interest; the input impedance must be 50 ohms in the band of interest.) Next a candidate schematic is chosen. (Active elements for amplification are selected. Circuitry for biasing the active elements and for matching the input impedance is added. Elements for producing a flat maximum of sufficient magnitude at the center frequency of interest are added.) An initial parameter guess is made. (Actual values for the parameters of both the active and passive elements are inserted. This may involve making educated guesses about parasitics or coming up with parameters such as lengths and position of stubs from simplified calculations.) An analysis of the circuit is made with the aid of an analysis program. The behavior of the circuit is compared with that of an ideal circuit meeting all the objectives. (The transducer gain and input impedance are calculated for several frequencies in the band of interest and compared with the values specified.) If the circuit meets the objective the design is finished (quite unlikely on the first trial) and a breadboard is built. If the objectives are not met and a preset maximum number of trials or schematics has not been exceeded, the parameter values are changed by the designer using as much experience and intuition as possible. (For example, the parameters may be positions and lengths of stubs.) The change should reduce the difference between the actual performance and the desired performance. (For example, if the gain is low for some frequencies, the change should raise it.) A new analysis of the circuit reveals that indeed the new set of parameters does give improved performance. The loop "analysis of circuit – comparison of real and ideal circuits – change parameter values" is traversed many times until either the objectives are met with sufficient approximation or too many trials have been attempted. If the latter is the case the designer may try a new candidate circuit and repeat the process. It may happen that the number of schematics tried unsuccessfully exceeds a preset maximum. The designer then uses the outermost loop and changes the circuit objectives (for example, reduces the gain requirement). Additional loops could be incorporated but are not shown in Fig. 1 for simplicity.

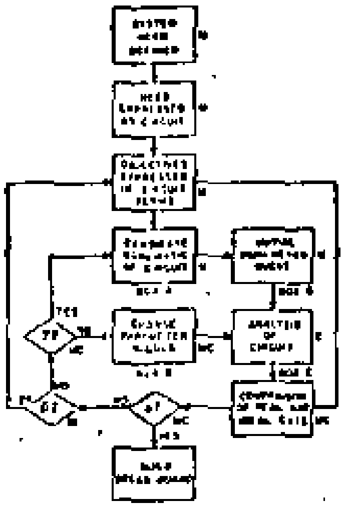

SYSTEM NEED DEFINED
M
NEED EXPRESSED AS CIRCUIT
M
OBJECTIVES EXPRESSED IN CIRCUIT TERMS
M
CANDIDATE SCHEMATIC OF CIRCUIT
M
BOX A
INITIAL PARAMETER GUESS
M
BOX B
YES
γ
NO
CHANGE PARAMETER VALUES
MC
BOX B
ANALYSIS OF CIRCUIT
C
BOX C
NO
β
YES
NO
α
MC
YES
COMPARISON OF REAL AND IDEAL CIRCUITS
MC
BUILD BREADBOARD
α = DOES CIRCUIT MEET OBJECTIVES
β = HAVE ENOUGH SCHEMATICS BEEN TRIED
γ = HAVE SUFFICIENT TRIALS BEEN ATTEMPTED

Fig. 1. Simplified diagram of the computer-aided design process.

In many problems the effect of parameter values on performance is complicated and the designer lacks intuition in determining parameter changes to improve performance. In such cases exploratory changes of parameters can be made to determine appropriate changes. This is feasible if each analysis is done with the aid of the computer. It is not difficult, however, to run into cases where the process becomes very laborious if several parameters are varied. Furthermore each analysis requires the user to look at printouts or plots, an operation which takes time and effort, especially if the design is done using the batch processing method. (This item is one of the strong motivations for the introduction of time-sharing in computer aided design.)

By defining an appropriate numerical criterion of performance and a search strategy for the parameter value changes, the innermost loop of Fig. 1 may be completely automated. Both operations are simple to program, but the eventual overall success of the method depends strongly upon how it is done. This is due to the fact that although qualitatively almost any search strategy applied to any initial set of parameters will converge to some local minimum error between the desired response and the actual response, quantitatively (in computer time) different search techniques will converge at very different rates. This point will be discussed at greater length in Part IV.

We now proceed to concentrate on the analysis box of Fig. 1 insofar as microwave circuits are concerned.

III. Computer Analysis of Microwave Circuits [16]

The main trouble with the circuit analysis programs available is that they suffer from one of the following defects:

1) They do not handle distributed circuits.
2) They do not accept circuits characterized empirically.
3) They do not handle general configurations.
4) They do not calculate quantities of interest to microwave engineers.

A circuit analysis program, BELNAP [16], designed specifically to avoid the problems mentioned above was implemented and is used at Bell Telephone Laboratories by many circuit designers working with microwaves and integrated circuits. The operation of BELNAP is very simple. It is based on well-known tools of multiport circuit analysis. With relatively little effort any nodal analysis program can be modified to avoid the problems mentioned above. A short review of the basic ideas is now given. Additional details may be obtained from [16].

All elements in a circuit are considered to be multiterminal black-boxes by BELNAP. Each one of these black-boxes is characterized by an indefinite admittance matrix (IAM), which is generally frequency dependent. The frequency dependency of the black boxes may be given to the computer in several ways, namely by:

1) analytical expressions,
2) tables of discrete values,
3) parameters in interpolating or approximating expressions.

For example, a capacitor (which may represent a fringing electric field) connected between nodes 2 and 4 in an n node circuit has an IAM which is an n×n matrix all of whose entries are zero except

$$Y_{22} = Y_{44} = Cp, \quad Y_{24} = Y_{42} = -Cp \qquad (1)$$

In this case analytical expressions give the frequency dependency of the IAM of the capacitor. Another case arising in microwave circuits of an IAM given by analytical expressions is the IAM of a pair of equal coupled lines of parameters per unit length L_{11}, L_{12}, C, C_m and length l connected to terminals 1, 2, 3, 4 as shown in Fig. 2. If this pair of lines is imbedded in a circuit with n terminals, the corresponding IAM is an n×n matrix all of whose entries are zero except the following:

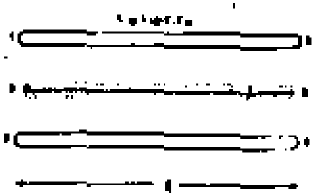


Fig. 2. Pair of equal coupled lines of length l over a ground plane. Quantities L_{11}, L_{12}, C, C_m are per unit length.

$$\begin{aligned}
&Y_{11} = Y_{22} = Y_{33} = Y_{44} = \tfrac{1}{2}A \coth (Ul) + \tfrac{1}{2}B \coth (Vl)\\
&Y_{12} = Y_{21} = Y_{34} = Y_{43} = \tfrac{1}{2}A \coth (Ul) - \tfrac{1}{2}B \coth (Vl)\\
&Y_{13} = Y_{31} = Y_{24} = Y_{42} = -\tfrac{1}{2}A \operatorname{csch} (Ul) - \tfrac{1}{2}B \operatorname{csch} (Vl)\\
&Y_{14} = Y_{41} = Y_{23} = Y_{32} = -\tfrac{1}{2}A \operatorname{csch} (Ul) + \tfrac{1}{2}B \operatorname{csch} (Vl) \qquad (2)\\
&Y_{15} = Y_{51} = Y_{25} = Y_{52} = Y_{35} = Y_{53} = Y_{45} = Y_{54}\\
&\qquad = A[\operatorname{csch} (Ul) - \coth (Ul)]\\
&Y_{55} = 4A[\coth (Ul) - \operatorname{csch} (Ul)]
\end{aligned}$$

where

$$A = \sqrt{\frac{C}{L_{11} + L_{12}}}, \qquad B = \sqrt{\frac{C + 2C_m}{L_{11} - L_{12}}},$$

$$U = j\omega\sqrt{(L_{11} + L_{12})C}, \qquad V = j\omega\sqrt{(L_{11} - L_{12})(C + 2C_m)}.$$

Similar expressions may be obtained for single transmission lines, RC lines and certain waveguides, etc., and incorporated into an analysis program.

Many microwave elements are difficult to model analytically because of the difficulty in solving the corresponding fields in closed form. This difficulty occurs either because not enough is known about the device or because of imperfections in manufacturing. Most of the active devices and those with fringing fields (bends, discontinuities, cavities, etc.) fall into this category. These may be characterized by numerical calculations of the fields or by external measurements of their scatterings at discrete frequencies. By mathematical transformations the IAM matrix can be obtained numerically at discrete frequencies. For example, if the scattering matrix of a multiport is measured at a given frequency, with all ports having a common terminal, the (definite) nodal admittance matrix Y is obtained by the equation [23]

$$Y = R_L^{-1/2}(I + S)^{-1}(I - S)R_L^{-1/2} \qquad (3)$$

where I is the unit matrix and R_L is a diagonal matrix whose entries are the terminating loads at the ports during the measurements. The IAM is then obtained from Y by adding a row and column equal to the negative of the sums of the rows and columns of Y. This is done at each measured frequency.

Intermediate frequency values may be obtained by any

od of approximation or interpolation [24]. Linear interpolation has been found to give adequate results for most . It is now the standard method used by MCNAP with polynomials in frequency, polynomials in the log of frequency and log-log graph polynomials available on request. To avoid accuracy problems, orthogonal polynomials are used. The orthogonal polynomials used are special sets which have been orthonormalized over an adjustable frequency interval in multiple precision by the Gram-Schmidt method [24]. The different sets of polynomials were chosen so that some curve shapes which appear often are well approximated with three or four terms. More sophisticated methods such as splines [25] could be used with more aesthetic results but with correspondingly more complicated computations. When a table has been approximated by polynomials only the coefficients need to be stored. Once each subcircuit (two-terminal elements being particular cases) of a circuit is characterized through its IAM as a black-box according to any of the three methods mentioned, the IAM of the complete circuit is found by simply adding the corresponding IAMs.

At any point in the analysis internal terminals may be eliminated and the IAM at specified terminals of the complete circuit or a subcircuit found numerically at discrete frequencies. The mathematics of this operation are quite simple. Let I_1 and I_2 be vectors of currents entering the p external terminals and q internal terminals, V_1 and V_2 the corresponding vectors of voltages (with a floating node as reference), and Y_{11}, Y_{12}, Y_{21}, Y_{22} submatrices of the IAM of proper dimensions for the equation

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \tag{4}$$

to be properly partitioned. Elimination of V_2 yields

$$I_{eq} = Y_{eq} V_1 \tag{5}$$

where

$$I_{eq} = I_1 - Y_{12} Y_{22}^{-1} I_2 \qquad Y_{eq} = Y_{11} - Y_{12} Y_{22}^{-1} Y_{21}. \tag{6}$$

I_{eq} is a vector of equivalent independent sources connected to the external terminals to account for the internal sources. Y_{eq} is the IAM of the network with the internal sources suppressed. If no internal sources exist then $I_2 = 0$ and $I_{eq} = I_1$.

IV. Automatic Parameter Optimization

To automate the inner loop of Fig. 1 once an analysis program is available, it is necessary to

1) define a desired response in numerical terms,
2) define a criterion of performance or "distance" between the desired response and actual response,
3) define an optimization or minimization strategy.

The definition of a desired response normally means that the user specifies quantities like return loss, insertion gain or voltage standing wave ratio which he desires the circuit to achieve at several discrete frequencies. The specified quantities may be thought of as a point in N-dimensional space. (For example if the return loss magnitude at four frequencies is specified and the value of the return loss magnitude at each frequency is measured on an axis, a circuit response is a 4-dimensional point. The "ideal circuit" response is also a 4-dimensional point in the same space.) A distance is then defined in the space. One of the most popular distances is the so called "Euclidean distance" defined as the square root of the sums of the squares of the differences between the coordinates of the ideal circuit response and the actual circuit response. Other definitions are possible, for example instead of squares, absolute values may be used. The Chebyshev norm (maximum deviation) and pth norms have been found to produce terrain which is more rugged and hence more difficult to explore than the Euclidean norm. They are also more difficult to compute. For many cases it is convenient to "weigh" some frequencies more than others. MCNAP uses a double weighted Euclidean distance as will be explained below.

It is not necessary that the point defined by the ideal circuit response correspond to only one quantity or only one port. The ideal circuit response may involve quite complicated conditions, for example, return loss at ports one and four and insertion loss at ports one and two, one and three and one and four; all at five discrete frequencies. It may be more important to achieve the right return loss than the insertion losses, hence one may weigh return losses with five and insertion losses with unity. Furthermore, one may weight the center frequency more than the frequencies at the edges of the band.

Finally, it is not necessary that the ideal circuit response be defined as a point. In some cases one would like to define it as a region. For example for a notch filter one may be satisfied to have at least 30 dB of loss at a given frequency rather than exactly 30 dB. Since such types of specifications are easily implemented with logical decision statements on the computer, they should be considered in designing a program.

The definition of desired response and criterion of performance are very delicate matters because the final results will heavily depend on them. In order to come up with good definitions the designer is forced to ask himself "What do I really want the circuit to do?" This question is a difficult one. A designer would usually prefer to decide a posteriori which of several circuits he likes most rather than to have to state a priori the characteristics of the best one. Not only will the final results differ for different definitions of desired response and criterion of performance, but also, the time for a search strategy to arrive at them. For example it is well-known by researchers in automatic optimization that a "Chebyshev distance" (the distance between two curves being given by their maximum deviation) is apt to present a much more rugged terrain (if an analogy between the function to be minimized and height in a surveyor's map is made) than Euclidean distance. The rugged terrain will force the search strategy to use large quantities of computer time. Another typical example is when a naive designer specifies a bandpass filter with sharp corners and vertical walls. Knowing that he cannot expect sharp corners he should round them in a reasonable manner, otherwise the computer will

spend all its time trying to square the corners and in trying to achieve the vertical walls at the expense of producing undesirable bulges in the band of interest. The end result is not at all what the designer expected. Usually it gives a horrible filter by anyone's standards. This could be corrected by proper frequency weighting. The only way to pick up all these pieces of information is by experience with different types of circuits and different definitions of desired response and criterion of performance. This is an area where interactive computing has a great application.

In implementing the search strategy one has quite a bit of choice [7], from grid search to gradient techniques of all kinds to random search. Each strategy has its own advantages and disadvantages and its own requirements. For example, gradient techniques require partial derivative information which is very hard to come by for complicated microwave circuits composed of combinations of analytically and experimentally characterized subnetworks. One possible way of obtaining partial derivatives is to find them numerically by evaluating the response with each parameter incremented a small amount. This is usually expensive computationally. Because of the problems in obtaining partial derivatives for microwave networks plus the fact that the terrains are usually rugged enough not to be worth the sophistication of "differentially" following the surfaces, it was decided to use a direct pattern search technique for MCNAP. The one finally implemented was proposed by Hooke and Jeeves [26]. The direct search method has a historic record of successfully climbing in very rugged terrain (which is usually the case in circuit problems) and growing linearly with the number of parameters. It is also quite easy to handle constraints on the values of the parameters. These are necessary to end with meaningful engineering results. (For example, it would be very undesirable to end with a negative characteristic impedance for a transmission line.)

The direct pattern search strategy has two major components: the exploratory move and the pattern move. Briefly, the strategy is described as follows. An initial guess is made of the n parameters for a problem and some error criterion is evaluated at that set of parameter values with the aid of an analysis program. Starting with the initial set of parameters, an exploratory move is made. This exploratory move varies the value of each parameter by some small amount and observes the effect of each of these variations on the error expression. Those changes that reduce the error are retained. It may be that certain of the parameters are slightly increased, others decreased, and still others unchanged to reduce the error. This new set of slightly modified parameters determines the "unit vector" in n-dimensional parameter space of a move which will reduce the error. The next step is to make a larger move in the parameter values in the direction indicated by the exploratory move. This larger move is called a pattern move. After the pattern move has been made, the error expression is again evaluated for the new parameter values. If the pattern move succeeded in reducing the error, another pattern move is made in the same direction as the first one. The pattern moves continue until one fails to reduce the error. At this point, a new exploratory move is made to determine a new direction for a pattern move. This entire process continues until an exploratory move is unable to reduce the error expression further. At this point, the program terminates. The flow chart of Fig. 3 outlines the pattern search strategy. The details of the exploratory move are indicated in Fig. 4. The steps are repeated for each parameter of the system.

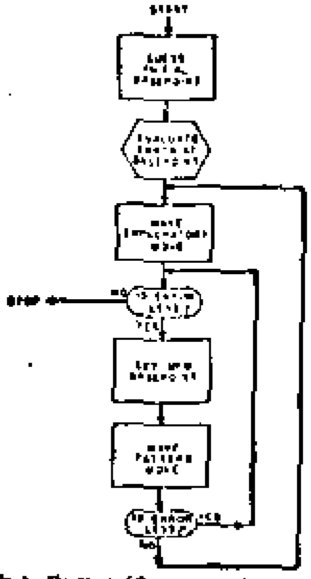


Fig. 3. Flowchart of the pattern search strategy.

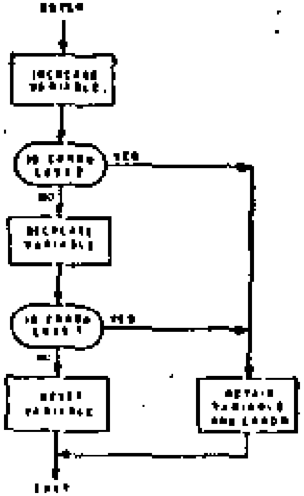


Fig. 4. Flowchart of the exploratory search strategy.

V. Improving the Computational Efficiency Through the IAM Black-Box Approach

A feature deemed critical in coupling an analysis program to an optimization routine is the numerical efficiency of each analysis. In a typical optimization run, the analysis program may be called several hundred times; therefore any saving in the computational analysis effort is multiplied by several hundred.

The indefinite admittance matrix formulation employed by IMPROVE allows a significant decrease in computational effort. In the optimization some subcircuits have no adjustable parameters. Since this fixed part of the network does not change with different parameter set trials, it should not be necessary to solve the equations of the invariant portion for every new parameter set. To avoid the repetitious analysis of the fixed part, the network is partitioned into its variant and invariant portion. The fixed portion of the network is characterized at its essential terminals and stored as a black-box. Essential terminals are those which are specified as external ports or those which have adjustable elements connected (see Fig. 5). To save computer core storage the subcircuits can be reduced to the essential terminals in preliminary passes. This can easily be done because BELNAP can punch on request the IAM of a circuit reduced to a given set of terminals. The punched data is in a format in which it can be read for a subsequent analysis or optimization. In this way very large circuits containing few variable elements can be handled in several passes.

In order to characterize the fixed portion of the network at its essential terminals, the nonessential terminals are eliminated by using (5) and (6). H. C. So [27] uses this method with a hybrid formulation for lumped circuits and reports considerable computer-time savings when compared with efficient programs not using the partitioning technique. It has not been possible to compare IMPROVE in microwave problems with other analysis programs because of the unavailability of other general configuration programs capable of handling distributed multiport networks. However, we expect that the relative savings from the IAM black-box approach would be even greater than those reported by So, because he has to consider the time necessary for the computation of the hybrid matrix at each frequency. In our approach, computation of an IAM involves only adding the IAMs of the subcircuits. Additionally, the algorithms for hybrid analysis are considerably more complicated than those for nodal analysis so the programming effort for implementing our method is considerably smaller.

The principal reason given by So [27] for using general hybrid analysis is that, since the adjustable parameters may be distributed arbitrarily in the network, the open-circuit impedance matrix or short-circuit admittance matrix of the resulting n-port frequently does not exist. This is particularly true of open-circuit impedance matrices. In practical circuits the authors have yet to find a single case in which the IAM of any subcircuit embedded in a real network does not exist. Granted that networks such as ideal transformers and perfectly coupled mutual inductances do not have admittance or impedance matrices. These devices are, however,

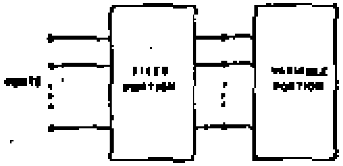


Fig. 5. Network partitioned into its invariant and variable parts for analysis by IMPROVE.

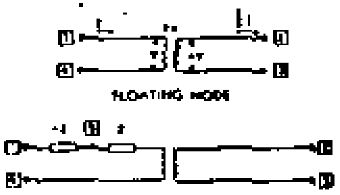


Fig. 6. Ideal transformer has no indefinite admittance matrix. Addition of virtual node 5 removes degeneracy.

theoretical idealizations which never appear in microwave networks. On the other hand, even from a purely theoretical point of view, as long as the complete circuit possesses an IAM, the problem can easily be circumvented as follows: [16] for every degenerate device (such as an ideal transformer) insert a virtual node (a node which does not exist physically) to which positive and negative impedances of equal values are connected. These cancel each other's effects and therefore leave the circuit undisturbed. This is shown in Fig. 6 for an ideal transformer. The additional virtual node destroys the degeneracy because the circuit with an additional node does have an IAM. The virtual node is eliminated after the additional subcircuits are connected. This occurs when the total IAM at the real terminals is no longer degenerate. A computer program can accept positive or negative parameters with equal ease. Since the value of the canceling artificial impedances can be chosen by the designer, they may be chosen so that they are comparable to other impedances in the network. That is, if the other impedances are of the order of 10 ohms the artificial impedances should also be of the same order. This simple artifice also has application in other instances, such as obtaining Norton equivalents of voltage sources without a series impedance and avoiding ill-conditioned equations when impedances of very different magnitudes are connected to the same node [28].

VI. Brief Description of IMPROVE (Example)

The method of analysis used by IMPROVE is identical to that of the program BELNAP [16]. The circuits handled may contain:

1) Passive R's, L's, and C's.
2) Uniformly distributed transmission lines through the specification of R, L, G, and C per unit length and the

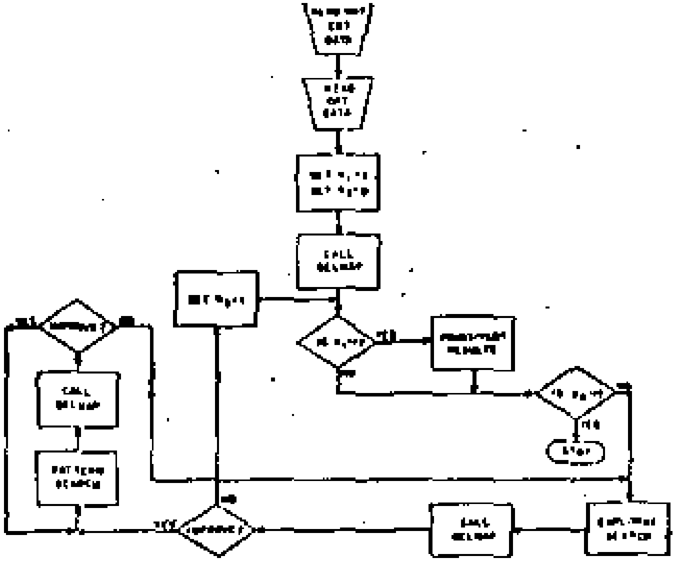


Fig. 7. Simplified flowchart of analysis-optimization program IMPROVE.

length or characteristic impedance and electrical length at a given reference frequency.

3) Active devices for which conventional modeling with controlled sources and passive elements is adequate.
4) "Black-boxes" for which conventional modeling is difficult or impossible but whose terminal characteristics (S, Y, Z, H, or $ABCD$ parameters) can be obtained. (For example, one can make terminal measurements at several frequencies on a microwave transistor, a directional coupler or similar black-box and input this data without first modeling the device. Intermediate frequency values are interpolated by the program.) Theoretically modeled devices may be defined using external subroutines.

Additionally for optimization purposes IMPROVE accepts the following specifications from the user:

1) Variable parameters (including minimum and maximum values for all variable parameters). Parameters which vary together.
2) Frequencies and weighting factors for the frequencies.
3) Desired performance characteristics
   a) Impedance (magnitude and/or angle or real and/or imaginary parts).
   b) Return loss.
   c) Insertion gain (magnitude in dB or phase).
   d) Voltage gain (real, imaginary, magnitude and/or angle).
   Any of the previous up to 4 ports.
4) Weights for the performance characteristics. (For example, a user may weigh impedances 10 and insertion gains 1.)

The parameters which may be declared variable in IMPROVE are resistors, capacitors, inductors, controlled sources, per-unit-length parameters of transmission lines (R, L, C, G), lengths of lines, characteristic impedances of lines, and electrical lengths of lines. Because some parameters may vary together it is possible to approximate some distributed elements with lumped circuits and thus determine their per-unit-length parameters approximately. Also it is possible to force some circuits to retain certain desirable physical symmetries.

A simplified flow chart of the operation of IMPROVE is shown in Fig. 7.

The program outputs the value of the error between the desired response and the response of the first and final trial circuits, a list of the final circuit parameters and a complete analysis (including plots) of the final circuit.

Two examples are now presented to illustrate the use of IMPROVE and BELNAP. One of the biggest problems facing microwave engineers is that of unwanted parasitics and their modeling to predict correct circuit behavior. This modeling can be formulated as an optimization or design problem. When the modeling of parasitics is solved, the parasitics can be used in favor of the designer to fine tune circuits. This is especially important when many circuits will be mass produced.

Consider now the 20-dB switched microwave attenuator of Fig. 9. We first wish to characterize each of the elements

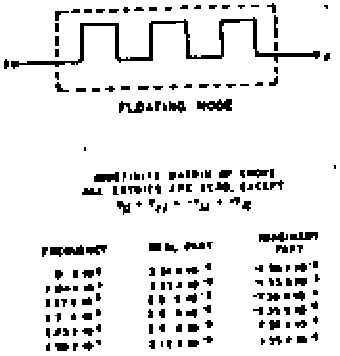

FLOATING NODE
ADMITTANCE MATRIX OF CHOKE
ALL ENTRIES ARE ZERO, EXCEPT
$Y_{11} = Y_{22} = -Y_{12} = -Y_{21}$
FREQUENCY
REAL PART
IMAGINARY PART

Fig. 8. Choke characterized as black box for analysis with BELNAP.

within the circuit and use the BELNAP program to analyze the performance of the ideally (no parasitics) interconnected network.

The diodes were modeled by equivalent circuits as shown in Fig. 9. The values of the elements in the model are conveniently determined by using the optimization program IMPROVE to match measured data on the diodes. This approach was used here to illustrate equivalent circuit modeling. Alternatively, measured data on the device may be directly fed into the program as in the case of the quarter wavelength chokes of the circuit which are shown in Fig. 8 as two-terminal black boxes. Here, measured data on the Y parameters of the chokes are used as a description to the program. As shown in the input description of Table I, the choke is given as type number 10, and the Y parameters given at 6 frequencies (0.9 GHz through 1.36 GHz). Also shown in Table I is the input of R's, L's and C's described by their connection nodes and element values.[4] The three transmission lines are incorporated into the circuit by giving their connection nodes, R, L, G, C per unit length, and length of the line.

The configuration of Fig. 9 was built and its performance measured. The measured values of return loss and insertion gain are shown in Table II. Observe that they differ noticeably from the response predicted by the computer (shown under initial value). The discrepancy can be attributed to parasitic effects in the final circuit layout. These parasitics include fringing capacitance at the termination of each of the transmission lines (including input and output), lead inductance associated with each of the diodes (except diode 1 which required virtually no lead length in the final layout) and inductance associated with each of the resistors.

It was desirable to determine the value of the parasitic elements which were accounting for the discrepancy between predicted and measured response. To do this, the measured data was used as direct input to the optimization program.

TABLE I

CARD INPUT TO BELNAP OF THE CIRCUIT OF FIG. 12

```
TWENTY DB SWITCHED MICROWAVE ATTENUATOR
 $INPUT NODES    20, PORTS 2, FREQI 1 E9, FREQF 1.5E9,
   FROMC .05E9, PLOT 1 $
 $BRANCH NODES 3,0,C .3E-12 $
 $BRANCH NODES 4,0,C .3E-12 $
 $BRANCH NODES 3,4,L .3E-9 $
 $BRANCH NODES 4,0,R 5 $
 $BRANCH NODES 1,4,C .3E-12 $
 $BRANCH NODES 1,18,C .2E-12 $
 $BRANCH NODES 6,18,L .3E-9 $
 $BRANCH NODES 1,18,R 29.0 $
 $BRANCH NODES 7,9,C .3E-12 $
 $BRANCH NODES 8,9,C .2E-12 $
 $BRANCH NODES 7,8,L .3E-9 $
 $BRANCH NODES 8,9,R 5 $
 $BRANCH NODES 7,11,C .3E-12 $
 $BRANCH NODES 13,14,C .3E-12 $
 $BRANCH NODES 3,14,L .3E-9 $
 $BRANCH NODES 13,14,R 5 $
 $BRANCH NODES 2,15,C .3E-12 $
 $BRANCH NODES 2,17,C .3E-12 $
 $BRANCH NODES 16,17,L .3E-9 $
 $BRANCH NODES 2,17,R 290. $
 $BRANCH NODES 17,0,R 41 $
 $BRANCH NODES 21,0,R 41 $
 $BRANCH NODES 9,11,R 247.3 $
 $BRANCH NODES 1,0,C 1 E-20 $
 $BRANCH NODES 2,0,C 1 E-20 $
 $BRANCH NODES 5,0,C 1 E-20 $
 $BRANCH NODES 15,0,C 1 E-20 $
 $BRANCH NODES 8,0,C 1 E-20 $
 $BRANCH NODES 10,0,C 1 E-20 $
 $BRANCH NODES 11,0,C 1 E-20 $
 $BRANCH NODES 12,0,C 1 E-20 $
 $BRANCH NODES 4,1,L 1 E-12 $
 $BRANCH NODES 1,7,L 1 E-12 $
 $BRANCH NODES 12,11,L 1 E-12 $
 $BRANCH NODES 15,16,L 1 E-12 $
 $BRANCH NODES 10,19,L 1 E-12 $
 $BRANCH NODES 11,20,L 1 E-12 $
 $BRANCH NODES 3,10,L 1.E-12 $
 $BRANCH $
 $MATRIX NODES 9,10,0,L .2163E-6,C .8670E-12,X 3.23 $
 $MATRIX NODES 11,12,0,L .2163E-6,C .8670E-12,X 3.23 $
 $MATRIX NODES 5,15,0,L .2163E-6,C .8670E-12,X 4.44 $
 $MATRIX NODES 1,0,TYPE 10 $
7
6
.9 3.1E 6 -1.9RE-3
1.047 3.1E 6 -1.15E-3
1.17E-1 2.9E-6 -2.5E-4
1.19 2.6E 6 -1.33E-8
1.4E 1 2.9E 6 7.33E-4
1.5E 4 1.E2. 6 1.33E-3
  $MATRIX NODES 2,0,TYPE 10 $
  $MATRIX NODES 13,0,TYPE 10 $
  $MATRIX $
  $LOADS NODES 1,0,R 50. $
  $LOADS NODES 2,0,R 50. $
  $LOADS $
```

Note: an equivalent circuit is shown in Fig. 11 with no parasitics. The parasitic values entered are negligible.

Return loss at ports 1 and 2, and insertion gain were specified at 11 frequencies between 1 GHz and 1.5 GHz. The circuit of Fig. 10 which includes the parasitic elements was described. (The data shown in Table I was fed to IMPROVE and the parasitic elements were declared variable.) Values of C_1, L_2, L_3, and L_4 were sought to meet the measured data. Sixty-seven exploratory moves and 187 pattern moves were

[4] Note in Table I the presence of several capacitors with values 10^{-20} and inductors of 10^{-12}. These were introduced only to avoid having to renumber the circuit for the optimization run. The inputs of BELNAP and IMPROVE are compatible.

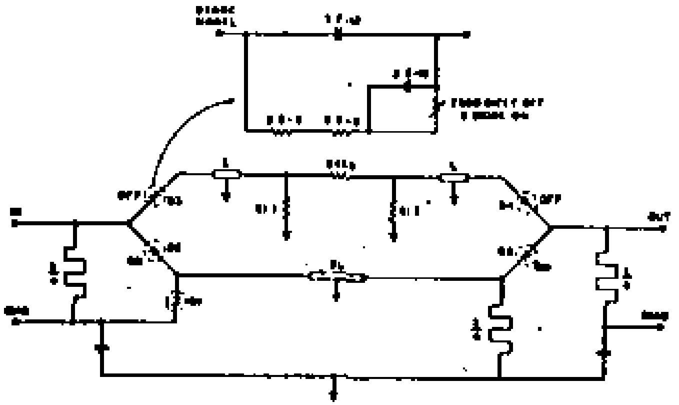


Fig. 9. 20 dB switched microwave attenuator to be analyzed by BELNAP.

TABLE II

VALUES FOR 11 FREQUENCIES OF RETURN LOSS AT PORTS 1 AND 2 AND INSERTION GAIN FROM PORT 1 TO PORT 2 OF THE CIRCUIT OF FIG. 12, MEASURED AND COMPUTED WITHOUT PARASITICS WITH BELNAP, COMPUTED WITH PARASITICS DETERMINED BY IMPROVE

Frequency ($\times 10^9$ Hz)	Desired Value (measured)	Initial Value (no parasitics)	Final Value (from IMPROVE)
Return loss at 1, dB			
1.00	29.0	34.77	29.10
1.05	27.0	31.92	27.11
1.10	25.1	29.21	31.73
1.15	21.1	27.16	23.30
1.20	22.1	24.65	23.13
1.25	21.3	23.28	21.01
1.30	20.0	22.11	20.04
1.35	19.0	21.05	19.10
1.40	18.0	20.10	18.26
1.45	17.3	19.24	17.52
1.50	16.1	18.44	16.80
Return loss at 2, dB			
1.00	20.0	23.82	19.51
1.05	13.0	20.18	16.17
1.10	13.3	15.09	11.49
1.15	13.0	9.134	11.44
1.20	20.3	17.13	19.15
1.25	11.3	28.80	37.24
1.30	21.3	45.41	24.90
1.35	21.3	35.13	21.44
1.40	21.8	33.40	20.01
1.45	19.3	24.68	19.80
1.50	18.3	22.67	17.94
Insertion Gain (1 to 2), dB			
1.00	−20.5	−20.33	−20.17
1.05	−20.5	−20.39	−20.49
1.10	−20.4	−20.74	−20.50
1.15	−21.0	−24.80	−22.86
1.20	−20.8	−21.46	−20.91
1.25	−20.7	−20.16	−20.96
1.30	−20.4	−20.61	−20.79
1.35	−20.4	−20.56	−20.38
1.40	−20.4	−20.55	−20.99
1.45	−20.8	−20.33	−20.41
1.50	−20.4	−20.56	−20.61

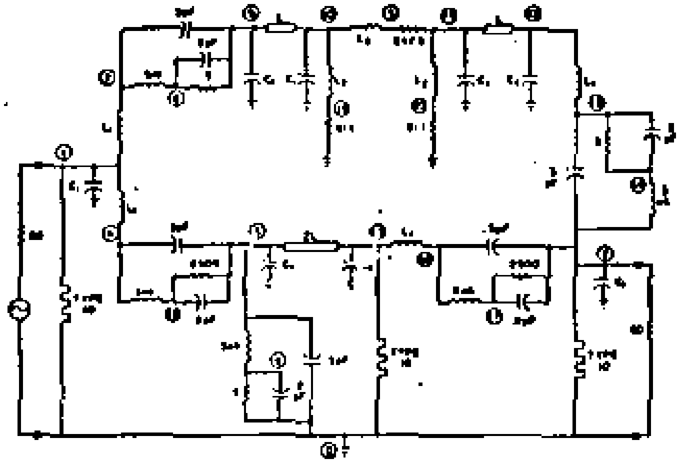


Fig. 10. Equivalent circuit of the microwave attenuator of Fig. 9 with unknown parasitics inserted.

TABLE III

VALUES OF PARASITIC ELEMENTS CALCULATED BY IMPROVE FOR THE CIRCUIT OF FIG. 11

Nodes		Element
1	0	C=1.1700E-13
2	0	C=1.1700E-13
3	0	C=1.1700E-13
13	0	C=1.1700E-13
9	0	C=1.1744E-13
10	0	C=1.1744E-13
11	0	C=1.1744E-13
12	0	C=1.1799E-13
4	1	L=3.4170E-10
1	7	L=3.4112E-10
15	16	L=3.4532E-10
10	19	L=3.1694E-10
11	20	L=3.1694E-10
3	10	L=3.1735E-09

required to converge. Seven minutes of GE635 CPU time were required. The final element values are shown in Table III. Some of the elements were constrained to be equal within the program, since they were approximately equal in the circuit layout. The response of the optimized network is compared to the response of the initial network without parasitics and the measured data. Table II shows the values of input and output return loss and insertion gain both before and after optimization. Figs. 11, 12, and 13 show the error versus frequency before and after the optimization. Good agreement is noted, indicating that the optimization has found element values that closely approximate the effects of the parasitics. A sample of printout of the analysis of the final circuit at one of the frequencies is shown in Table IV.

The above example was intended to illustrate the flexibility in matching and topology available to the microwave engineer. A second example will be used to illustrate the design capability that such an optimization program offers. An important problem faced by microwave engineers is that of matching two arbitrary impedances. In the usual case, one impedance is the characteristic impedance of a transmission line and the second impedance is specified at a set of frequency points. This is the situation, for example, in conjugate matching of the input and output impedances of a transistor [32], [33]. One approach that has been employed with good results is the successive manual application of an analysis program to optimize the matching network [33]. The parameter values are varied and from the many responses that are recorded, the designer chooses the optimum. The following example will carry this procedure one step further by automatically varying the parameters until an optimum solution is achieved.

Consider the double stub matching network of Fig. 14, which will be used to match a 50 ohm line to the input impedance of a transistor specified at a set of data points. The network, as viewed from port 2 was described to the program, along with the requirements. The real part of Z_{in} was set to the real part of the transistor input impedance, and the imaginary part set to the negative of the imaginary part of the transistor impedance. The center frequency of operation was 1.6 GHz and the matching was done over a 400 MHz bandwidth. The parameters declared variable were the five line impedances and line lengths. The initial and final values of these parameters are shown in Table V. Six minutes of GE635 CPU time were required. The initial and final errors in the desired response are shown in Figs. 15 and 16.

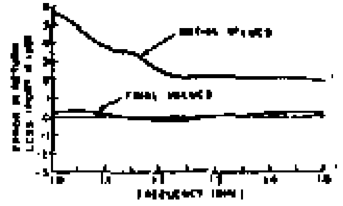


Fig. 11. Initial and final errors in return loss (port 1).

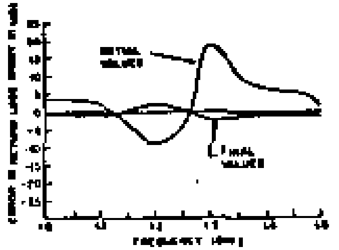


Fig. 12. Initial and final errors in return loss (port 2).

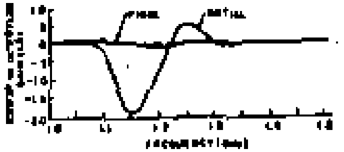


Fig. 13. Initial and final errors in insertion gain.

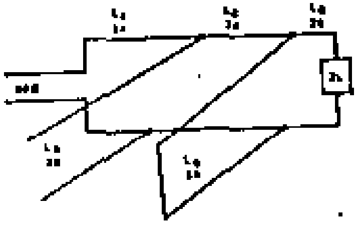


Fig. 14. Matching network.

TABLE IV

OUTPUT OF IMPROVE FOR 1.35 GHz COMPUTATIONS ON FINAL CIRCUIT OF FIG. 11 WITH PARASITICS DETERMINED BY AUTOMATIC OPTIMIZATION

```
Frequency = 0.1350E 10

Admittance Matrix of Network
1.9874E-02     3.6216E-04
3.7010E-04     3.6714E-03

5.4314E-04     1.9400E-02
5.6114E-03     1.1105E-01

Admittance Matrix of Loads
2.0000E-02     0
0              0

0              2.0000E-02
0              0

Driving Point Impedance
   Port        Real           Imaginary       Magnitude       Angle
    1          0.4799E 02    -0.1121 01      0.4814E 02     -0.1007E 02
    2          0.4909E 02    -0.2411 01      0.4993E 02     -0.2749E 01

Insertion Gain
   Ports       dB             Angle
   1 to 2     -0.2104E 02    -0.1054E 03
   2 to 1     -0.2004E 02    -0.1054E 03

Voltage Ratios
               dB             Angle           Real            Imaginary
E2/E1         -0.2050E 02    -0.1021E 03     -0.1614E 01     -0.9574E-01
E1/E2         -0.2061E 02    -0.1040E 03     -0.2245E 00     -0.8999E-01

Return Loss
   Port        dB
    1          0.2101E 01
    2          0.3234E 00
```

TABLE V

PARAMETER VALUES OF MATCHING NETWORK

	Initial	Final
Z_A	50	45
Z_B	50	79
Z_C	50	74
Z_D	50	24
Z_E	50	94
L_A	0.1λ	0.043λ
L_B	0.1λ	0.064λ
L_C	0.1λ	0.074λ
L_D	0.1λ	0.19λ
L_E	0.1λ	0.049λ

λ = wavelength at 1.6 GHz.

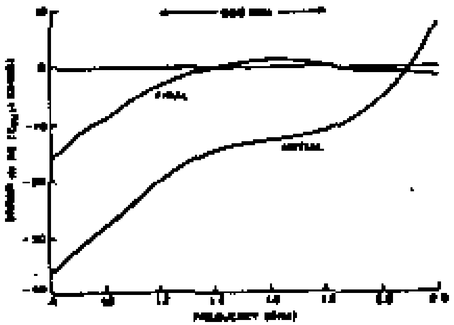


Fig. 15. Initial and final errors in Re (Z_{in}).

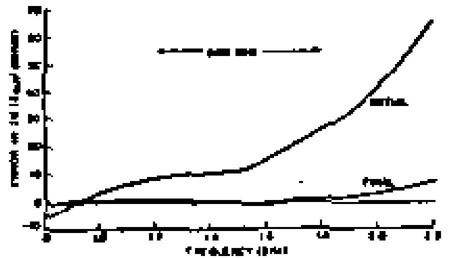


Fig. 16. Initial and final error in Im (Z_{in}).

VII. Conclusions

In this paper we have presented computer methods of analysis and design of linear networks which are suitable for microwave circuits. The methods have been implemented in two computer programs, an analysis program, BELNAP, reported on previously [16], and an analysis-optimization program, BELNOVE.

Lest the reader be misled to conclude that automatic parameter optimization is a panacea for circuit design, a few tempering comments should be made.

The success of automatic parameter optimization depends heavily upon the ability of the design engineer to avoid the pitfalls associated with it. The most serious are:

1) there is no guarantee that the method will converge to a global minimum; all it will do is go to a nearby local minimum;
2) for some functions the convergence may be painfully slow.

Because of these two stumbling blocks a designer must use automatic optimization with great care. For example, he should come as close to a solution as possible before he applies automatic optimization. For this purpose he will use all sorts of simplifying assumptions to be able to do some analytical "ball park" calculations, use approximate synthesis methods, etc. If necessary, he will do some "by hand" searches on the computer to make sure he starts with a circuit which comes close to the specifications. Some preliminary calculations may save him considerable computer time (we mean times of the order of 30 minutes CPU on large third generation computers).

We could give the readers the following advice.

1) Do not make too many parameters free at the same time. If you have relatively good estimates of some parameters, fix them, determine the ones you do not know, and then free the others. This requires more than one pass of the optimization but saves much computer time.
2) Do not use too many frequency points until you are very close to the final circuit. That is, do not use 20 frequency points when the circuit is not even in the ball park; remember every additional frequency is more computation. Start with three frequencies and increase their number when the response is close to the requirement.
3) Do not ask for unreasonable or contradictory things. Know the limitations of your circuits. Do not expect the phase to be going up when the magnitude is coming down or expect to get gain from passive elements. Do not expect to get square corners (they usually imply circuits which predict the future). Do not ask for power gains better than optimum.
4) If you are inexperienced, do not set time limits on your runs too high.

In short, do not approach automatic computer optimization naively.

In spite of the previous comments, when used properly, automatic parameter optimization is a very useful tool in the design of matching networks and directional couplers (where only approximate synthesis techniques exist). It is particularly useful in the design of nonideal networks containing active devices where parasitic and other effects have to be included and where synthesis techniques do not exist.

Considerable room for research remains in improving automatic optimization methods for microwave circuit design. Because of the pitfalls indicated above, it is very convenient to have the designer monitor the optimization as much as possible. This can be done if one uses time-shared operation or a dedicated machine. The matter of man-machine communication should receive careful consideration. In this respect the programs presented in this paper are being improved to simplify the input language which at present is somewhat complicated. It is contemplated that certain terms which network theorists prefer will be replaced by engineering quantities. (One example is the specification of lines which in microwave engineering are generally specified in terms of electrical lengths and characteristic impedance at a reference frequency, rather than R, L, G, C per unit length.) The organization of the files and programs should be specifically planned for interative operation [29]. With respect to this item a general input-output language is being developed so that a group of programs including linear and nonlinear transient, worst case, statistical variability and layout programs, can access stored descriptions of circuits without having to translate the different coding schemes from program to program. Graphical aids should be provided to avoid forcing the user to go through long lists of printout.

On the numerical side, a breakthrough in efficiency is needed because most problems are too large computationally to handle on a time-shared basis. Such a breakthrough has already been made for dc circuits and single frequency ac circuits by using singular imbedding [30]. Encouraging preliminary results have been obtained for multifrequency ac and transient design. Undoubtedly the area still has room for vigorous growth. We hope this paper encourages other researchers to take the challenge.

Acknowledgment

The authors would like to acknowledge the contributions of M. P. Greer and L. A. Davieau, the writers of BELNOVE and BELNAP.

References

[1] 1620 Electronic Circuit Analysis Program (ECAP) User's Manual, IBM Corporation, White Plains, N. Y., 1965.
[2] S. R. Sedore, "SCEPTRE: A program for automatic network analysis," IBM Journal, vol. 11, no. 6, pp. 627–637, November 1967.
[3] L. D. Milliman, W. A. Massena, R. H. Dickhaut, and A. C. Hong, "CIRCUS: a digital computer program for transient analysis of electronic circuits (User's guide)," 2 vols., The Boeing Co., Washington, D. C., January 1967.
[4] L. P. McNamee and H. Potash, "A user's guide and programmer's manual for NASAP," University of California, Los Angeles, Dept. of Engineering, Rept. 68-38, August 1968.
[5] F. F. Kuo and J. F. Kaiser, Eds., *System Analysis by Digital Computer*. New York: Wiley, 1966.
[6] G. J. Herskowitz, Ed., *Computer-Aided Integrated Circuit Design*. New York: McGraw-Hill, 1968.
[7] P. E. Fleischer, "Optimization techniques in system design," in *System Analysis by Digital Computer*, F. F. Kuo and J. F. Kaiser, Eds. New York: Wiley, 1966.
[8] G. C. Temes and D. A. Calahan, "Computer-aided network optimization: the state-of-the-art," *Proc. IEEE*, vol. 55, pp. 1832–1863, November 1967.
[9] D. A. Calahan, "Computer solution of the network realization problem," *Proc. 2nd Allerton Conf. on Circuit and System Theory*, pp. 175–193.
[10] M. A. Murray-Lasso and W. D. Baker, "Computer design of multistage transistor bias circuits," *Proc. 1967 Allerton Conf. on Circuit and System Theory*, pp. 533–543.
[11] L. S. Lasdon and A. D. Waren, "Optimal design of filters with bounded, lossy elements," *IEEE Trans. Circuit Theory*, vol. CT-13, pp. 175–187, June 1966.
[12] F. G. Schultz and J. A. Huber, "The application of Carroll's optimization technique to network synthesis," *Proc. 1965 Allerton Conf. on Circuit and System Theory*, pp. 182–191.
[13] H. A. Spang, "A review of minimization techniques for nonlinear functions," *SIAM Review*, vol. 4, pp. 343–365, 1962.
[14] D. J. Wilde, *Optimum Seeking Methods*. Englewood Cliffs, N. J.: Prentice-Hall, 1964.
[15] S. H. Brooks, "A comparison of maximum seeking methods," *J. Operations Res. Soc.*, vol. 7, pp. 430–457, 1959.
[16] M. A. Murray-Lasso, "Analysis of linear integrated circuits by digital computer using black-box techniques," in *Computer-Aided Integrated Circuit Design*, G. J. Herskowitz, Ed. New York: McGraw-Hill, 1968.
[17] H. Wyndrum, Jr., "The exact synthesis of distributed RC networks," *IEEE Conv. Rec.*, pt. 7, pp. 91–93, 1965.
[18] M. Saito, "Synthesis of transmission line networks by multivariable techniques," *Proc. Symp. on Modern Network Synthesis* (Brooklyn Polytechnic Institute), vol. 16, pp. 353–392, 1966.
[19] D. C. Youla, "Synthesis of n-ports containing lumped and distributed elements," *Proc. Symp. on Modern Networks*, vol. 16, 1966.
[20] R. W. Newcomb, *Linear Multiport Synthesis*. New York: McGraw-Hill, 1966.
[21] D. Hazony, *Elements of Network Synthesis*. New York: Reinhold, 1963.
[22] H. Ozaki and J. Ishii, "Synthesis of transmission-line networks and the design of UHF filters," *IRE Trans. Circuit Theory*, vol. CT-2, pp. 325–336, December 1955.
[23] L. Huelsman, *Circuits, Matrices, and Linear Vector Spaces*. New York: McGraw-Hill, 1963.
[24] P. J. Davis, *Interpolation and Approximation*. Waltham, Mass.: Blaisdell, 1963.
[25] J. H. Ahlberg, E. N. Nilson, and J. L. Walsh, *The Theory of Splines and Their Applications*. New York: Academic Press, 1967.
[26] R. Hooke and T. A. Jeeves, "Direct search solution of numerical and statistical problems," *J. ACM*, vol. 8, pp. 212–229, April 1961.
[27] H. C. So, "Analysis and iterative design of networks using on-line simulation," in *System Analysis by Digital Computer*, F. F. Kuo and J. F. Kaiser, Eds. New York: Wiley, 1966.
[28] M. A. Murray-Lasso, "Black box models for linear integrated circuits," *IEEE Trans. Education*, Special Issue on Computer-Aided Design (to be published).
[29] D. T. Ross, "The AED approach to generalized computer-aided design," M.I.T. Electronic Syst. Lab., Cambridge, Mass., Rept. ESL-R-305, April 1967.
[30] E. B. Kozemchak and M. A. Murray-Lasso, "Computer-aided circuit design by singular imbedding," *Bell Sys. Tech. J.*, vol. 48, pp. 275–315, January 1969.
[31] R. M. Fano, "Theoretical limitations on the broad-band matching of arbitrary impedances," *J. Franklin Inst.*, vol. 249, pp. 57–83, 139–155.
[32] F. E. Emery, M. O'Hagan, and S. O. [illegible], "Optimal design of matching networks for microwave transistor amplifiers," *1966 G-MTT Internat'l Symp. Digest*, pp. 100–106.
[33] V. G. Gelnovatch and T. F. Burke, "Computer-aided design of wide-band integrated microwave transistor amplifiers on high dielectric substrates," *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-16, pp. 429–439, July 1968.

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO OPTIMO DE SISTEMAS DE INGENIERIA

PROBLEMAS ASOCIADOS A SISTEMAS ELECTRICOS

"DESPACHO ECONOMICO DE CARGA"

MARZO, 1982

INTRODUCCION

El objetivo primordial en la operación de un sistema eléctrico de potencia es el de proporcionar a los consumidores la energía eléctrica un servicio sin interrupciones a frecuencia constante y dentro de límites tolerables de voltaje en todos los puntos de suministro. Esto implica el controlar en tiempo real la generación para satisfacer en todo instante las continuas variaciones de la demanda. Esta acción, de igualar minuto a minuto la generación a la demanda, es un problema fundamental en todo sistema de conversión de energía y se torna particulamente complejo en una red eléctrica en donde se operan en forma simultánea un gran número de unidades generadoras de muy diversas características y entre las que se destacan por su importancia las siguientes :

a) Capacidad de generación

b) Tipo de combustible utilizado

c) Eficiencia

d) Costos de operación

e) Características de respuesta

El problema de despacho de carga consiste en decidir que porción de la demanda deberá ser suministrada por cada una de las unidades operando en el sistema y además de las características antes mencionadas deberá considerar los siguientes factores adicionales:

a) Efectos de la distribución de la generación sobre las pérdidas de transmisión.

b) Capacidad de transmisión de las líneas.

c) Políticas de localización de reserva rodante.

d) Acuerdos de compra-venta de energía con otros sistemas.

e) Políticas de operación de plantas hidroeléctricas.

Si al resolver el problema de despacho de carga se busca entre las múltiples soluciones factibles aquella que minimiza el costo de operación, al problema se denomina despacho económico. El propósito de este trabajo es el de plantear este último problema y presentar algunas de las técnicas utilizadas en su solución.

CONSIDERACIONES Y DEFINICIONES

Los avances logrados en años recientes en la tecnología de conversión de energía, caracterizados por incrementos en tamaño, presión y temperatura de las unidades generadoras de vapor, han dado como resultado una continua mejoría en la eficiencia de operación de las unidades. En consecuencia ha aumentado la disparidad en el consumo específico de las fuentes alternativas de generación, que podrían en un tiempo dado ser utilizadas para satisfacer una determinada demanda. Esta disparidad sumada a las variaciones en los costos de combustible y en los factores de pérdidas por transmisión, hacen necesaria la utilización de técnicas eficientes de despacho económico para lograr ahorros considerables de combustible al operar un sistema eléctrico de potencia.

Como un ejemplo de los avances logrados se puede mencionar[1] que en el año de 1940 el consumo específico para generación de energía eléctrica utilizando combustibles fósiles era en promedio de 4132.8 KCal/KW Hr (16,400 BTU/KW Hr). Para 1950 este promedio había mejorado a 3528 KCal/KW hr (14,000 BTU/KW hr) y para 1960 a 2772 KCal/KW hr (11,000 BTU/KW hr). En la tabla 1 se pueden observar los consumos específicos de unidades generadoras de diferente tipo y capacidad. Estos datos fueron obtenidos en un estudio reciente realizado en CFE[2]

TIPO DE COMBUSTIBLE	CAPACIDAD (MW)	CONSUMO ESPECIFICO (KCal/KW hr)
Carbón	600	2260
Carbón	300	2420
Combustoleo	600	2220
Combustoleo	300	2375
Combustoleo	150	2450
Gas	50	3960

Tabla 1. Tipo, Capacidad y Consumos Especificos de Plantas

La efectividad con que una planta térmica cumple su propósito de conversión de energía se define por su curva de entrada-salida, en la que se representa la entrada (H) en el eje de las ordenadas en KCal/hr y en el eje de las abcisas la salida (P) en MW. Una curva típica de entrada-salida se muestra en la figura 1.

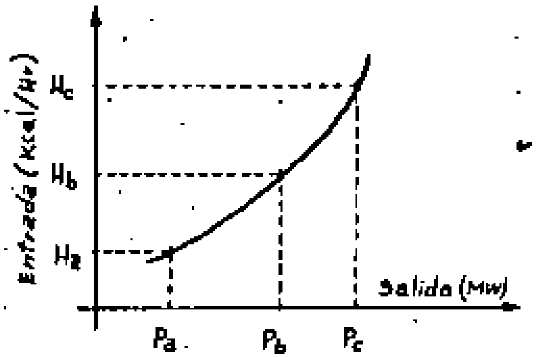


Figura 1. Curva Típica de Entrada-Salida

Para una salida dada el cociente de entrada entre salida recibe el nombre de consumo específico. Este cociente se puede graficar contra la salida como se muestra en la figura 2.

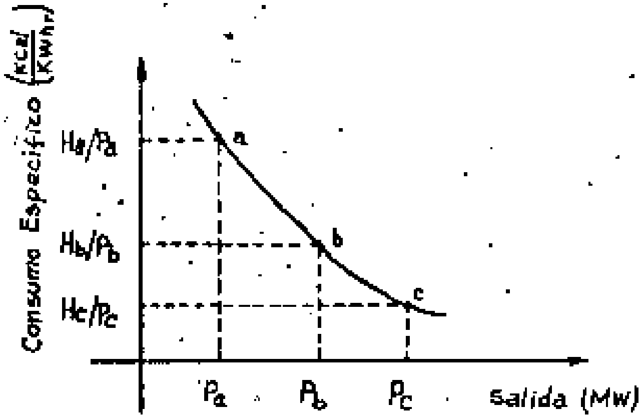


Figura 2. Curva de Consumo Especifico

La eficiencia de la unidad generadora se define como el cociente de la salida entre la entrada, de ahí que la eficiencia promedio para una salida dada sea inversamente proporcional al consumo específico.

Para obtener la curva de costo contra potencia generada se multiplica el costo del combustible dado en pesos por kilocaloría por la entrada en kilocaloría/hora y esta curva se representa en la figura 3.

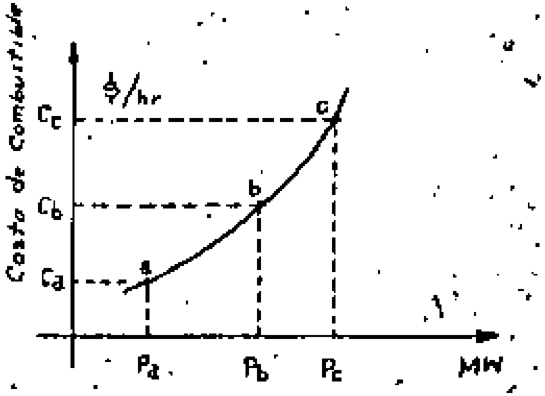


Figura 3. Curva Típica de Costo de Produccion Contra MW

PLANTEAMIENTO DEL PROBLEMA DE DESPACHO ECONOMICO SIN PERDIDAS

Tal como se mencionó con anterioridad, el objetivo del despacho económico es el de satisfacer con costo mínimo una demanda determinada. Matemáticamente el problema se plantea como un problema de optimización en

los siguientes términos:

minimizar : $C_T(PG_i) = \sum_{i=1}^{g} C_i(PG_i)$ (1)

respetando la restricción de satisfacer la demanda :

$$\sum_{i=1}^{g} PG_i = PD \qquad (2)$$

y las restricciones de operación de las unidades generadoras .

$$PG_i^{min} \leq PG_i \leq PG_i^{max} \qquad (3)$$

en donde:

C_T = Costo total de generación

$C_i(PG_i)$ = Costo de generar "PG_i" MW con la i-ésima unidad (Fig. 3)

PG_i = Nivel de generación de la unidad "i"

PD = Demanda total del sistema

PG_i^{min} = Límite inferior de operación de la unidad "i"

$PG_i^{máx}$ = Límite superior de operación de la unidad "i"

g = Número de generadores en el sistema

La técnica básica en una gran cantidad de métodos de optimización es la de convertir el problema con restricciones en un problema de minimización sin restricciones. Esta conversión se logra introduciendo una nueva variable por cada restricción de igualdad en el problema (del tipo de (2)) . En nuestro caso tendríamos

minimizar $L(PG_i, \lambda) = C_T(PG_i) + \lambda(PD - \sum_{i=1}^{g} PG_i)$ (4)

en donde la nueva función objetivo es L y el número de variables se ha incrementado a g + 1. Es un hecho conocido que el mínimo de una función como (4) se encuentra en el punto donde las parciales de dicha función con respecto a sus variables es cero, entonces :

$$\frac{\partial L}{\partial PG_i} = 0 \; ; \quad \frac{\partial C_T(PG_i)}{\partial PG_i} = \lambda \qquad (5)$$

$$\frac{\partial L}{\partial \lambda} = 0 \; ; \quad PD = \sum_{i=1}^{g} PG_i \qquad (6)$$

De la ecuación (5) se puede observar que la solución del problema de despacho se encuentra donde todos los costos incrementales de generación $\partial C/\partial PG$ son iguales y la ecuación (6) indica que esta solución cumple la restricción de satisfacer la demanda PD.

Un algoritmo para encontrar la solución al problema de despacho, simplificado al no tomar en consideración las pérdidas de transmisión, sería el siguiente:

1. Da un valor inicial a la variable λ
2. Encuentra el valor de cada PG_i de la ecuación 5
3. Si el valor encontrado de PG_i excede el límite superior entonces $PG_i = PG_i^{max}$; si el valor encontrado en PG_i es menor que el límite inferior entonces $PG_i = PG_i^{min}$. Esta verificación se hace para cada una de las unidades generadoras.
4. Si $\sum_{i=1}^{g} PG_i = PD$ termina el proceso . En caso contrario continúa en el paso 5.

5. Si $\sum_{i=1}^{g} PG_i > PD$ reduce el valor de λ .

   Si $\sum_{i=1}^{g} PG_i < PD$ incrementa el valor de λ .
6. Con el nuevo valor de λ regresa al punto 2.

Como se mencionó antes este algoritmo no toma en consideración las pérdidas de transmisión sin embargo, se incluyó por su sencillez que a la vez ilustra el mecanismo para obtener una distribución económica de la generación que satisface una demanda dada.

PLANTEAMIENTO DEL PROBLEMA DE DESPACHO ECONOMICO CON PERDIDAS DE TRANSMISION

Es posible a través de un conjunto de constantes conocidas como constantes B incluir las pérdidas de transmisión sin necesidad de conocer en el instante de cálculo la verdadera topología del sistema. Obviamente el método da resultados aproximados pero es un paso adelante del método anterior en donde las pérdidas son totalmente ignoradas. El planteamiento contiene ligeras variantes del anterior como se puede observar:

Minimizar $C_T(PG_i) = \sum_{i=1}^{g} C_i(PG_i)$ (7)

sujeto a : $\sum_{i=1}^{g} PG_i = PD + PL(PG_i)$ (8)

y $PG_i^{min} \leq PG_i \leq PG_i^{max}$ (9)

en donde $P_L(PG_i)$ es función de las pérdidas y está dada por :

$$PL\,(PG_i) = B_0 + \sum_{i=1}^{g} B_i PG_i + \sum_{i=1}^{g} \sum_{j=1}^{g} PG_i\, B_{ij}\, PG_j \qquad (10)$$

y las B's que intervienen en esta fórmula de pérdidas son constantes calculadas por fuera del proceso. El método de cálculo de estas constantes sale del objetivo de este trabajo y se puede estudiar en las referencias 3 y 4.

Nuevamente la técnica de solución consiste en minimizar una función objetivo a la que se han agregado las restricciones de igualdad, esto es:

$$\text{minimizar } \; L(PG_i\,,\lambda\,) = C_T\,(PG_i) + \lambda\,(PD + PL - \sum_{i=1}^{g} PG_i\,) \qquad (11)$$

Al igualar a cero las parciales de L con respecto a sus variables obtenemos el conjunto de ecuaciones.

$$\frac{\partial L}{\partial PG_i} = \frac{\partial C_T}{\partial PG_i} + \lambda\left(\frac{\partial PL}{\partial PG_i} - 1\right) = 0 \qquad (12)$$

$$y \quad \frac{\partial L}{\partial \lambda} = PD + PL - \sum_{i=1}^{g} PG_i = 0 \qquad (13)$$

$$\text{donde} \quad \frac{\partial PL}{\partial PG_i} = B_i + 2 \sum_{j=1}^{g} B_{ij}\, PG_j \qquad (14)$$

La ecuación 12 se puede re-escribir en la forma

$$\frac{1}{(1 - \dfrac{\partial PL}{\partial PG_i})} \; \frac{\partial C_T}{\partial PG_i} = \lambda \qquad (15)$$

Si comparamos esta ecuación con la ecuación 5 podemos observar que el efecto de las pérdidas de transmisión es el de ajustar los costos incrementales de cada unidad y que en la solución dichos costos incrementales

ajustados son todos iguales. El algoritmo para resolver el problema de despacho económico introduciendo el efecto de las pérdidas de transmisión a través de las constantes B es el siguiente:

1. Dar un valor inicial a λ

2. Resuelve la ecuación 15 para obtener los valores de todas las PG.

3. Compara el valor obtenido para cada PG_i con sus límites de operación y :

si $PG_i > PG_i^{max} \rightarrow PG_i = PG_i^{max}$

si $PG_i < PG_i^{min} \rightarrow PG_i = PG_i^{min}$

4. Calcula de la ecuación 10 el valor de las pérdidas PL.

5. Si: $\sum_{i=1}^{g} PG_i = PD + PL$ termina; en caso contrario continúa en el paso 6.

6. Si $\sum_{i=1}^{g} PG_i > PD + PL$ reduce el valor de λ.

   Si $\sum_{i=1}^{g} PG_i < PD + PL$ incrementa el valor de λ.

7. Con el nuevo valor de λ regresa al paso 2.

El siguiente grado de complejidad en la solución del problema de despacho económico consiste en incluir la solución de la red para considerar de manera exacta las pérdidas de transmisión. Obviamente el uso de esta técnica, denominada flujos óptimos por incluir la solución del problema de flujos de carga en el proceso de optimización, requiere de mayor información que las técnicas anteriores a saber, la topología del sistema, los parám

tros de las líneas y las inyecciones de potencia en los diferentes nodos de la red.

FLUJOS OPTIMOS EN LA SOLUCION DEL PROBLEMA DE DESPACHO ECONOMICO

Los flujos de potencia en un sistema de N nodos se caracterizan por el conjunto de ecuaciones complejas

$$S_k^* = V_k \angle -\delta_k \sum_{m=1}^{N} \left(G_{km} + j B_{km} \right) V_m \angle \delta_m \qquad (16)$$

en donde

V_k : magnitud de voltaje en el nodo k

δ_k : ángulo de fase en el nodo k

$G_{km} + j B_{km}$: elemento de la matriz de admitancia nodal

$S_k^* = P_k^N - j Q_k^N$: potencia compleja neta inyectada en el nodo "k"

Utilizando la notación $P_k (V, \delta) - j Q_k (V, \delta)$ para el lado derecho, la ecuación 16 puede escribirse como un conjunto de 2N ecuaciones del tipo

$$P_k^N - P_k (V, \delta) = 0, \; k = 1, \ldots N \qquad (17)$$

$$Q_k^N - Q_k (V, \delta) = 0, \; k = 1, \ldots, N \qquad (18)$$

Como es sabido cada nodo de la red está caracterizado por cuatro variables, P_k^N, Q_k^N V_k y δ_k y en el problema de flujos se especifican dos de ellas mientras las otras dos son incógnitas. Dependiendo de que variables sean especificados los nodos de la red se pueden clasificar en tres tipos como se muestra en la tabla 2.

NODO TIPO	DATOS	INCOGNITAS
Referencia	V, δ	P^N, Q^N
Carga	P^N, Q^N	V, δ
Voltaje controlado	V, P^N	δ, Q^N

Tabla 2. Diferentes Tipos de Nodo en el Problema de Flujos

Estas variables pueden clasificarse como variables dependientes (x), variables de control (u) y parámetros (p), esto es :

$$[x] = \left[\begin{array}{ll} \cdot V & \text{en c/u de los nodos de carga} \\ \cdot \delta & \\ \cdot \delta & \text{en c/u de los nodos de v. controlado} \end{array} \right]$$

$$[u] = \left[\begin{array}{lll} \cdot V & : & \text{en el nodo de referencia} \\ \cdot V & : & \text{en c/u de los nodos de voltaje controlado} \\ \cdot PG & : & \text{potencia de generación disponibles para despacho económico} \\ \cdot \text{etc.} & & \end{array} \right]$$

$$[p] = \left[\begin{array}{ll} P^N, Q^N & \text{en nodos de carga} \\ \delta \quad : & \text{en el nodo de referencia} \\ \cdot \text{parámetros de líneas} & \\ \cdot \text{etc.} & \end{array} \right]$$

Si definimos en términos de x, u, p, el conjunto de ecuaciones de flujos tenemos :

$$\left[g\ (x,\ u,\ p) \right] = \begin{cases} \left.\begin{array}{l} \text{eq } 17 \\ \text{eq } 18 \end{array}\right\} \begin{array}{l} \text{para c/u de los nodos} \\ \text{de carga.} \end{array} \\ \\ \text{eq } 17 \quad \text{para cada uno de los nodos de v. controlado} \end{cases} \qquad (19)$$

podemos entonces plantear el problema de flujos óptimos de la siguiente manera

minimizar $C_T(x,u)$(Costo total de Gen.,) (20)

Sujeto a las ecuaciones de la red

$$g\ (x,\ u,\ p) \qquad (21)$$

y a los límites de operación de las variables de control

$$U^{min} \leq U \leq U^{max} \qquad (22)$$

Nuevamente al igual que en los casos anteriores introducimos tantas variables auxiliares como restricciones de igualdad haya en la ecuación (21) y convertimos el problema a un problema de minimización sin restricciones esto es ;

minimizar $L\ (x,\ u,\ \lambda\ ) = C\ (x,u) + \lambda^T g(x,\ u,\ p)$ (23)

Las condiciones necesarias para el mínimo de (23) estarán dadas por las siguientes ecuaciones:

$$\frac{\partial L}{\partial x} = \frac{\partial C_T}{\partial x} + \frac{\partial g}{\partial x}^T \lambda = 0 \qquad (24)$$

$$\frac{\partial L}{\partial u} = \frac{\partial C_T}{\partial u} + \frac{\partial g}{\partial u}^T \lambda = 0$$

$$\frac{\partial L}{\partial \lambda} = g\ (x,\ u,\ p) = 0 \qquad (25)$$

en donde el último conjunto de ecuaciones es nuevamente el conjunto de ecuaciones de flujos dado por (19).

Las ecuaciones (24), (25) y (19) son no-lineales y su solución debe obtenerse en forma iterativa. El método más simple para obtener la solución es el llamado método del gradiente cuya idea básica es la de pasar de una solución factible a otra moviéndose en la dirección contraria del gradiente ya que este vector apunta en la dirección en que la función objetivo L tiene mayor crecimiento (en nuestro problema el gradiente esta dado por la ecuación 25). El algoritmo básico para la solución de este problema es el siguiente:

1. Dar un valor inicial a las variables de control u
2. Con los valores presentes de u resuelve el conjunto de ecuaciones no lineales (19)
3. Encuentra el valor de λ resolviendo la ecuación
$$\lambda = -\ \frac{\partial g}{\partial x}^{T-1} \frac{\partial C_T}{\partial x}$$
4. Con el valor de λ calcula el gradiente
$$\nabla L = \frac{\partial C_T}{\partial u} + \frac{\partial g}{\partial u} \lambda$$
5. Si el gradiente es lo suficientemente pequeño, termina el proceso, si no continúa en el paso 6.

6. Encuentre un nuevo conjunto de variables de control

$$u^{nueva} = u^{anterior} + \Delta u$$

$$y \quad \Delta u = -\alpha \nabla L$$

7. Verifica que los nuevos valores de u no violan las restricciones y si:

$$u_i^{nuevo} > u_i^{max} \rightarrow u_i^{nuevo} = u_i^{max}$$

$$u_i^{nuevo} < u_i^{min} \longrightarrow u_i^{nuevo} = u_i^{min}$$

8. Con los nuevos valores de u encontrados en los pasos 6 y 7 regresa al paso 2.

En el paso 6 del algoritmo que es en el que se calculan los cambios al vector de control se introdujo una nueva variable α. Esta variable tiene la función de graduar el cambio para evitar problemas de oscilaciones alrededor del mínimo y existen técnicas para seleccionarla óptimamente[5] en cada ciclo de ajuste de las variables de control.

CONCLUSIONES

La diversidad en tipos, eficiencia y tamaños de las plantas termoeléctricas y los altos costos que los combustibles fósiles han alcanzado en los últimos años, han originado la necesidad de contar con técnicas de despacho de carga que permitan satisfacer la demanda a un mínimo costo. Los avances tecnológicos recientes que han dado pauta a la utilización de las computadoras digitales para el control en tiempo real de los sistemas de potencia han permitido la utilización de técnicas de despacho más sofisticadas que permiten una representación más exacta de las condiciones bajo las que opera el sistema y en consecuencia las posibilidades de obtener mayores ahorros en la operación aumentan considerablemente.

REFERENCIAS

1. N. Cohn, "Control of Interconnected Power Systems", capítulo 17 "Handbook of Automation Computation and Control," Vol. 3. John Wiley, 1961.

2. Estudio AJBCO-DLP-7730, Oficina de Estudios Especiales de la Gerencia de Estudios e Ingeniería Preliminar, CFE "Base de Costos para los Estudios de Expansión de la Generación", Enero 1977.

3. L.K.Kirchmayer, "Economic Operation of Power Systems", John Wiley, 1958.

4. A.M. Sasson, "Optimal Load Flow a Practical Outlook," en Application of Optimization Methods in Power System Engineering, IEEE Tutorial Course Text, No. 76 CH 1107 - 2 - PWR, 1976.

5. H.W. Dommel y W.F. Tinney. "Optimal Power Flow Solutions", IEEE Trans., Vol. PAS 87, No. 10 Octubre de 1968.

6. O.I. Elgerd, "Electric Energy Systems Theory: An Introduction", Capítulo 8, Mc. Graw Hill, 1971.

7. W.D. Stevenson Jr., "Elements of Power System Analysis" Capítulo 11, Mc. Graw Hill, 1962 (segunda edición).

'alo

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO OPTIMO DE SISTEMAS DE INGENIERIA

OPTIMACION DE MECANISMOS

DR. JORGE ANGELES ALVAREZ

MARZO, 1982

"SINTESIS DE MECANISMOS GENERADORES DE FUNCION CON TRANSMISION OPTIMA"

J. Angeles
División de Estudios de Posgrado
de la Facultad de Ingeniería, UNAM.
Apdo. Postal 70-256. México 20, D.F.

J.L. Hernández
Ford Motor Company, S.A.
Cuautitlán, Edo. de México.

Abstract

The linkage-synthesis problem for function generation is approached from the viewpoint of system optimization. An objective function is proposed, whose minimization guarantees an optimal transmission. The optimization method resorted to is the Complex Method, which showed excellent results as to speed of convergence. The procedure is illustrated with an example of application to a planar linkage, but the extension to space linkages is straightforward.

Resumen

El problema de la síntesis de mecanismos generadores de función es abordado desde el punto de vista de la optimación de sistemas. Se propone una función objetivo cuya minimización garantiza una transmisión óptima. El método de optimación utilizado es el Cómplex, que mostró excelentes resultados en cuanto a rapidez de convergencia. Se ilustra este procedimiento con un ejemplo de aplicación a un mecanismo plano, pero la extensión a mecanismos espaciales es inmediata.

Introducción

En el problema de síntesis de mecanismos para generación de funcion se trata de hallar un conjunto de valores $p_1,\ldots,p_n$ de parámetros de un mecanismo de topología dado, que produzcan un cojunto de pares de valores entrada-salida $\{(\psi_i,\phi_i)\}_1^n$, donde ψ y ϕ son variables que representan la entrada y la salida del mecanismo, respectivamente. Estas variables pueden ser desplazamientos angulares o lineales. Igualmente, los parámetros p_i $(i=1,\ldots,n)$ pueden ser distancias o ángulos. Por ejemplo, dado el mecanismo de la Fig 1, los parámetros son las longitudes a_1, a_2, a_3 y a_4 de los eslabones, mientras que la entrada y la salida son los ángulos ψ y ϕ, respectivamente. La ecuación que relaciona los parámetros con las variables de entrada y de salida es la ampliamente conocida ecuación de Frendenstein (ref. 1):

$$k_1-k_2\cos\phi+k_3\cos\psi+\cos(\phi-\psi)=0 \qquad (1)$$

donde

$$k_1=\frac{a_3^2-a_1^2-a_2^2-a_4^2}{2a_2a_4},\ k_2=\frac{a_1}{a_2},\ k_3=\frac{a_1}{a_4} \qquad (2)$$

Dado que la ec (1) contiene tres parámetros independientes, k_1, k_2 y k_3, sólo se puede generar 3 pares de valores (ψ_i,ϕ_i) por medio de ella. En efecto, sustituyendo esos 3 pares de valores en la ec. (1) se obtiene el siguiente sistema lineal de ecuaciones algebraicas,

$$\underline{A}\underline{k}=\underline{b} \qquad (3)$$

donde

$$\underline{A}=\begin{bmatrix}1 & -\cos\phi_1 & \cos\psi_1\\ 1 & -\cos\phi_2 & \cos\psi_2\\ 1 & -\cos\phi_3 & \cos\psi_3\end{bmatrix},\ \underline{k}=\begin{bmatrix}k_1\\ k_2\\ k_3\end{bmatrix},\ \underline{b}=\begin{bmatrix}-\cos(\psi_1-\phi_1)\\ -\cos(\psi_2-\phi_2)\\ -\cos(\psi_3-\phi_3)\end{bmatrix} \qquad (4)$$

El sistema de ecuaciones (3) se puede resolver por varios métodos, de los cuales el más eficiente es el de Gauss o descomposición LU (ref. 2). En caso de necesitar generar más de 3 pares de valores (ψ_i,ϕ_i), por ejemplo m>3, la matriz $\underline{A}$ de la ec. (3) resulta rectangular, de mx3 y el sistema de ecuaciones es sobredeterminado. En este caso, generalmente no existe una solución que satisfaga simultáneamente todas las ecuaciones, pero se puede hallar aquel valor de $\underline{k}$, $\underline{k}_0$, que minimice la norma cuadrática del error

$$\underline{e}=\underline{A}\underline{k}-\underline{b} \qquad (5)$$

La solución $\underline{k}_0$ se puede hallar de una manera muy eficiente mediante el método de reflexiones de Householder (ref.3), que ya ha sido aplicado con éxito a la síntesis de mecanismos para un número excesivo de puntos de precisión (ref.4).

Volviendo a la ec. (3), y considerando el caso en el que se tenga igual número de puntos de precisión que de parámetros a determinar, esto es, 3, es bien sabido (ref.5) que el sistema de ecuaciones tiene una solución única, por lo que el mecanismo sintetizado será único también. Por esta razón, es posible que este mecanismo tenga una operación pobre, como por ejemplo, una mala transmisión. La transmisión de un mecanismo es una cualidad asociada a la ventaja mecánica del mismo, o sea, la relación entre el momento obtenido a la salida y el momento

suministrado a la entrada. Una forma de cuantificar esa transmisión, ampliamente aceptada, es mediante el llamado ángulo de transmisión, μ, que aparece en la Fig 1. Mientras este ángulo más próximo esté de un valor de 90° o de 270°, mayor será la ventaja mecánica, anulándose ésta cuando aquel ángulo vale 0° o 180°. Es deseable, entonces, que el ángulo de transmisión adquiera valores que se desvíen lo menos posible de 90° o de 270°, según el caso. En seguida se discute como modificar la ecuación de Freudenstein para poder convertir el problema en uno de optimación.

El problema de síntesis óptima

La ecuación de Frendenstein, tal como aparece en la ec. (3), no permite ninguna optimación. Sin embargo, si los ángulos de entrada y de salida se miden no desde la línea determinada por el eslabón 1, sino desde líneas que formen ángulos α y β con esta línea, respectivamente, la ecuación de Freudenstein se transforma en

$$k_1-k_2\cos(\phi+\beta)+k_3\cos(\psi+\alpha)+\cos(\phi-\psi-\alpha+\beta)=0 \qquad (6)$$

donde, debido a que los ángulos α y β están aún indeterminados, se tiene un conjunto de 5 incógnitas. Para el problema de generación de 3 pares de valores entrada-salida, (ψ_i,ϕ_i), entonces, se puede asignar libremente valores a dos de esas incógnitas. Si esas dos incógnitas son α y β, éstas se pueden escoger de manera que produzcan el mejor ángulo de transmisión, μ. Una función positiva definida cuya minimización produce valores de μ próximos a 90° o a 180° es

$$z=\frac{1}{2\pi}\int_0^{2\pi}\cos^2\mu d\psi \qquad (7)$$

donde se ha supuesto que el eslabón de entrada, 2, gira vuelta completa. La condición que deben satisfacer las longitudes de los eslabones, para que el de entrada gire vuelta completa, está dada por las desigualdades (ref. 1,p.63)

$$a_2>a_1 \qquad (8)$$

$$a_4>a_1 \qquad (9)$$

$$a_3+a_4>a_2-a_1 \qquad (10)$$

$$a_3+a_4>a_1+a_2 \qquad (11)$$

El ángulo de transmisión está dado por (ref.6):

$$\cos\mu=-\frac{a_2^2-a_3^2-a_4^2+a_1^2-2a_1a_2\cos(\psi+\alpha)}{2a_3a_4} \qquad (12)$$

Así, el problema de optimación resultante es el siguiente: "Minimizar z dada por la ec.(7) sujeta a las restricciones de igualdad (3) y a las de desigualdad (8) a (11)".

Los métodos disponibles para resolver el problema de optimación propuesto son básicamente de dos tipos:

i) métodos de funciones de penalización y ii) métodos directos. Los métodos de funciones de penalización consisten en transformar el problema dado, que contiene restricciones de desigualdad, en una secuencia de problemas sin este tipo de restricciones, y hallar los valores óptimos de las variables de decisión (α y β en este caso) para cada uno de esos problemas. El óptimo del problema original, que contiene restricciones de desigualdad, se obtiene por extrapolación (ref.7). El método de funciones de penalización ya se ha usado con éxito en la síntesis de mecanismos (ref.8). Los métodos directos manejan las restricciones de desigualdad directamente, esto es, no transforman el problema en uno sin este tipo de restricciones.

En cualquier caso el método de optimación a seguir dependerá de cuántas derivadas, con respecto a las variables de decisión, se tengan disponibles, esto es

i) ninguna derivada se puede calcular.

ii) se pueden calcular sólo primeras derivadas

iii) se tiene acceso a derivadas hasta de orden 2.

Dentro de los métodos aplicables al primer caso se tiene el de Powell y el Cómplex. El de Powell (ref.9) calcula el óptimo de una función sin restricciones de disigualdad y sin requerir de derivadas, mientras que el Cómplex (ref.10) calcula el óptimo de una función sujeto a restricciones de desigualdad, también sin requerir de derivadas. Ambos métodos ya han sido probados en la síntesis de mecanismos (refs.11 y 12).

Si se tiene acceso a primeras derivadas, se tienen los métodos de gradiente y de cuasi Newton (refs.13 y 14). Finalmente, si se dispone hasta de segundas derivadas, se puede aplicar el método de Newton-Raphson (ref.15) funciones de penalización para calcular las raíces del gradiente de la función objetivo a la que se ha aumentado las funciones de penalización adecuadas.

En el ejemplo que sigue se utilizó el método Cómplex.

Ejemplo de aplicación

Se desea sintetizar un mecanismo plano RRRR como el que aparece en la Fig 1, que tenga una transmisión óptima, en el sentido de que minimice el valor RMS de cosμ, dado

por la ec. (7), de manera que su eslabón de entrada gire vuelta completa –desigualdades (8) a (11)–y genere la función

$\phi_1 = 30°$, $\psi_1 = 45°$

$\phi_2 = 150°$, $\psi_2 = 60°$

$\phi_3 = 270°$, $\psi_3 = 90°$

Como la ecuación de Freudenstein sólo contiene 3 parámetros independientes, a una de las 4 longitudes a_i se le puede dar un valor arbitrario. Hágase, por ejemplo

$$a_1 = 1$$

ya que cualquiera que sea la solución del problema, esta longitud siempre es positiva. La ecuación de Freudenstein se utiliza, desde luego, en la forma de la ec. (6), con α y β como variables de decisión, que permitan la optimización del mecanismo.

Con el objeto de simplificar los cálculos, escríbanse la función objetivo z y las restricciones (8) a (11) en términos de los parámetros k_i. Así,

$$a_2 = \frac{1}{k_2}$$

$$a_3 = \frac{\sqrt{D}}{k_2 k_3}$$

$$a_4 = \frac{1}{k_3} \qquad (13)$$

$$z = \frac{E}{2D}$$

donde

$$D = 2k_1k_2k_3+k_3^2+k_2^2+k_2^2k_3^2 \qquad (14)$$

y

$$E = 2(k_1k_3+k_2)^2+k_3^4 \qquad (15)$$

La matriz $\underline{A}$ y el vector $\underline{b}$ se transforman claramente en

$$\underline{A} = \begin{pmatrix} 1 & -\cos(\phi_1+\beta) & \cos(\phi_1+\alpha) \\ 1 & -\cos(\phi_2+\beta) & \cos(\phi_2+\alpha) \\ 1 & -\cos(\phi_3+\beta) & \cos(\phi_3+\alpha) \end{pmatrix}, \quad \underline{b} = \begin{pmatrix} -\cos(\phi_1-\psi_1-\alpha+\beta) \\ -\cos(\phi_2-\psi_2-\alpha+\beta) \\ -\cos(\phi_3-\psi_3-\alpha+\beta) \end{pmatrix} \qquad (16)$$

Las restricciones (8) a (11) se reducen a

$$k_2<1$$
$$k_3<1 \qquad (17)$$
$$k_4 \equiv k_2-k_3+D^{1/2}-k_2k_3>0$$

Adicionalmente, limítense los valores de α y β entre 0 y 2π. Así se tienen además las siguientes desigualdades

$$0\leq\alpha\leq 2\pi$$
$$\qquad (18)$$
$$0\leq\beta\leq 2\pi$$

Para la solución del problema de optimación propuesto se utilizó el paquete OPTIM (ref.16), que dio como solución los siguientes valores numéricos

a_2=	0.3774	u. de longitud
a_3=	0.7450	u. de longitud
a_4=	1.001	u. de longitud
α =	227.6°	
β =	157.1°	

El mecanismo correspondiente se muestra en la Fig 2 y la curva μ vs. ϕ, en la Fig 3.

Conclusiones y recomendaciones

El método Complex mostró una buena rapidez de convergencia, pues el número máximo de iteraciones requerido fue de 23. Sin embargo, debe tenerse en cuenta que el problema presentado sólo contiene dos variables de decisión. Si el número de estas variables aumenta es posible que el método presente dificultades para converger. En ese caso, debe ensayarse el método de Newton-Raphson con funciones de penalización y amortiguamiento, lo cual acelera notablemente la convergencia. La función objetivo utilizada es cuadrática y positiva definida, lo cual hace que tenga primeras y segundas derivadas continuas, por lo que se facilita su uso para el método de Newton-Raphson. El método Complex no requiere tal continuidad en esas derivadas y se pudo haber usado, en cambio, otra función objetivo como

$$z = \text{máx}|\cos\mu|$$

El hecho de haber usado una función cuadrática fue motivado por razones de comparación, en caso de que se desea ensayar con el método de Newton-Raphson, por ejemplo.

Referencias

1. Angeles J., Análisis y Síntesis Cinemáticos de Sistemas Mecánicos, Ed. Limusa, S.A., México, D.F., 1978, p. 45.

2. Forsythe G. y Moler C.B., Computer Solution of Linear Algebraic Systems, Prentice Hall, Inc., Englewood Cliffs, N.J., 1967, p. 27

3. Moler C.B., Matrix Eigenvalue and Least Square Computations, Computer Science Department, Stanford University, Stanford Cal.,1973, pp. 4.1-4.15

4. Angeles J., "Optimal Synthesis of linkages using Householder reflections", Proceedings of the Fifth World Congress on the Theory of Machines and Mechanisms, vol. 1, Montreal, Canadá, julio 8-13, 1979, pp. 111-114.

5. Finkbeiner D.T., Introduction to Matrices and Linear Transformations, W.H. Freeman and Co., San Francisco, 1966, p. 99

6. Denavit J. y R.S. Hartenberg, Kinematic Synthesis of Linkages, McGraw-Hill Book Co., N. York, 1964. p. 319.

7. Aoki M., Introduction to Optimization Techniques. Fundamental and Applications of Nonlinear Programming, The MacMillan Co., N. York, 1971, pp. 199-204

8. Alizade R.I., Novruzbekov I. G. y Sandor G.N., "Optimization of Four-bar function generating mechanisms using penalty functions with inequality and equality constraints", Mechanism and Machine Theory, vol. 10, 1975, pp. 327-336

9. Powell M. J. D., "An efficient method for finding the minimum of a function of several variables without calculating derivatives", Computer Journal, vol. 7, No. 4, 1964, pp. 155-162.

10. Box M.J., "A new method of constrained optimization and a comparisson with other methods", Computer Journal, vol. 8, 1965, pp. 42-52.

11. Suh C.H. y C.W. Radcliffe, Kinematics and Mechanisms Design, John Wiley and Sons, N. York, 1978, pp. 215-217.

12. Dukkipati R.V., Sankar S. y Osman M.O.M., "On the use of complex method of constrained optimization in linkage synthesis", Proceedings of the Fifth World Congress on the Theory of Machines and Mechanisms, vol. 1, julio 8-13, 1979, Montreal, Canadá, pp. 382-387.

13. Zoutendijk G., Methods of Feasible Directions, Elsevier Publishing Co., Amsterdam, 1960.

14. Fox R.L. and Gupta K. C., "Optimization technology as applied to mechanism design" J. Eng. Ind., Trans. ASME, Serie B, vol. 95, mayo 1973, pp. 657-663.

15. Isaacson E. y Keller H. B., Analysis of Numerical Methods, John Wiley and Sons, Inc., N. York, 1966, pp. 115-119

16. Evans L. B., Optimization Techniques for Use in Analysis of Chemical Processes, A Set of Notes, Massachussetts Institute of Technology, Cambridge (USA), 1971.

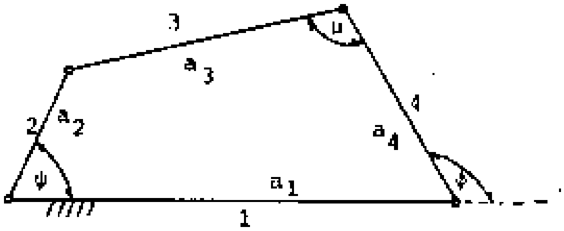


Fig 1. Mecanismo plano RRRR

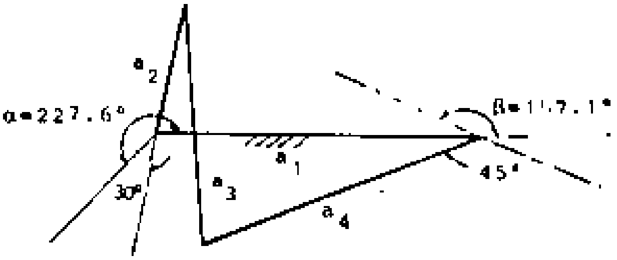


Fig 2. Mecanismo RRRR plano generador de función con transmisión óptima.

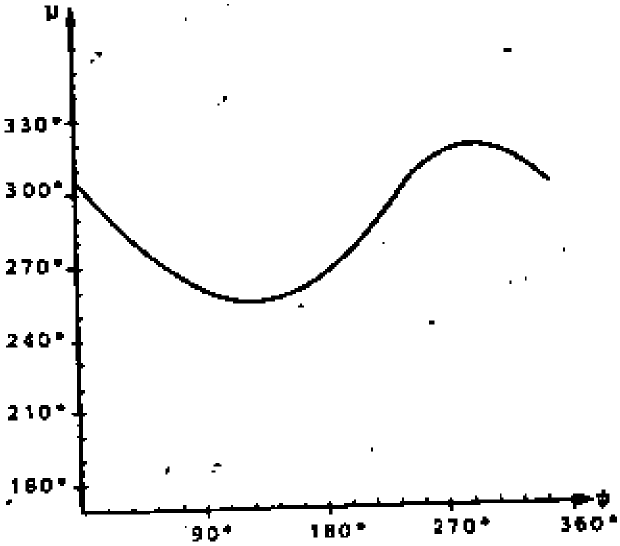


Fig 3. Curva de μ vs. ψ

Proceedings of the Fifth World Congress on Theory of Machines and Mechanisms – 1979
Published by The American Society of Mechanical Engineers

OPTIMAL SYNTHESIS OF LINKAGES USING HOUSEHOLDER REFLECTIONS

J. Angeles, Professor
National University of Mexico (UNAM)
Mexico, D. F., Mexico

ABSTRACT

The uncostrained overdetermined problem of kinematic linkage syethesis is solved in an efficient way using Householder reflections. The problem formulation leads to a system of either linear or nonlinear equations in more equations than unknowns. The linear problem is solved directly by application of a finite number of successive reflections to the space of unknowns with the purpose of taking the system of equations into upper triangular form, which allows for the computation of the unknowns by back substitution.The nonlinear problem is solved via the Newton-Raphson method which computes, at each iteration, the correction to the vector of unknowns as the least-square solution to an overdetermined linear system in exactly the same way as described before for linear problems. Introduction of the said method produces accurate results in relatively short processing times, as shown in the examples presented.

ZUSAMMENFASSUNG

Das uneingeschränkte und überbestimmte Problem der kinematischen Getriebesynthese wird effizient gelöst mit Hilfe der Householder-Spiegelungen. Die Problemstellung leitet zu einem System von entweder linearen oder nichtlinearen Gleichungen mit mehr Gleichungen als Unbekannten. Das lineare Problem wird direkt gelöst mittels Anwendung einer finiten Zahl aufeinanderfolgenden Spiegelungen zum Raum der Unbekannten mit dem Ziel des Übertragens des Gleichungssystems zu einer höheren dreieckigen Form, welche die Rechnung der Unbekannten durch Rückersetzung erlaubt. Das nicht-lineare Problem wird mitteles der Newton-Raphson-Methode gelöst, die zu jeder Iteration die Bessetungen der Unbekannten aus der wenigsten Quadraten-Lösung zu einem Überbestimmten linearen Gleichungssystem errechnet, auf der gleichen Weise wie bei der Methode für lineare Systeme schon beschrieben wurde. Die Einführung dieser Methode führt zu deutlichen Erfolgen in relativ kurzen Prozessierzeiten, wie mittels der eingeschloßenen Beispielen gezeigt wird.

NOMENCLATURE

$\underline{A}$: upper-case underlined character, an m×n matrix.
$\underline{A}^{-1}$: the inverse of $\underline{A}$, when $\underline{A}$ is square and nonsingular
$\underline{A}^T$: the transpose of A
$\underline{a}$: lower-case underlined latin character, an m-dimensional vector
$|a|$: the absolute value of a, when a is real; the modulus of a, when a is complex.
$||\underline{a}||$: the Eucliean norm of vector a, i.e. the square root of the sum of the squares of its components
detA: the determinant of the square matrix A
$\underline{f}(\underline{x})$: an m-dimensional vector function of the n-dimensional vector argument x
$\underline{f}'(\underline{x})$: the Jacobian m×n matrix of $\underline{f}$ with respect to $\underline{x}$

PROBLEM FORMULATION

The equations arising in the realm of kinematic synthesis of linkages constitute either linear or nonlinear algebraic[1] systems ($\underline{1}$,$\underline{2}$), whose unknowns are the geometric parameters (lengths and angles) of the linkage. If these paramenters are arranged within the n-dimensional vector $\underline{x}$, the said equations are of the form

$$\underline{A}\underline{x}=\underline{b} \qquad (1)$$

where A and b are a known m×n matrix and an m-dimensional known vector, respectively, when the system is linear. If it is nonlinear, then the synthesis equations are of the form

$$\underline{f}(\underline{x})=\underline{0} \qquad (2)$$

$\underline{f}$ being an m-dimensional vector containing a set of m scalar functions $f_i(\underline{x})$ whose arguments are the unknown parameters of the linkage. When the number of specified conditions to be met by the linkage matches that of the unknowns, matrix A in (1) is square and vector $\underline{f}$ is of dimension n. In most technical problems, however, the number of prescribed conditions surpasses that of geometric paramenters available, the linkage synthesis problem thus leading to an overdetermined system of equations. This class of systems in general does not admit an exact solution, but it is possible to find a vector $\underline{x}_o$ that renders the quadratic error a minimum. Thus, the least-squares problem can be stated as:

"Find the value of x that minimizes the Euclidean norm[2] of either Ax-b, or that of f(x), depending on whether the system is linear or nonlinear".

The linear overdetermined system (1) admits a unique solution $\underline{x}_o$ that renders $||\underline{A}\underline{x}-\underline{b}||$ a minimum, provided A is of full rank, i.e. if rank$\underline{A}$=n. This value is given as ($\underline{3}$)

$$\underline{x}_1=(\underline{A}^T\underline{A})^{-1}\underline{A}^T\underline{b} \qquad (3)$$

where $(\underline{A}^T\underline{A})^{-1}\underline{A}^T$ is called a "Moore-Penrose generalized

[1] Algebraic as opposed to differential or integral equations.

[2] See the nomenclature for the definition of this term.

inverse of A". An extensive treatment of the linear least-squares problem is found in (4).

The nonlinear problem may admit multiple local minima; these can be found by application of the Newton-Raphson method (5), which at each iteration, computes the correction vector Δx_k as the least-square solution to the overdetermined linear system

$$f'(x_k)\Delta x_k = -f(x_k) \qquad (4)$$

This is a system like that appearing in eq.(1). Thus, its least-square solution is

$$\Delta x_k = -\left[f'(x_k)^T f'(x_k)\right]^{-1} f'(x_k)^T f(x_k) \qquad (5)$$

The new value of the unknown vector is then

$$x_{k+1} = x_k + \Delta x_k \qquad (6)$$

The procedure is stopped when the Euclidean norm of the correction vector is sufficiently small within the imposed accuracy, i.e. when

$$||\Delta x|| \le \epsilon \qquad (6)$$

ϵ being a "small" real positive number. The problem thus, whether linear or nonlinear, reduces to compute the minimizing value x_o given by eq. (3). An efficient way of computing this value, outlined next, does not require to invert any matrix. The computation is done by application of Householder reflections.

HOUSEHOLDER REFLECTIONS

An extensive account of this topic can be found in the specialized literature (4,6). For this reason, this theory is not treated here. A Householder relfection is a linear, improper orthogonal and symmetric transformation, i.e., if H is its m×m matrix representation, then

$$H = H^T = H^{-1}, \det H = -1 \qquad (8)$$

When n such tranformations are defined suitably, bly, their effect on matrix A appearing in eq. (1) is to take it into upper triangular form. This way, the transformed equations are equivalent to the following

$$U x_o = c \qquad (9)$$

$$O x_o = d \qquad (10)$$

where U is an upper triangular n×n matrix and O is the (m-n)×n zero matrix, c and d being n-and (m-n)-dimensional vectors, with $d \neq 0$. Thus, eq. (9) is determined and can readily be solved by back substitution, its solution x_o being the least-square solution to the overdetermined system. Eq. (10) is inconsistent and $||d||$ represents the Euclidean norm of the error in the approximation. Since the original system (1) is transformed into (9),(10) via a succession of orthogonal transformations, the error in the transformed coordinates, d, has the same Euclidean norm as that in the original coordinates. Hence, $||d||$ is the error associated with the original system.

APPLICATIONS TO KINEMATIC LINKAGE SYNTHESIS

Although in many practical applications the problems of linkage synthesis involve inequality constraints, still a considerably large class of synthesis problems are unconstrained. Moreover, efficient optimization techniques exist that handle inequality constraints by introducing suitable penalty functions (7), thus turning the problem an unconstrained one. For these reasons, the study of unconstrained optimization problems is of substantial technical interest. Applications to linkage synthesis problems are next illustrated with two examples.

Example 1. Synthesis of an RSSR function generator

The layout of an RSSR linkage, shown in Fig 1, indicates the different geometric parameters of this linkage: a_1, a_2, a_3, are the lengths of the output-, coupler-and input links, respectively; a_4 is the distance between the axes of the input and the output links; α_4 is the angle between the aforementioned axes, positive about ED; s_1 and s_4 are distances of points C and O along the axes of the output-and the input links, respectively. All over, the sign convention of Denavit and Hartenberg (1,pp.344-345) is observed. The input angle is ψ and the output angle is ϕ. For matching six pairs of input-output values (ψ_j, ϕ_j) with this linkage, Denavit and Hartenberg (1,pp.358-362) established the following relation

$$k_1\cos\phi_j + k_2\sin\phi_j - k_3\cos\psi_j + k_4\sin\psi_j + k_5(\sin\phi_j\cos\psi_j - \cos\alpha_4\sin\psi_j\cos\phi_j) + k_6 = \cos\phi_j\cos\psi_j + \cos\alpha_4\sin\psi_j\sin\phi_j \qquad (11)$$

where

$$k_1 = \frac{a_4 + s_4\sin\alpha_4\tan\phi_0}{a_3},\quad k_2 = \frac{s_4\sin\alpha_4 - a_4\tan\phi_0}{a_3}$$

$$k_3 = \frac{a_4}{a_1\cos\phi_0},\quad k_4 = \frac{s_1\sin\alpha_4}{a_1\cos\phi_0};\quad k_5 = \tan\phi_0 \qquad (12)$$

$$k_6 = \frac{a_1^2 - a_2^2 + a_3^2 + a_4^2 + s_1^2 + s_4^2 + 2s_1 s_4\cos\alpha_4}{2a_1 a_3\cos\phi_0}$$

α_4 being assigned.

In the latter definitions, ϕ_0 measures the location of the zero of the output dial from the dotted line passing through C, parallel to line ED, as shown in Fig 1.

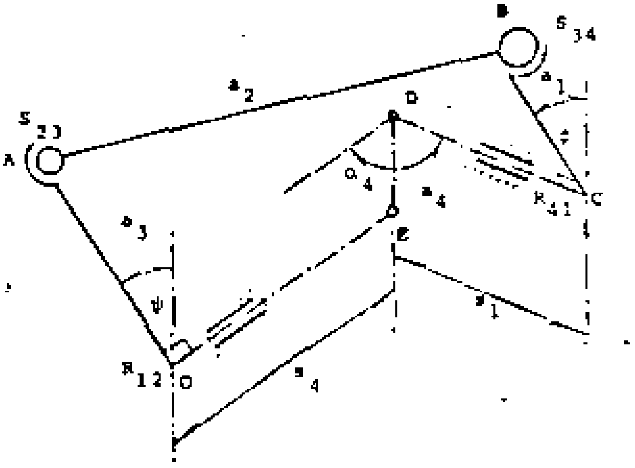


Fig 1 An RSSR linkage

For six precision-point synthesis, eqs. (11) yield a system of six linear equations in six unknowns which, when nonsingular, produces unique values $k_1, k_2, \ldots, k_6$. With these values known, the linkage parameters are computed from eqs. (12) for a given value of a_1. If more than six precision points are required, however, the system becomes overdetermined,

in which case an efficient method to obtain its least-square solution is via Householder reflections.

In (8), Suh and Mecklenburg solve the overdetermined unconstrained problem of this linkage with 19 prescribed input-output values. For comparison purposes, the solution developed in this example makes use of the same prescribed values. These are shown in Table 1

The method employed in (8) is that of Powell's (9), which does not require the computation of derivatives and leads quickly to convergence for quadratic functions of the independent variables. At this point, two remarks are in order: First, the derivatives of the synthesis equations are easily computed from either Devanit and Hartenberg's formulation, eqs. (11), or from Suh and Radcliffe's formulation (2), the first one being advantageous because of producing a linear system of equations. Second: The objective function of Suh and Mecklenburg's (8) is quadratic in the synthesis which, in turn, are quadratic in the independent variables; thus, their objective function is quartic in the independent variables, for which reason the quick convergence properties of Powell's method are not fully utilized. Furthermore, squaring the synthesis functions may introduce spurious local minima, as is apparent form the fact that three optimal solutions are reported in (8).

TABLE 1. Specified input-output pairs for the synthesis of the RSSR function generating linkage.

	φ(degreees)	ψ(degrees)
1	0.0	0.0
2	5.0	2.4
3	10.0	5.1
4	15.0	8.2
5	20.0	11.5
6	25.0	15.2
7	30.0	19.1
8	35.0	23.3
9	40.0	27.7
10	45.0	32.3
11	50.0	37.2
12	55.0	42.3
13	60.0	47.5
14	65.0	53.0
15	70.0	58.7
16	75.0	64.6
17	80.0	70.9
18	85.0	78.0
19	90.0	90.0

One advantage of using Householder reflections is that no explicit squaring is required, and the unique solution is obtained directly by the application of n(=6) reflections. Another advantage is that, since less computations are required, as compared to Powell's method, the round-off error is lowered. The approximation error obtained using each method is shown in Table 2.

The root mean square errors were essentially the same: that obtained by Powell's method was 0.00185269°, whereas the one obtained by Householder reflections, 0.00182254°. However, the differences in the resulting linkage parameters were more notorious. These are

Solution by Powell's method	Solution by Householder's method
a_1= 1.253803	a_1= 0.911269
a_2= 2.759566	a_2= 2.620568
a_3= 0.495003	a_3= 0.803577
a_1= 2.262110	a_1=-1.196259
a_4=-1.375210	a_4=-2.417356

In this problem, a_4 was set equal to 1, whereas a_4, equal to 90°.

TABLE 2. Approximation error in overdetermined RSSR linkage synthesis

	APROXIMATION ERROR USING POWELL'S METHOD	APROXIMATION ERROR USING HOUSEHOLDER'S METHOD
	(degrees)	(degrees)
1	0.00000000	0.01420363
2	-.00120000	0.00100735
3	-.03060000	-.03515343
4	0.02330000	0.01590827
5	-.02680000	-.03451603
6	0.03260000	0.02595072
7	0.01600000	0.01079245
8	0.03830000	0.03440434
9	0.01320000	0.01196148
10	-.03790000	-.04106664
11	-.00640000	-.00968497
12	0.02270000	0.01930349
13	-.04220000	-.04497542
14	-.00020000	-.00153544
15	0.03350000	0.03434290
16	0.01600000	0.01324614
17	0.00020000	0.00139267
18	-.01250000	-.01970407
19	0.02980000	0.00412793

Example 2. Synthesis of the RR plane dyad for rigid-body guidance

A rigid body (shaded rectangle) appears in Fig 2, in "reference" configuration C_0 and in a different configuration C_j. Each configuration is defined by the position of a point, R, and angle, θ. In that figure, O represents the origin of the complex plane, and the arrows represent complex numbers associated with the location of the labelled points. The purpose of this class of synthesis problem is to locate point A whose reference and successive positions, $A_0, A_j (j=1, ..., n)$ lie on a circumference centered at B, for which reason, A_0 and B are called, respectively, "circular" and "central" points, within the Burmester Theory (10). Thus, AB_0 can constitute a rigid link to guide the rigid body. This is an RR plane dyad.

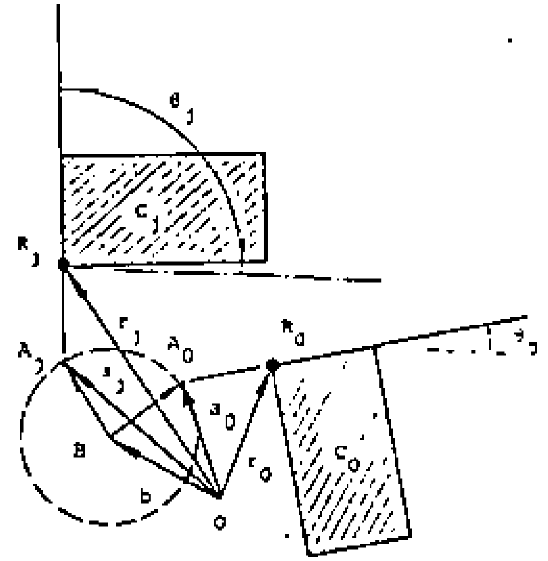


Fig 2 An RR plane dyad to guide a rigid body through n successive configurations

The constancy of the length of line BA throughout its n configurations leads to

$$|e^{i\theta'_j}(a_0-r_0)+r_j-b|^2=|a_0-b|^2, j=1,\dots,n \qquad (13)$$

where $\theta'_j\equiv\theta_j-\theta_0$. Eqs. (13) constitute the synthesis equations for this problem, a_0 and b being the unknowns. It is well known (2, p.146) that this problem allows to conduct a rigid body through five specified configurations. Some technical problems, however, may require to guide the body through more than five configurations, as shown in Table 3. Different syntheses were obtained for these, starting from the first 6 configurations, then adding the next ones, one at each time, until the 16 configurations were included

TABLE 3. Successive configurations of a rigid body

j	x_j(cm)	y_j(cm)	θ_j(degrees)
0	7.880	-0.260	313.720
1	8.490	-7.290	332.330
2	7.680	2.820	349.930
3	6.300	4.210	353.180
4	4.580	4.950	359.870
5	2.740	5.010	355.840
6	1.010	4.410	356.300
7	0.259	3.880	3.900
8	-0.400	-3.090	3.670
9	0.250	-3.760	3.690
10	1.000	-4.290	4.150
11	2.730	-4.890	5.120
12	4.560	-4.830	6.810
13	6.280	-4.090	10.000
14	7.660	-2.700	13.000
15	8.440	-0.610	18.000
16	7.790	-2.690	46.270

The procedure converged for all given inital guesses, produced by means of a random number generating subprogram, in less than 30 iterations (usually around 20). Contrary to the determined case (5 prescribed configurations), for which two different meaningful solutions exist, for the cases tried here the procedure converged always to the same single solution, except for 6 and 17 configurations, which produced two different solutions. The error in the approximation was normalized, to yield a dimensionless number, in the following way: Let

$$f_j=|e^{i\theta'_j}(a_0-r_0)+r_j-b|^2-|a_0-b|^2, j=1,\dots,m \qquad (14)$$

If the synthesis were exact, then all f_j would be negligibly small. In approximate syntheses, however, these functions attain finite values. The kinematic meaning of these values is that they represent the difference between the length of the RR dyad in its initial configuration, and that in its jth fonfiguration, i.e. $A_jB_j-A_0B_0$, if the synthesized linkage were to satisfy the prescribed conditions exactly. The dimensionless error in the approximation, e_j, associated with the jth configuration, is then

$$e_j=|f_j|/|a_0-b|^2, j=1,\dots,m \qquad (15)$$

where a_0 and b are those obtained from the least-square solution to the nonlinear system of equations. Notice that the errors thus defined are quadratic. To obtain a representative value of the overall error, the average of the square roots of the m errors defined in (14) should be taken, i.e.

$$e=\sum_1^m\sqrt{|f_j|}\ /\ n|a_0-b|$$

Some of the results obtained are shown next.

TABLE 4. Overdetermined synthesis of the RR dyad for rigid-body guidance.

For 6 configurations,

First solution:	Second solution:
a_0=-0.961467-i2.826960	a_0=7.690390+i2.700030
b=-1.643590-i7.997190	b=0.748466-i0.609952
Error = 17.02%	Error = 13.49%

For 17 configurations,

First solution:	Second solutoon:
a_0=5.123750+i2.254620	a_0=1.443950-i6.704520
b=0.549476-i7.03377	b=6.370810-i9.315060
Error = 38.74%	Error = 60.77%

CONCLUSIONS

Householder reflections apprar to be far more efficient in solving linear problems arising within the field of unconstrained optimal synthesis of linkages. As to nonlinear problems, the extension is straightforward. Regarding constrained problems, these could be handled using this method by introducing suitable slack variables and penalty functions. As to processor times, the first example consummed 11.8 sec, whereas the time reported (8) using Powell's method is 2.2 min, the method introduced here thus appearing to be more economical. With regard to the synthesis for rigid-body guidance, it is necessary to investigate whether for overdetermined problems, in general two different solutions can be expected, thus enabling the designer to synthesize RRRR plane linkages for overdetermined rigid-body guidance problems.

ACKNOWLEDGEMENTS

This research project was sponsored by the School of Enginnering at the National Autonomous University of Mexico. The computer programming was performed by Mr. Mario Siller. Householder's method was implemented via subroutines HECOMP and HOLVE, due to Moler (11)

REFERENCES

1. Denavit J. and Hartenberg R.S. Kinematic Synthesis of Linkages, Mc Graw-Hill, N. York, 1964
2. Suh C.H., and Radcliffe C.W., Kinematics and Mechanisms Design, Wiley, N. York, 1978
3. Ben-Israel A. and Greville T.N.E., Generalized Inverses: Theory and Applications, Wiley, N.York, 1974, pp.103-104
4. Stewart G.W., Introduction to Matrix Computations, Academic Press, N. York, 1973, pp.208-249
5. Björk Å. and Dahlquist G., Numerical Methods, Prentice-Hall, Englewood Cliffs, 1974, pp.443-444.
6. Businger P. and Golub G.H., "Linear Least Squares Solutions by Householder Transformations", in Wilkinson J.H. and Reinsch C., eds., Handbook for Automatic Computation, Vol II, Springer-Verlag, N. York, 1971, pp. 111-118
7. Fox R.L and Gupta K.C., "Optimization Technology as Applied to Mechanism Design", Journal of Engineering for Industry, Trans. ASME, Series B, Vol. 95, May 1973, pp. 657-663
8. Suh C.H. and Mecklenburg A.W., "Optimal Design of Mechanisms with the Use of Matrices and Least Squares", Mechanism and Machine Theory, Vol.8, pp. 479-495
9. Powell M.J.D., "An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives", Computer Journal, Vol.7, No.4, 1964, pp.303-307
10. Burmester K., Lehrbuch der Kinematik, Vol.1, „Die ebene Bewegung", Verlag von Arthur Felix, 1886
11. Moler C.B., Matrix Eigenvalue and Least-Square Computations, Computer Science Department, Stanford University, March 1973

DIVISION DE EDUCACION CONTINUA
FACULTAD DE INGENIERIA U.N.A.M.

DISEÑO OPTIMO DE SISTEMAS DE INGENIERIA

ANTECEDENTES MATEMATICOS Y NUMERICOS DE LAS TECNICAS DE OPTIMACION

DR. JORGE ANGELES ALVAREZ

MARZO 1982

1. MATHEMATICAL PRELIMINARIES

1.0 INTRODUCTION. Some relevant mathematical results are collected in this chapter. These results find a wide application within the realm of analysis, synthesis and optimization of mechanisms. Often, rigorous proofs are not provided; however a reference list is given at the end of the chapter, where the interested reader can find the required details.

1.1. VECTOR SPACE, LINEAR DEPENDENCE AND BASIS OF A VECTOR SPACE.

A vector space, also called a linear space, over a field F [1.1][*], is a set V of objects, called vectors, having the following properties:

a) To each pair $\{\underline{x}, \underline{y}\}$ of vectors from the set, there corresponds one (and only one) vector, denoted $\underline{x} + \underline{y}$, also from V, called "the addition of $\underline{x}$ and $\underline{y}$" such that

i) This addition is commutative, i.e.

$$\underline{x} + \underline{y} = \underline{y} + \underline{x}$$

ii) It is associative, i.e., for any element $\underline{z}$ of V,

$$\underline{x} + (\underline{y} + \underline{z}) = (\underline{x} + \underline{y}) + \underline{z}$$

iii) There exists in V a unique vector $\underline{0}$, called "the zero of V", such that, for any $\underline{x} \in V$,

$$\underline{x} + \underline{0} = \underline{x}$$

iv) To each vector $\underline{x} \in V$, there corresponds a unique vector $-x$, also in V, such that

$$\underline{x} + (-\underline{x}) = \underline{0}$$

[*] Numbers in brackets designate references at the end of each chapter.

b) To each pair $(\alpha ,\underline{x})$, where $\alpha \in F$ (usually called "a scalar") and $\underline{x} \in V$, there corresponds one vector $\alpha\underline{x} \in V$, called "the product of the scalar α times $\underline{x}$", such that:

i) This product is associative, i.e. for any $\beta \in F$,

$$\alpha(\beta\underline{x}) = (\alpha\beta)\underline{x}$$

ii) For the identity 1 of F (with respect to multiplication) the following holds

$$1\underline{x} = \underline{x}$$

c) The product of a scalar times a vector is distributive, i.e.

i) $\alpha(\underline{x} + \underline{y}) = \alpha\underline{x} + \alpha\underline{y}$

ii) $(\alpha + \beta)\underline{x} = \alpha\underline{x} + \beta\underline{x}$

Example 1.1.1. The set of triads of real numbers (x,y,z) constitute a vector space. To prove this, define two such triads, namely (x_1,y_1,z_1) and (x_2,y_2,z_2) and show that their addition is also one such triad and it is commutative as well. To prove associativity, define one third triad, (x_3,y_3,x_3), and so on.

Example 1.1.2 The set of all polynomials of a real variable, t, of degree less than or equal to n, for $0 \leq t \leq 1$, constitute a vector space over the field of real numbers.

Example 1.1.3 The set of tetrads of the form $(x,y,z,1)$ do not constitute a vector space (Why?)

Given the set of vectors $\{\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_n\} \subset V$ and the set of scalars $\{\alpha_1,\alpha_2,\ldots,\alpha_n\} \subset F$ not necessarily distinct, a linear combination of the n vectors is the vector defined as

$$\underline{c} = \alpha_1 \underline{x}_1 + \alpha_2 \underline{x}_2 + \ldots + \alpha_n \underline{x}_n$$

The said set of vectors is linearly independent (ℓ. i.) if $\underline{c}$ equals zero implies that all α's are zero as well. Otherwise, the set is said to be linearly dependent (ℓ. d.)

Example 1.1.4 The set containing only one nonzero vector, $\{\underline{x}\}$, is ℓ.i.

Example 1.1.5 The set containing only two vectors, one of which is the origin, $\{\underline{x},\underline{0}\}$, is ℓ.d.

The set of vectors $\{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n\} \subset V$ spans V if and only if every vector $\underline{v} \in V$ can be expressed as a linear combination of the vectors of the set.

A set of vectors $B = \{x_1, x_2, \ldots, x_n\} \subset V$ is a basis for V if and only if:

i) B is linearly independent, and

ii) B spans V

All bases of a given space V contain the same number of vectors. Thus, if B is a basis for V, the number n of elements of B is the dimension of V (abreviated: n=dim V)

Example 1.1.6 In 3-dimensional Euclidean space the unit vectors $\{\underline{i}, \underline{j}\}$ lying parallel to the X and Y coordinate axes span the vectors in the X-Y plane, but do not span the vectors in the physical three-dimensional space.

Exercise 1.1.1 Prove that the set B given above is a basis for V if and only if each vector in V can be expressed as a unique linear combination of the elements of B.

1.2 LINEAR TRANSFORMATION AND ITS MATRIX REPRESENTATION

Henceforth, only finite-dimensional vector spaces will be dealt with and, when necessary, the dimension of the space will be indicated as an exponent of the space, i.e., V^n means dim V=n.

A transformation $\underline{T}$, from an m-dimensional vector space U, to an n-dimensional vector space V is a rule which establishes a correspondence between an element of U and a unique element of V. It is represented as:

$$\underline{T}: U^m \rightarrow V^m \tag{1.2.1}$$

If $\underline{u} \in \underline{U}^m$ and $\underline{v} \in V^m$ are such that $\underline{T}: \underline{u} \rightarrow \underline{v}$, the said correspondence may also be denoted as

$$\underline{v} = \underline{T}(\underline{u}) \tag{1.2.3a}$$

$\underline{T}$ is linear if and only if, for any $\underline{u}$, $\underline{u}_1$ and $\underline{u}_2 \in U$, and $\alpha \in F$,

i) $\underline{T}(\underline{u}_1 + \underline{u}_2) = \underline{T}(\underline{u}_1) + \underline{T}(\underline{u}_2)$ and (1.2.3b)

ii) $\underline{T}(\alpha\underline{u}) = \alpha\underline{T}(\underline{u})$ (1.2.3c)

Space U^m over which $\underline{T}$ is defined is called the "domain" of T, whereas the subspace of V^m containing vectors $\underline{v}$ for which eq. (1.2.3a) holds is called the "range" of $\underline{T}$. A subspace of a given vector space V is a subset of V and is in turn a vector space, whose dimension is less than or equal to that of V

Exercise 1.2.1 Show that the range of a given linear transformation of a vector space U to a vector space V contitutes a subspace, i.e. it satisfies properties a) and b) of Section 1.1.

For a given $\underline{u} \in U$, vector $\underline{v}$, as defined by (1.2.2) is called the "image of $\underline{u}$ under $\underline{T}$", or, simply, the "image of $\underline{u}$" if $\underline{T}$ is selfunderstood.

An example of a linear transformation is an orthogonal projection onto a plane. Notice that this projection is a transformation of the three-dimensional Euclidean space onto a two-dimensional space (the plane). The domain of $\underline{T}$ in this case is the physical 3-dimensional space, while its range is the projection plane.

If $\underline{T}$, as defined in (1.2.1), is such that all of V contains $\underline{v}$'s such that (1.2.2) is satisfied (for some $\underline{u}$'s), $\underline{T}$ is said to be "onto". If $\underline{T}$ is such that, for all distinct $\underset{\sim}{u}_1$ and $\underset{\sim}{u}_2$, $\underset{\sim}{T}(\underset{\sim}{u}_1)$ and $\underset{\sim}{T}(\underset{\sim}{u}_2)$ are also distinct, $\underset{\sim}{T}$ is said to be one-to-one. If $\underset{\sim}{T}$ is onto and one-to-one, it is said to be invertible.

If $\underset{\sim}{T}$ is invertible, to each $\underset{\sim}{v} \in V$ there corresponds a unique $\underset{\sim}{u} \in U$ such that $\underset{\sim}{v} = \underset{\sim}{T}(\underset{\sim}{u})$, so one can define a mapping $\underset{\sim}{T}^{-1} : V \rightarrow U$ such that

$$\underset{\sim}{u} = \underset{\sim}{T}^{-1}(\underset{\sim}{v}) \qquad (1.2.4)$$

$\underset{\sim}{T}^{-1}$ is called the "inverse" of $\underset{\sim}{T}$.

Exercise 1.2.2 Let $\underset{\sim}{P}$ be the projection of the three-dimensional Euclidean space onto a plane, say, the X-Y plane. Thus, $\underset{\sim}{v} = \underset{\sim}{P}(\underset{\sim}{u})$ is such that the vector with components $(x, y, z)'$, is mapped into the vector with components $(x, y, 0)$.

i) Is $\underset{\sim}{P}$ a linear transformation?

ii) Is $\underset{\sim}{P}$ onto?, one-to-one?, invertible?

A very important fact concerning linear transformations of finite dimensional vector spaces is contained in the following result:

Let $\underset{\sim}{L}$ be a linear transformation from U^m to V^n. Let B_u and B_v be bases for U^m and V^n, respectively. Then clearly, for each $\underset{\sim}{u}_i \in B_u$ its image $\underset{\sim}{L}(\underset{\sim}{u}_i) \in V$ can be expressed as a linear combination of the $\underset{\sim}{v}_k$'s in B_v. Thus

$$L(\underset{\sim}{u}_i) = \alpha_{1i}\underset{\sim}{v}_1 + \alpha_{2i}\underset{\sim}{v}_2 + \ldots + \alpha_{ni}\underset{\sim}{v}_n \qquad (1.2.5)$$

Consequently, to represent the images of the m vectors of B_u, mn scalars like those appearing in (1.2.5) are required. These scalars can be arranged in the following manner:

$$[\underset{\sim}{A}] = \begin{bmatrix} \alpha_{11} & \alpha_{12} \ldots \alpha_{1m} \\ \alpha_{21} & \alpha_{22} \ldots \alpha_{2m} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \alpha_{n1} & \alpha_{n2} \ldots \alpha_{nm} \end{bmatrix} \qquad (1.2.6)$$

where the brackets enclosing $\underline{A}$ are meant to denote a matrix, i.e. an array of numbers, rather than an abstract linear transformation. $[\underline{A}]$ is called "The matrix of $\underline{L}$ referred to B_u and B_v" . This result is summarized in the following:

DEFINITION 1.2.1 *The i th column of the matrix representation of $\underline{L}$, referred to B_u and B_v, contains the scalar coefficients a_{ji} of the representation (in terms of B_v) of the image of the i th vector of B_u"*

Example 1.2.1 What is the representation of the reflexion $\underline{R}$ of the 3-dimensional Euclidean space E^3 into itself, with respect to one plane, say the X-Y plane, referred to unit vectors parallel to the X,Y,Z axes?.

Solution: Let $\underline{i}$, $\underline{j}$, $\underline{k}$, be unit vectors parallel to the X, Y and Z axes, respectively. Clearly,

$$\underline{R}(\underline{i}) = \underline{i}$$

$$\underline{R}(\underline{j}) = \underline{j}$$

$$\underline{R}(\underline{k}) = -\underline{k}$$

Thus, the components of the images of $\underline{i}$, $\underline{j}$ and $\underline{k}$ under $\underline{R}$ are:

$$\{\underline{R}(\underline{i})\} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \{\underline{R}(\underline{j})\} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \{\underline{R}(\underline{k})\} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$$

Hence, the matrix representation of $\underline{R}$, denoted by $[\underline{R}]$, is

$$[\underline{R}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \qquad (1.2.7)$$

Notice that, in this case, U = V and so, it is not necessary to use two different bases for U and V. Thus, $[\underline{R}]$, as given by (1.2.7), is the matrix representation of the reflection $\underline{R}$ under consideration, referred to the basis $\{\underline{i}, \underline{j}, \underline{k}\}$.

1.3 RANGE AND NULL SPACE OF A LINEAR TRANSFORMATION

As stated in Section 1.2, the set of vectors $\underline{v} \in V$ for which there is at least one $\underline{u} \in U$ such that $\underline{v} = \underline{L}(\underline{u})$ is called "the range of $\underline{L}$" and is represented as $\underline{R}(\underline{L})$, i.e. $\underline{R}(L) = \{\underline{v}=\underline{L}(\underline{u}): \underline{u} \in U\}$.

The set of vectors $\underline{u}_0 \in U$ for which $\underline{L}(\underline{u}_0) = \underline{0} \in V$ is called "the null space of L" and is represented as $N(\underline{L})$, i.e. $N(\underline{L}) = \{\underline{u}_0 : \underline{L}(\underline{u}_0)=\underline{0}\}$.

It is a simple matter to show that $R(\underline{L})$ and $N(\underline{L})$ are subspaces of V and U, respectively*.

The dimensions of $\text{dom}(\underline{L})$, $R(\underline{L})$ and $N(\underline{L})$ are not independent, but they are related (see [1.2]):

$$\dim \text{dom}(\underline{L})=\dim R(\underline{L}) + \dim N(\underline{L}) \qquad (1.3.1)$$

Example 1.3.1 In considering the projection of Exercise 1.2.1, U is E^3 and thus $R(\underline{P})$ is the X-Y plane, $N(\underline{P})$ is the Z axis, hence of dimension 1. The X-Y plane is two-dimensional and $\text{dom}(\underline{L})$ is three-dimensional, hence (1.3.1) holds.

Exercise 1.3.1 Describe the range and the null space of the reflection of Example 1.2.1 and verify that eq. (1.3.1) holds true.

1.4 EIGENVALUES AND EIGENVECTORS OF A LINEAR TRANSFORMATION

Let $\underline{L}$ be a linear transformation of V into itself (such an $\underline{L}$ is called an "endomorphism"). In general, the image $\underline{L}(\underline{v})$ of an element $\underline{v}$ of V is linearly independent with $\underline{v}$, but if it happens that a nonzero vector $\underline{v}$ and its image under $\underline{L}$ are linearly dependent, i.e. if

$$\underline{L}(\underline{v}) = \lambda \underline{v} \qquad (1.4.1)$$

* The proof of this statement can be found in any of the books listed in the reference at the end of this chapter.

such a $\underset{\sim}{v}$ is said to be an eigenvector of $\underset{\sim}{L}$, corresponding to the eigenvalue λ. If $[\underset{\sim}{A}]$ is the matrix representation of $\underset{\sim}{L}$, referred to a particular basis then, dropping the brackets, eq. (1.4.1) can be rewritten as

$$\underset{\sim}{A}\underset{\sim}{v} = \lambda \underset{\sim}{v} \qquad (1.4.2)$$

or else

$$(\underset{\sim}{A} - \lambda \underset{\sim}{I})\underset{\sim}{v} = \underset{\sim}{0}\,. \qquad (1.4.3)$$

where $\underset{\sim}{I}$ is tho identity matrix, i.e. the matrix with the unity on its diagonal and zeros elsewhere. Equation (1.4.3) states that the eigenvectors of $\underset{\sim}{L}$(or of $\underset{\sim}{A}$, clearly) lie in the null space of $\underset{\sim}{A} - \lambda \underset{\sim}{I}$. One trivial vector $\underset{\sim}{v}$ satisfying (1:4.3) is, of course, $\underset{\sim}{0}$, but since in this context $\underset{\sim}{0}$ has been discarded, nontrivial solutions have to be sought. The condition for (1.4.3) to have nontrivial solutions is, of course, that the determinant of $\underset{\sim}{A} - \lambda \underset{\sim}{I}$ vanishes, i.e.

$$\det\,(\underset{\sim}{A} - \lambda \underset{\sim}{I}) = 0 \qquad (1.4.4)$$

which is an nth order polynomial in λ, n being the order of the square matrix $\underset{\sim}{A}$ (1.3). The polynomial

$$P(\lambda) \equiv \det\,(\underset{\sim}{A} - \lambda \underset{\sim}{I})$$

is called "the characteristic polynomial" of $\underset{\sim}{A}$. Notice that its roots are the eigenvalues of $\underset{\sim}{A}$. These roots can, of course, be real or complex; in case $P(\lambda)$ has one complex root, say λ_1, then $\overline{\lambda}_1$ is also a root of $P(\lambda)$, $\overline{\lambda}_1$ being the complex conjugate of λ_1. Of course, one or several roots could be repeated. The number of times that a particular eigenvalue λ_i is repeated is called the algebraic multiplicity of λ_i.

In general, corresponding to each λ_i there are several linearly independent eigenvectors of $\underset{\sim}{A}$. It is not difficult to prove (Try it!) that the ℓ.i. eigenvectors associated with a particular eigenvalue span a subspace. This subspace is called the "spectral space" of λ_i, and its dimension is called

[illegible]

"the geometric multiplicity of λ_i".

Exercise 1.4.1 Show that the geometric multiplicity of a particular eigenvalue cannot be greater than its algebraic multiplicity.

A Hermitian matrix is one which equals its transpose conjugate. If a matrix equals the negative of its transpose conjugate, it is said to be skew Hermitian. For Hermitian matrices we have the very imoortant result:

THEOREM 1.4.1 *The eigenvalues of a Hermitian matrix are real and its eigenvectors are mutually orthogonal (i.e. the inner product, which is discussed in detail in Sec. 1.8, of two distinct eigenvectors, is zero).*

The proof of the foregoing theorem is very widely known and is not presented here. The reader can find a proof in any of the books listed at the end of the chapter.

1.5 CHANGE OF BASIS

Given a vector $\underline{v}$, its representation $(v_1, v_2, \ldots, v_n)^T$ referred to a basis $B = \{\underline{\beta}_1, \underline{\beta}_2, \ldots, \underline{\beta}_n\}$, is defined as the ordered set of scalars that produce $\underline{v}$ as a linear combination of the vectors of B. Thus, $\underline{v}$ can be expressed as

$$\underline{v} = v_1\underline{\beta}_1 + v_2\underline{\beta}_2 + \ldots + v_n\underline{\beta}_n \qquad (1.5.1)$$

A vector $\underline{v}$ and its representation, though isomorphic* to each other, are essentially different entities. In fact, $\underline{v}$ is an abstract algebraic entity satisfying properties a) and b) of Section 1.1, whereas its representation is an array of numbers. Similarly, a linear transformation, $\underline{L}$, and its representation, $(\underline{L})_B$, are essentially different entities. A question that could arise naturally is: Given the representations $(\underline{v})_B$ and $(\underline{L})_B$ of $\underline{v}$ and L, respectively, referred to the basis B, what are the corresponding

* Two sets are isomorphic to each other if similar operations can be defined on their elements.

representations referred to the basis $C = \{\underline{\lambda}_1, \underline{\lambda}_2, \ldots, \underline{\lambda}_n\}$?

Let $(\underline{A})_B$ be the matrix relating both B and C, referred to B, i.e.

$$(\underline{A})_B = \begin{bmatrix} \alpha_{11} & \alpha_{12} \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \alpha_{2n} \\ \vdots & \vdots & \\ \alpha_{n1} & \alpha_{n2} & \alpha_{nn} \end{bmatrix} \tag{1.5.2}$$

and

$$\underline{\gamma}_1 = \alpha_{11}\underline{\beta}_1 + \alpha_{21}\underline{\beta}_2 + \ldots + \alpha_{n1}\underline{\beta}_n$$

$$\underline{\gamma}_2 = \alpha_{12}\underline{\beta}_1 + \alpha_{22}\underline{\beta}_2 + \ldots + \alpha_{n2}\underline{\beta}_n$$

$$\vdots$$

$$\underline{\gamma}_n = \alpha_{1n}\underline{\beta}_1 + \alpha_{2n}\underline{\beta}_2 + \ldots + \alpha_{nn}\underline{\beta}_n$$

Thus, calling v'_i the <u>ith</u> component of $(\underline{v})_C$, then

$$\underline{v} = v'_1\underline{\gamma}_1 + v'_2\underline{\gamma}_2 + \ldots + v'_n\underline{\gamma}_n \tag{1.5.4}$$

and, from (1.5.3), (1.5.4) leads to

$$\underline{v} = \sum_j v'_j \sum_i \alpha_{ij}\underline{\beta}_j \tag{1.5.5}$$

or, using index notation* for compactness,

$$\underline{v} = \alpha_{ij} v'_j \underline{\beta}_i \tag{1.5.6}$$

Comparing (1.5.1) with (1.5.6),

$$v_i = \alpha_{ij} v'_j \tag{1.5.7}$$

i.e.

$$(\underline{v})_B = (\underline{A})_B (\underline{v})_C$$

* According to this notation, a repeated index implies that a summation over all the possible values of this index is performed.

or, equivalently,

$$(\underline{v})_C = (\underline{A})_B^{-1}(\underline{v})_B \qquad (1.5.8)$$

Now, assuming that $\underline{w}$ is the image of $\underline{v}$ under $\underline{L}$,

$$(\underline{w})_B = (\underline{L})_B\ (\underline{v})_B \qquad (1.5.9)$$

or, referring eq. (1.5.9) to the basis C, instead,

$$(\underline{w})_C = (\underline{L})_C\ (\underline{v})_C \qquad (1.5.10)$$

Applying the relationship (1.5.8) to vector $\underline{w}$ and introducing it into eq. (1.5.10),

$$(\underline{A}^{-1})_B\ (\underline{w})_B = (\underline{L})_C\ (\underline{A})_B^{-1}(\underline{v})_B$$

from which the next relationship readily follows

$$(\underline{w})_B = (\underline{A})_B\ (\underline{L})_C\ (\underline{A})_B^{-1}\ (\underline{v})_B \qquad (1.5.11)$$

Finally, comparing (1.5.9) with (1.5.11),

$$(\underline{L})_B = (\underline{A})_B\ (\underline{L})_C\ (\underline{A})_B$$

or, equivalently,

$$(\underline{L})_C = (\underline{A})_B^{-1}\ (\underline{L})_B\ (\underline{A})_B \qquad (1.5.12)$$

Relationships (1.5.8) and (1.5.12) are the answers to the question posed at the beginning of this Section. The right hand side of (1.5.12) is a similarity transformation of $(\underline{L})_B$

Exercise 1.5.1 Show that, under a similarity transformation, the characteristic polynomial of a matrix remains invariant.

Exercise 1.5.2 The trace of a matrix is defined as the sum of the elements on its diagonal. Show that the trace of a matrix remains invariant under a similarity transformation Hint: Show first that, if $\underline{A}$, $\underline{B}$ and $\underline{C}$ are nxn matrices,

$$\mathrm{Tr}(\underline{A}\underline{B}\underline{C}) = \mathrm{Tr}(\underline{B}\underline{C}\underline{A}).$$

1.6 DIAGONALIZATION OF MATRICES

Let $\underset{\sim}{A}$ be a symmetric nxn matrix and $\{\lambda_i\}$ its set of n eigenvalues, some of which could be repeated. Assume $\underset{\sim}{A}$ has a set of n linearly independent* eigenvectors, $\{\underset{\sim}{e}_i\}$, so that

$$\underset{\sim}{A}\underset{\sim}{e}_i = \lambda_i\underset{\sim}{e}_i \tag{1.6.1}$$

Arranging the eigenvectors of $\underset{\sim}{A}$ in the matrix

$$\underset{\sim}{Q} = \left(\underset{\sim}{e}_1,\underset{\sim}{e}_2,\ldots,\underset{\sim}{e}_n\right) \tag{1.6.2}$$

and its eigenvalues in the diagonal matrix

$$\underset{\sim}{\Lambda} = \text{diag }(\lambda_1,\lambda_2,\ldots,\lambda_n) \tag{1.6.3}$$

eq. (1.6.1) can be rewritten as

$$\underset{\sim}{A}\underset{\sim}{Q} = \underset{\sim}{Q}\underset{\sim}{\Lambda} \tag{1.6.4}$$

Since the set $\{\underset{\sim}{e}_i\}$ has been assumed to be ℓ.i., $\underset{\sim}{Q}$ is non-singular; hence from (1.6.4)

$$\underset{\sim}{\Lambda}=\underset{\sim}{Q}^{-1}\underset{\sim}{A}\underset{\sim}{Q} \tag{1.6.5}$$

which states that the diagonal matrix containing the eigenvalues of a matrix $\underset{\sim}{A}$ (which has as many ℓ.i. eigenvectors as its number of columns or rows) is a similarity transformation of $\underset{\sim}{A}$; furthermore, the transformation matrix is the matrix containing the components of the eigenvectors of $\underset{\sim}{A}$ as its columns. On the other hand, if $\underset{\sim}{A}$ is Hermitian, its eigenvalues are real and its eigenvectors are mutually orthogonal. If this is the case and the set $\{\underset{\sim}{e}_i\}$ is normalized, i.e., if $||\underset{\sim}{e}_i|| = 1$, for all i, then

$$\underset{\sim}{e}_i^T\underset{\sim}{e}_j = 0,\ i \neq j \tag{1.6.6a}$$

$$\underset{\sim}{e}_i^T\underset{\sim}{e}_i = 1 \tag{1.6.6b}$$

* Some square matrices have less than n ℓ.i eigenvectors, but these are not considered here.

where $\underset{\sim}{e}_i^T$ is the transpose of $\underset{\sim}{e}_i$ ($\underset{\sim}{e}_i$ being a column vector, $\underset{\sim}{e}_i^T$ is a row vector). The whole set of equations (1.6.6), for all i and all j can then be written as

$$\underset{\sim}{Q}^T\underset{\sim}{Q} = \underset{\sim}{I} \qquad (1.6.7)$$

where $\underset{\sim}{I}$ is the matrix with unity on its diagonal and zeros elsewhere. Eq. (1.6.7) states a very important fact about $\underset{\sim}{Q}$, namely, that it is an orthogonal matrix. Summarizing, a symmetric nxn matrix $\underset{\sim}{A}$ can be diagonalized via a similarity transformation, the columns of whose matrix are the eigenvectors of $\underset{\sim}{A}$

The eigenvalue problem stated in (1.6.1) is solved by first finding the eigenvalues $\{\lambda_i\}_1^n$. These values are found from the following procedure: Write eq. (1.6.1) in the form

$$(\underset{\sim}{A} - \lambda_i\underset{\sim}{I})\underset{\sim}{e}_i=\underset{\sim}{0} \qquad (1.6.8)$$

This equation states that the set $\{\underset{\sim}{e}_i\}_1^n$ lies in the null space of $\underset{\sim}{A} - \lambda_i\underset{\sim}{I}$. For this matrix to have nonzero vectors in its null space, its determinant should vanish, i.e.

$$\det(\underset{\sim}{A}-\lambda_i\underset{\sim}{I})\equiv P(\lambda_i)=0 \qquad (1.6.9)$$

whose left hand side is its characteristic polynomial, which was introduced in section 1.4. This equation thus contains n roots, some of which could be repeated.

A very useful result is next summarized, though not proved.

THEOREM (Cayley-Hamilton). A square matrix satisfies its own characteristic equation, i.e. if $P(\lambda_i)$ is its characteristic polynomial, then

$$P(\underset{\sim}{A}) = \underset{\sim}{0} \qquad (1.6.10)$$

A proof ot this teorem can be found either in [1.3, pp. 148-150] or in [1.4, pp. 112-115]

Exercise 1.6.1 A square matrix A is said to be strictly lower triangular (SLT) if $a_{ij}=0$, for $j\geq i$. On the other hand, this matrix is said to be nillpotent of index k if k is the lowest integer for which $A^k = 0$.

i) Show that an nxn SLT matrix is nillpotent of index $k\leq n$.

ii) Show that an nxn SLT matrix A staisfies the following indentity:

$$(I+A)^{-1}=\sum_{1}^{n}(-1)^{k-1}A^{k-1}$$

The inverse of I+A appears very often in the solution of linear algebraic systems by iterative methods.

1.7. BILINEAR FORMS AND SIGN DEFINITION OF MATRICES.

Given that the space of matrices does not constitute an ordered set (as is the case for the real, rational or integer sets), it is not possible to attribute a sign to a matrix. However, it will be shown that, if a bilinear form (in particular, a quadratic form) is associated with a matrix, then it makes sense to speak of the sign of a matrix. Before proceeding further, some difinitions are needed. Let u and v ∈ U, U being a vector space defined over the complex fied F. A bilinear form of u and v, represented as $\phi(u,v)$ is a mapping from U into F, having the following properties:

i) It is linear in both u and v:

$$\phi(u_1+u_2,v) = \phi(u_1,v) + \phi(u_2,v) \tag{1.7.1a}$$

$$\phi(u,v_1+v_2) = \phi(u,v_1) + \phi(u,v_2) \tag{1.7.1b}$$

$$\phi(\alpha u,v)=\alpha\phi(u,v) \tag{1.7.1c}$$

$$\phi(u,\beta v)= \bar{\beta}\phi(u,v) \tag{1.7.1d}$$

where α and β ∈F, their conjugates being $\bar{\alpha}$ and $\bar{\beta}$, respectively.

ii) $\phi(\underline{v},\underline{u})$ is the complex conjugate of $\phi(\underline{u},\underline{v})$, i.e.

$$\phi(\underline{v},\underline{u}) = \bar{\phi}(\underline{u},\underline{v}) \qquad (1.7.1e)$$

The foregoing properties of conjugate bilinear forms suggest that one possible way of constructing a bilinear form is as follows:

Let

$$\phi(\underline{u},\underline{v}) = \underline{u}^{*}\underline{A}\underline{v} \qquad (1.7.2)$$

Exercise 1.7.1 Prove that definition (1.7.2) satisfies properties (1.7.1)

If, in (1.7.1), $\underline{v} = \underline{u}$, the bilinear form becomes the quadratic form

$$\phi(\underline{u}) = \underline{u}^{*}\underline{A}\ \underline{u} \qquad (1.7.3)$$

It will be shown that the bilinear form (1.7.2) defines a scalar product for a vector space under certain conditions on $\underline{A}$.

Definition: A scalar product, $p(\underline{u},\underline{v})$, of two elements for a vector space U is a complex number with the following properties:

i) It is Hermitian symmetric:

$$p(\underline{u},\underline{v}) = \bar{p}(\underline{v},\underline{u}) \qquad (1.7.4a)$$

ii) It is conjugate linear in both $\underline{u}$ and $\underline{v}$:

$$p(\underline{u}_1+\underline{u}_2,\underline{v}) = p(\underline{u}_1,\underline{v}) + p(\underline{u}_2,\underline{v}) \qquad (1.7.4b)$$

$$p(\underline{u},\underline{v}_1+\underline{v}_2) = p(\underline{u},\underline{v}_1) + p(\underline{u},\underline{v}_2) \qquad (1.7.4c)$$

$$p(\alpha\underline{u},\underline{v}) = \alpha p(\underline{u},\underline{v}) \qquad (1.7.4d)^{*}$$

$$p(\underline{u},\beta\underline{v}) = \bar{\beta} p(\underline{u},\underline{v}) \qquad (1.7.4e)$$

iii) It is real and positive definite:

$$p(\underline{u},\underline{u})>0, \text{ for } \underline{u}, \neq \underline{0} \qquad (1.7.4f)$$

$$p(\underline{u},\underline{u}) = 0, \text{ if and only if } \underline{u} = \underline{0} \qquad (1.7.4g)$$

* Note: conjugate linear in $\underline{v}$.

From definition (1.7.2) and properties (1.7.1), it follows that all that is needed for a bilinear form to constitute a scalar product for a vector space is that it is positive definite (and hence, real). Whether a bilinear form is positive definite or not clearly depends entirely on its matrix and not on its vectors. The following definition will be needed:

A square nxn matrix is said to be positive definite if (and only if), the quadratic form for any vector $\underset{\sim}{u} \neq \underset{\sim}{0}$ associated to it is real and positive and only vanishes for the zero vector. A positive definite matrix $\underset{\sim}{A}$ is symbolically designated as $\underset{\sim}{A} > 0$. If the said quadratic form vanishes for some nonzero vectors, then $\underset{\sim}{A}$ is said to be positive semidefinite, symbolically designated as $\underset{\sim}{A} \geq 0$. Negative definite and negative semidefinite matrices are similarly defined. Now:

THEOREM 1.7.1 *Any square matrix is decomposable into the sum of a Hermitian and a skew Hermitian part (this is called the Cartesian decomposition of the matrix)*

Proof. Write the matrix $\underset{\sim}{A}$ in the form

$$\underset{\sim}{A} \equiv \frac{1}{2}(\underset{\sim}{A}+\underset{\sim}{A}^*) + \frac{1}{2}(\underset{\sim}{A}-\underset{\sim}{A}^*) \qquad (1.7.5)$$

Clearly the first term of the right hand side is Hermitian and the second one is skew Hermitian.

THEOREM 1.7.2 *The quadratic form associated with a matrix $\underset{\sim}{A}$ is real if and only if $\underset{\sim}{A}$ is Hermitian. It is imaginary if and only if $\underset{\sim}{A}$ is skew Hermitian.*

Proof.

("if" part) Let $\underset{\sim}{A}$ be Hermitian; then

$$\phi(\underset{\sim}{u})=\underset{\sim}{u}^*\underset{\sim}{A}^*\underset{\sim}{u}=\underset{\sim}{u}^*\underset{\sim}{A}\underset{\sim}{u}$$

and

$$\bar{\phi}(\underset{\sim}{u})=\underset{\sim}{u}^*\underset{\sim}{A}\underset{\sim}{u}$$

Since

$$\text{Im}\,(\psi(\underline{u}))\equiv\frac{1}{2}\left(\psi(\underline{u})-\bar{\psi}(\underline{u})\right)$$

then

$$\text{Im}\,(\psi(\underline{u}))=0$$

On the other hand, if $\underline{A}$ is skew-Hermitian, then,

$$\psi(\underline{u})=\underline{u}^*\underline{A}^*\underline{u}=-\underline{u}^*\underline{A}\underline{u}$$

and

$$\bar{\psi}(\underline{u})=\underline{u}^*\underline{A}\underline{u}$$

Since

$$\text{Re}(\psi(\underline{u}))\equiv\frac{1}{2}\left(\psi(\underline{u})+\bar{\psi}(\underline{u})\right)$$

then

$$\text{Re}(\psi(\underline{u}))=0$$

thus proving the "if" part of the theorem.

Exercise 1.7.2 Prove the "only if" part of Theorem 1.7.2

What Theorem 1.7.2 states is very important, namely that Hermitian matrices are good candidates for defining a scalar product for a vector space, since the associated quadratic form is real. What is now left to investigate is whether this form turns out to be positive definite as well. Though this is not true for any Hermitian matrix, it is (obviously!) so for positive definite Hermitian matrices (by definition!). Futhermore, since the quadratic form of a positive definite matrix must, in the first place, be real, and since, for the quadratic form associated with a matrix to be real, the matrix must be Hermitian (from Theorem 1.7.2), it is not necessary to refer to a positive definite (or semidefinite) matrix as being Hermitian.

Summarizing: In order for the quadratic form (1.7.2) to be a scalar product, $\underline{A}$ must be positive definite. Next, a very important result concerning an easy characterization of positive definite (semidefinite) matrices is given.

THEOREM 1.7.3 *A matrix is positive definite (semidefinite) if and only if its eigenvalues are all real and greater than (or equal to) zero.*

Proof. ("only if" part).

Indeed, if a matrix A is positive definite (semidefinite), it must be Hermitian. Thus, it can be diagonalized (a consequence of Theorem 1.4.1). Furthermore, once the matrix is in diagonal form, the elements on its diagonal are its eigenvalues, which are real and greater than (or equal to) zero. It takes on the form

$$\underset{\sim}{A} = \begin{pmatrix} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \lambda_n \end{pmatrix} \qquad (1.7.10)$$

where

$$\lambda_i > (\geq)\ 0,\ i=1,2,\ldots,\ n$$

For any vector $\underset{\sim}{u} \neq \underset{\sim}{0}$, by definition,

$$\underset{\sim}{u}^*\underset{\sim}{A}^*\underset{\sim}{u}=\underset{\sim}{u}^*\underset{\sim}{A}\underset{\sim}{u}>(\geq)0 \qquad (1.7.11)$$

where the components of $\underset{\sim}{u}$ (with respect to the basis formed with the complete set of eigenvectors of $\underset{\sim}{A}$) are

$$\underset{\sim}{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} \qquad (1.7.12)$$

Substitution of (1.7.10) and (1.7.12) into (1.7.11) yields

$$\sum_{1}^{n}\lambda_i|u_i|^2>(\geq)0 \qquad (1.7.13)$$

Now, assume $\underline{u}$ is such that all but its $k^{\underline{th}}$ component vanish; in this case, (1.7.13) reduces to

$$\lambda_k|u_k|^2>(\geq)0$$

from which

$$\lambda_k>(\geq)0$$

and, since λ_k can be any of the eigenvalues of $\underline{A}$, the proof of this part is done. The proof of the "if" part is obvious and is left as an exercise for the reader.

<u>Exercise 1.7.2</u> Show that, if the eigenvalues of a square matrix are all real and greater than (or equal to) zero, the matrix is posite definite (semidefinite).

A very special case of a positive definite matrix is the identity matrix, $\underline{I}$, which yields the very well known scalar product

$$p(\underline{u},\underline{v})=\underline{u}^*\underline{I}^*\underline{v} = \underline{u}^*\underline{I}\underline{v}=\underline{u}^*\underline{v} \qquad (1.7.14)$$

In dealing with vector spaces over the real field, the arising inner product is real and hence, from Schwarz's inequality (1.4, p.125).

$$\frac{p(\underline{u},\underline{v})}{\sqrt{p(\underline{u},\underline{u})p(\underline{v},\underline{v})}}\leq 1$$

thus making it possible to define a "geometry" for then, the cosine of the angle between vectors $\underline{u}$ and $\underline{v}$ can be defined as

$$\cos(\underline{u},\underline{v})=\frac{p(\underline{u},\underline{v})}{\sqrt{p(\underline{u},\underline{u})p(\underline{v},\underline{v})}}$$

For vector spaces over the complex field, such an angle cannot be defined, for then the inner product is a complex number.

1.8 NORMS, ISOMETRIES, ORTHOGONAL AND UNITARY MATRICES.

Given a vector space V, a norm for $\underline{v} \in V$ is defined as a real-valued mapping from $\underline{v}$ into a real number, represented by $||\underline{v}||$, such that this norm

i) is positive definite, i.e.

$$||\underline{v}|| > 0, \text{ for any } \underline{v} \neq \underline{0}$$

$$||\underline{v}|| = 0 \text{ if and only if } \underline{v} = \underline{0}$$

ii) is linear homogeneous, i.e., for some $\alpha \in F$ (the field over which V is defined),

$$||\alpha\underline{v}|| = |\alpha|\,||\underline{v}||$$

$|\alpha|$ being the modulus (or the absolute value, in case α is real) of α.

iii) satisfies the triangle inequality, i.e. for $\underline{u}$ and $\underline{v} \in V$,

$$||\underline{u}+\underline{v}|| < ||\underline{u}|| + ||\underline{v}||$$

Example 1.8.1 Let v_i be the ith component of a vector $\underline{v}$ of a space over the complex field. The following are well defined norms for $\underline{v}$:

$$||\underline{v}|| = \max_{1<i<n} |v_i| \tag{1.8.1}$$

$$||\underline{v}|| = \left(\sum_{1}^{n} x_i|^p\right)^{\frac{1}{p}} \tag{1.8.2}$$

where p is a positive integer. For p = 2 in (1.8.2) the corresponding norm is the Euclidean norm, or the "magnitude" of $\underline{v}$.

Norm (1.8.1) is easy and fast to compute, and hence it is widely used in numerical computations. However, it is not suitable for physical or geometrical problems since it is not invariant*, i.e. it depends on the coordinate axes being used. The Euclidean norm has the advantage that it is invariant.

* Besides, there is no inner product associated with it and hence obviously no "geometry"

However, computing it requires n (the dimension of the space to which the vector under consideration belongs) multiplications (i.e. n square raisings), n-1 additions and one square root computation. In order to proceed further, some more definitions are needed.

An invertible linear transformation is called an "isometry" if it preserves the following scalar product

$$p(x,y) = p(Ax, Ay) = x^*A^*Ay \tag{1.8.3}$$

It is a very simple matter to show that, in order for a transformation P to be an isometry, it is required that its transpose conjugate, P^*, equals its inverse, i.e.,

$$P^*=P^{-1} \tag{1.8.4}$$

If P is defined over the complex field and meets condition (1.8.4), then it is said to be unitary. If P is defined over the real field, then $P^*=P^T$, the transpose of P and, if it satisfies (1.8.4), it is said to be orthogonal.

Exercise 1.8.1 Show that in order for P to be an isometry, it is necessary that P satisfies (1.8.4), i.e., show that under the similarity transformation

$$\zeta=Px,\ \eta = Py,\ B=PAP^{-1},$$

the following scalar product is preserved:

$$p(x,y) = p(\zeta,\eta)$$

1.9 PROPERTIES OF UNITARY AND ORTHOGONAL MATRICES.

Some important facts about unitary and orthogonal matrices are discussed in this section. Notice that all results concerning unitary matrices apply to orthogonal matrices, for the latter are a special case of the former.

THEOREM 1.9.1 *The set of eigenvalues of a unitary matrix lies on the unit circle $|z|^2 = 1$, centered at the origin of the complex plane.*

Proof: Let U be an nxn unitary matrix. Let λ be one of its eigenvalues and e a corresponding eigenvector, so that

$$Ue = \lambda e \tag{1.9.1}$$

Taking the transpose conjugate of both sides of (1.9.1),

$$e^*U^* = \bar{\lambda}e^* \tag{1.9.2}$$

Performing the corresponding products on both sides of eqs. (1.9.1) and (1.9.2),

$$e^*U^*Ue = \lambda\bar{\lambda}e^*e \tag{1.9.3}$$

But, since U is unitary, (1.9.3) leads to

$$e^*e = |\lambda|^2 e^*e$$

from which

$$|\lambda|^2 = 1, \text{q.e.d.}$$

<u>*Corollary 1.9.1*</u> *If an nxn unitary matrix is of odd order (i.e. n is odd), then it has at least one real eigenvalue, which is either + 1 or -1,*

<u>Exercise 1.9.1</u> Prove Crollary 1.9.1

1.10 <u>STATIONARY POINTS OF SCALAR FUNCTION OF A VECTOR ARGUMENT.</u>

Let $\phi = \phi(x)$ be a (scalar) real function of a vector argument, x, assumed to be continuos and differentiable up to second derivatives within a certain neighborhood around some x_0. The <u>stationary points</u> of this function are defined as those values x_0 of x where the gradient of ϕ, $\phi'(x)$ vanishes. Each stationary point can be an extremum or a <u>saddle point</u>. An extremum, in turn, can be either a local maximum or minimum. The function ϕ attains a local maximum at x_0 if and only if

$$f(x_0) \geq f(x)$$

for any x in the neighborhood of x_0, i.e., for any x such that

$$||x-x_0|| \leq \epsilon$$

ϵ being an arbitrarily small positive number. A local minimum is correspondingly defined. If an extremum is neither a local maximum nor a local minimum, it is said to be a saddle point. Criteria to decide whether an extremum is a maximum, a minimum or a saddle point are next derived.

An expansion of ϕ around $\underline{x}_0$ in a Taylor series illustrates the kind of stationary point at hand. In fact, the Taylor expansion of ϕ is

$$\phi(\underline{x}) = \phi(\underline{x}_0) + \phi'(\underline{x}_0)^T(\underline{x}-\underline{x}_0) + \frac{1}{2}(\underline{x}-\underline{x}_0)^T \phi''(\underline{x}_0)(\underline{x}-\underline{x}_0) + R \qquad (1.10.1)$$

where R is the residual, which contains terms of third and higher orders.

Then the increment of ϕ at x_0, for a given increment $\Delta x = x - x_0$, is given by

$$\Delta\phi = \phi'(\underline{x}_0)^T\Delta\underline{x} + \frac{1}{2}\Delta\underline{x}^T\phi''(\underline{x}_0)\Delta\underline{x} \qquad (1.10.2)$$

if terms of third and higher orders are neglected.

From eq. (1.10.2) it can be concluded that the linear pant of $\Delta\phi$ vanishes at a stationary point, which makes clear why such points are called stationary Whether $\underline{x}_0$ constitutes an extremum or not, depends on the sign of $\Delta\phi$. It is a maximum if $\Delta\phi$ is nonpositive for arbitrary $\Delta\underline{x}$. It is a minimum if the said increment is nonnegative for arbitrary $\Delta\underline{x}$. If the sign of the increment depends on $\Delta\underline{x}$, then $\underline{x}_0$ is a saddle point for reasons which are brought up in the following. Eq. (1.10.2) shows that the sign of $\Delta\phi$ depends entirely on the quadratic term, at a stationary point. Whether this term is nonpositive or nonnegative, it is sufficient that the Hessian matrix $\phi''(x)$ be sign semidefinite at x_0. Notice, however, that this condition on the Hessian matrix is only sufficient, but not necessary, for it is based on Eq. (1.10.2), which is truncated after third-order terms. In fact, a function whose Hessian at a stationary point is sign-semidefinite can constitute either a maximum, a minimum, or a saddle point as shown next.

From the foregoing discussion, the following theorem is concluded.

THEOREM 1.10.1 *Extrema and saddle points of a differentiable function occur at stationary points. For a stationary point to constitute a local maximum (minimum) it is sufficient, although not necessary, that the*

corresponding Hessian matrix be negative (positive) semidefinite. For the said point to constitute a saddle point, it is sufficient that the corresponding Hessian matrix sign-indefinite at this stationary point. A hypersurface in an n-dimensional space resembles a hyperbolic paraboloid at a saddle point, the resemblance lying in the fact that, at its stationary point, the sign of the curvature of the surface is different for each direction. To illustrate this, consider the hyperbolic paraboloid of Fig. 1.10.1 for which, when seen from the X-axis, its stationary point (the origin) appears as a minimum (positive curvature), whereas, if seen from the Y-axis, it appears as a maximum (negative curvature). In fact, it is none of these.

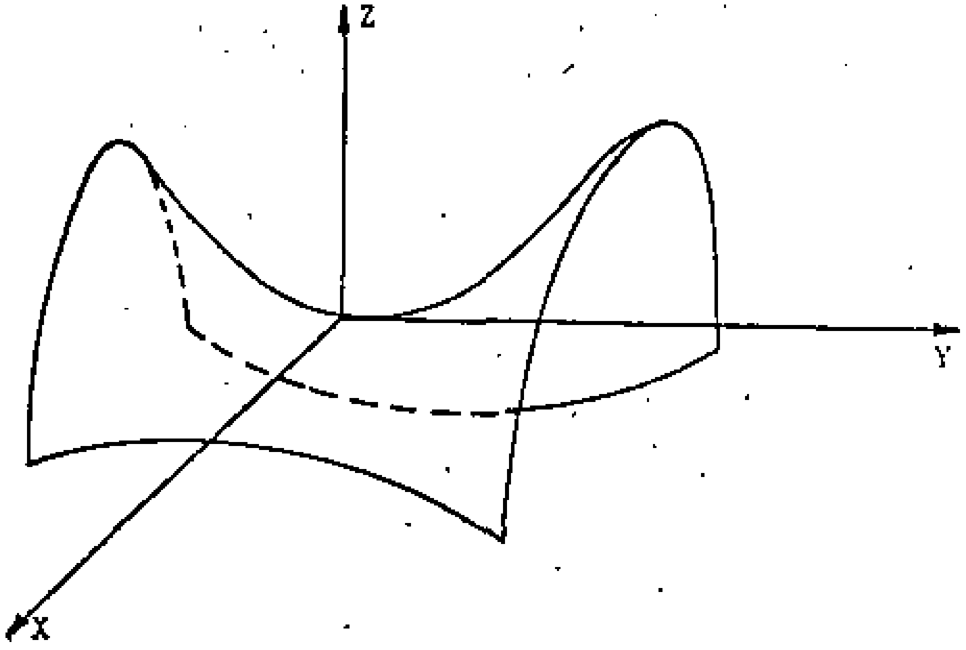


Fig. 1. 10.1 Saddle point of a 3-dimensional surface

Corollary 1.10.7 *The quadratic form*

$$\phi(\underline{x}) = \underline{x}^T\underline{A}\underline{x}+\underline{b}^T\underline{x}+c$$

has a unique extremum at $x_0=-\frac{1}{2}\underline{A}^{-1}\underline{b}$, *if* $\underline{A}^{-1}$ *exists. This is a maximum (minimum) if* $\underline{A}$ *is negative (positive) semidefinite*

Exercise 1.10.1 Prove Corollary 1.10.7

Example 1.10.1 The function $\phi = x_1^4 + x_2^4 + \ldots + x_n^4$ has a local minimum at $x_1 = x_2 = \ldots = x_n = 0$. The Hessian matrix of this function, however, vanishes at this minimum.

Example 1.10.2 The function $= x_1^4 - x_2^4$ has a stationary point at the origin, which is a saddle point. Its Hessian matrix, however, vanishes at this point.

Example 1.10.3 The function $x_1^2 + x_2^4$ has a minimum at $(0,0)$. At this point its Hessian matrix is positive semidefinite.

1.11 LINEAR ALGEBRAIC SYSTEMS.

Let $\underset{\sim}{A}$ be an mxn matrix and $\underset{\sim}{x}$ and $\underset{\sim}{b}$ be n-and m-dimensional vectors where, in general, $m \neq n$. Equation

$$\underset{\sim}{A}\underset{\sim}{x}=\underset{\sim}{b} \qquad (1.11.1)$$

is a linear algebraic system. It is linear because, if $\underset{\sim}{x}_1$ and $\underset{\sim}{x}_2$ are its solutions for $\underset{\sim}{b}=\underset{\sim}{b}_1$ and $\underset{\sim}{b}=\underset{\sim}{b}_2$, and α and β are scalars, then $\alpha\underset{\sim}{x}_1+\beta\underset{\sim}{x}_2$ is a solution for $\underset{\sim}{b}=\alpha\underset{\sim}{b}_1+\beta\underset{\sim}{b}_2$. It is algebraic as opposed to differential or dynamic because it does not involve derivatives. There are three different cases regarding the solution of eq. (1.11.1), depending on whether m is greater than, less than or equal to n. These are discussed next:

i) m>n. In this case the number of equations is greater than that of unknowns. The system is overdetermined and there is no guarantee of the existence of a certain $\underset{\sim}{x}_0$ such that $\underset{\sim}{A}\underset{\sim}{x}_0=\underset{\sim}{b}$.

A very simple example of such a system is the following:

$$x_1=5 \qquad (1.11.1a)$$

$$x_1=3 \qquad (1.11.1b)$$

where m=2 and n=1. If $x_1=5$, the first equation is satisfied but the second one is not. If, on the other hand, $x_1=3$, the second equation is satisfied, but the first one is not. However, a system with m>n could have a solution, which could even be unique if, out of the m equations involved, only n are linearly independent, the remaining m-n being linearly dependent on the n ℓ.i. equations. As an example, consider the following system

$$x_1+x_2=5 \tag{1.11.2a}$$

$$x_1-x_2=3 \tag{1.11.2b}$$

$$3x_1+x_2=13 \tag{1.11.2c}$$

whose (unique) solution is

$$x_1=4, x_2=1 \tag{1.11.3}$$

Here equation (1.11.2c) is linearly dependent on (1.11.2a) and (1.11.2b) In general, however, for m>n it is not possible to satisfy all the equations of a system with more equations than unknowns; but it is possible to "satisfy" them with the minimum possible error. Assume that $\underline{x}_0$ does not satisfy all the equations of an mxn system, with m>n, but satisfies the system with the least possible error. Let $\underline{e}$ be the said error, i.e.

$$\underline{e}=\underline{A}\underline{x}_0-\underline{b} \tag{1.11.4}$$

The Euclidean norm of $\underline{e}$ is

$$||\underline{e}||^2 = (\underline{A}\underline{x}_0-\underline{b})^T (\underline{A}\underline{x}_0-\underline{b}) \tag{1.11.5}$$

Expanding $||\underline{e}||^2$, it is noticed that it is a quadratic form of $\underline{x}_0$, i.e.

$$\phi(\underline{x}_0)=||\underline{e}||^2=\underline{x}_0^T\underline{A}^T\underline{A}\underline{x}_0-2\underline{b}^T\underline{A}\underline{x}_0+\underline{b}^T\underline{b} \tag{1.11.6}$$

The latter quadratic form has an extremum where $\phi'(\underline{x}_0)$ vanishes. The corresponding value of $\underline{x}$, $\underline{x}_0$, is found by setting $\phi'(\underline{x}_0)$ equal to zero, i.e.

$$\phi'(\underline{x}_0)=2\underline{A}^T\underline{A}\underline{x}_0-2\underline{A}^T\underline{b}=\underline{0} \tag{1.11.7}$$

If $\underline{A}$ is of full rank, i.e., if rank $(\underline{A})=n$, then $\underline{A}^T\underline{A}$, an nxn matrix, is also of rank n $\left(1.4\right)$, i.e. $\underline{A}^T\underline{A}$ is invertible and so, from eq. (1.11.5)

$$\underline{x}_0=(\underline{A}^T\underline{A})^{-1}\underline{A}^T\underline{b}=\underline{A}^I\underline{b} \tag{1.11.8}$$

where $\underline{A}^I$ is a "pseudo-inverse" of $\underline{A}$, called the "Moore-Penrose generalized inverse" of $\underline{A}$. A method to determine $\underline{x}_0$ that does not require the computation of $\underline{A}^I$ is given in (1.5) and (1.6). In (1.7), an iterative method to compute $\underline{A}^I$ is proposed. The numerical solution of this problem is presented in section 1.12. This problem arises in such fieds as control theory, curve-fitting (regressions) and mechanism synthesis.

ii) $m<n$. In this case the number of equations is less than that of unknowns. Hence, if the system is consitent*, it has an infinity of solutions. For instance, the system

$$x+y=3, \qquad (1.11.9)$$

in which $m=1$ and $n=2$, admits infinitely many solution, namely all points lying on the line

$$y=x+3 \qquad (1.11.10)$$

Now consider the system

$$x+y+z=1 \qquad (1.11.11a)$$

$$x+y-z=1 \qquad (1.11.11b)$$

with $m=2$ and $n=3$. This system admits an infinity of solutions all with $z=0$.

In case a system with $m<n$ admits a solution, it in fact admits infintely many, which is not difficult to prove. Indeed, partition matrix $\underline{A}$ and vector $\underline{x}$ in the form

$$\underline{A}=\left[\underbrace{A_1}_{m}\;\vdots\;\underbrace{A_2}_{n-m}\right]\,m,\quad \underline{x}=\begin{bmatrix}\underline{x}_1\\ -\,-\,-\\ \underline{x}_2\end{bmatrix}\begin{matrix}\}\,m\\ \\ \}\,n-m\end{matrix}$$

Thus, eq. (1.11.1) is equivalent to

$$\underline{A}_1\underline{x}_1+\underline{A}_2\underline{x}_2=\underline{b} \qquad (1.11.13)$$

* i.e. if $\underline{b}\in R(\underline{A})$

In the latter equation, if rank(A_1) = m, A_1^{-1} exists and a solution to (1.11.13) is

$$\underline{x}_1=\underline{A}_1^{-1}\underline{b}, \qquad \underline{A}_2\underline{x}_2=\underline{0} \tag{1.11.14}$$

where $\underline{x}_1$ is unique, as will be shown for the case m=n, and $\underline{x}_2$ is a vector lying in the null space of $\underline{A}_2$. Clearly, there are as many linearly independent solutions (1.11.12) as linearly independent vectors in the null space of $\underline{A}_2$.

From the foregoing discussion, if m<n, system (1.11.1) admits an infinity of solutions. However, among those infinitely many solutions, there is exactly one whose Euclidean norm is a minimum. That "optimal" solution is found next, via a quadratic programming problem, namely,

$$\text{Min}\phi(\underline{x})=\underline{x}^T\underline{x} \tag{1.11.15}$$

subject to

$$\underline{A}\underline{x}=\underline{b} \tag{1.11.16}$$

Applying the Lagrange multiplier technique [1.8], let $\underline{\lambda}$ be an m-dimensional vector whose components are called Lagrange multipliers. Define, then, the new quadratic form

$$\psi(\underline{x})=\underline{x}^T\underline{x}+\lambda^T(\underline{A}\underline{x}-\underline{b}) \tag{1.11.17}$$

which reduces to the original one (1.11.15), when (1.11.16) is satisfied. $\psi(\underline{x})$ has an extremum where its gradient $\psi'(\underline{x})$ vanishes. This condition is

$$\psi'(\underline{x})=2\underline{x}+\underline{A}^T\underline{\lambda}=\underline{0} \tag{1.11.18}$$

from which

$$\underline{x}=-\frac{1}{2}\underline{A}^T\underline{\lambda} \tag{1.11.19}$$

However, $\underline{\lambda}$ is yet unknown. Substituting the values of $\underline{x}$ given in (1.11.19). in (1.11.16), one obtains

$$-\frac{1}{2}\underline{A}\underline{A}^T\underline{\lambda}=\underline{b} \tag{1.11.20}$$

From which, if AA^T is of full rank,

$$\lambda=-2(AA^T)^{-1}b \qquad (1.11.21)$$

Finally, substituting the latter value of λ into eq. (1.11.19),

$$x=A^T(AA^T)^{-1}b=A^+b \qquad (1.11.22)$$

where

$$A^+\equiv A^T(AA^T)^{-1}$$

is another pseudo-inverse of A.

Exercise 1.11.1 Can both pseudo-inverses of A, the one given in (1.11.8) and that of (1.11.23) exist for a given matrix A? Explain.

The foregoing solution (1.11.22) has many interpretations: in control theory it yields the control taking a system from a known initial state to a desired final one while spending the minimum amount of energy. In Kinematics it finds two interpretations which will be given in Ch. 2, togehter with applications to hypoid gear design.

Exercise 1.11.2 Show that the image of the error (1.11.4) is perpendicular to x_0 as given by (1.11.8). This result is known as the "Projection Theorem" and finds extensive applications in optimization theory [1.9].

iii) m=n. This is the best known case and an extensive discussion of it can be found in any elementary linear algebra textbook. The most important result in this case states that if A is of full rank, i.e. if det $A \neq 0$, then the system has a unique solution, which is given by

$$x=A^{-1}b$$

1.12 NUMERICAL SOLUTION OF LINEAR ALGEBRAIC SYSTEMS

Consider the system (1.11.1) for all three cases discussed in section 1.11.

i) m=n. The first case that will be discussed here is that for m=n.

There are many methods to solve such a linear algebraic system, but all of them fall into one of two categories, namely, a) direct methods and b) iterative methods. Because the first ones are more suitable to be applied in nonlinear algebraic systems, which will be discussed in section 1.13, only direct methods will be treated here. There is an extensive literature dealing with interative methods, of which the treatise by Varga (1.10) discusses the topic very extensively.

As to direct methods, Gauss'algorithm is the one which has received most attention (1.11), (1.12). In (1.11) the LU decomposition algorithm is presented and, with further refinements, in (1.12). The solution is obtained in two steps:

In the first step the matrix of the system, $\underset{\sim}{A}$, is factored into the product of a lower triangular matrix, $\underset{\sim}{L}$, times an upper triangular one $\underset{\sim}{U}$, in the form

$$\underset{\sim}{A} = \underset{\sim}{L}\underset{\sim}{U} \qquad (1.12.1)$$

where the diagonal of $\underset{\sim}{L}$ contains ones in all its entries. Matrix $\underset{\sim}{U}$ contains the <u>singular</u> values of $\underset{\sim}{A}$ on its diagonal, and all its elements below the main diagonal are zero. The singular values of a matrix $\underset{\sim}{A}$ are the nonnegative square roots of the eigenvalues of $\underset{\sim}{A}^T\underset{\sim}{A}$. These are real and nonnegative, which is not difficult to prove.

<u>Exercise 1.12.1</u> Show that if $\underset{\sim}{A}$ is a nonsingular nxn matrix, $\underset{\sim}{A}^T\underset{\sim}{A}$ is positive definite, and if it is singular, then $\underset{\sim}{A}^T\underset{\sim}{A}$ is positive semi-definite. (Hint: Compute the norm of $\underset{\sim}{A}\underset{\sim}{x}$, for arbitraty $\underset{\sim}{x}$).

The $\underset{\sim}{L}\underset{\sim}{U}$ decomposition of $\underset{\sim}{A}$ is performed via the DECOMP subprogram appearing in (1.12). If $\underset{\sim}{A}$ happens to be singular, DECOMP detects this by computing det $\underset{\sim}{A}$, which is done performing the product of the singular values of $\underset{\sim}{A}$, and if this product turns out to be zero, sends a massage to the user thereby warning him that he cannot proceed any further.

If A is not singular, the user calls the SOLVE subprogram, which computes the solution to the system by back substitution, i.e. from (1.12.1) in the following manner: The equation

$$LUx=b \tag{1.12.2}$$

can be written as

$$Ly=b$$

by setting $Ux=y$. Thus

$$y=L^{-1}b=c \tag{1.12.3}$$

where L^{-1} exists since det L (the product of the elements on the diagonal of L) is equal to one (1.11). Substituting (1.12.3) into $Ux=y$, one obtains the final solution:

$$x=U^{-1}c$$

where U^{-1} exists because A has been detected to be nonsingular*. The flow diagram of the whole program appears in Fig 1.12.1 and the listings of DECOMP and SOLVE in Figs. 1.12.2 and 1.12.3

ii) $m>n$. Next, the numerical solution of the overdetermined linear system $Ax=b$ is discussed. In this case the number of equations is greater than that of unknowns and hence the sought "solution" is that x_0 which minimizes the Euclidean norm of the error Ax_0-b. This is done by application of Householder reflections (1.5) to both A and b. A Householder reflection is an orthogonal transformation H which has the property that

$$H^{-1}=H^T=H \tag{1.12.4}$$

Given an m-vector a with components $a_1, a_2, \ldots, a_m$, the Householder reflection H (a function of a) defined as

* In fact, there is no need to explicitely compute L^{-1} and U^{-1}, for the triangular structure of L and U permits a recursive solution.

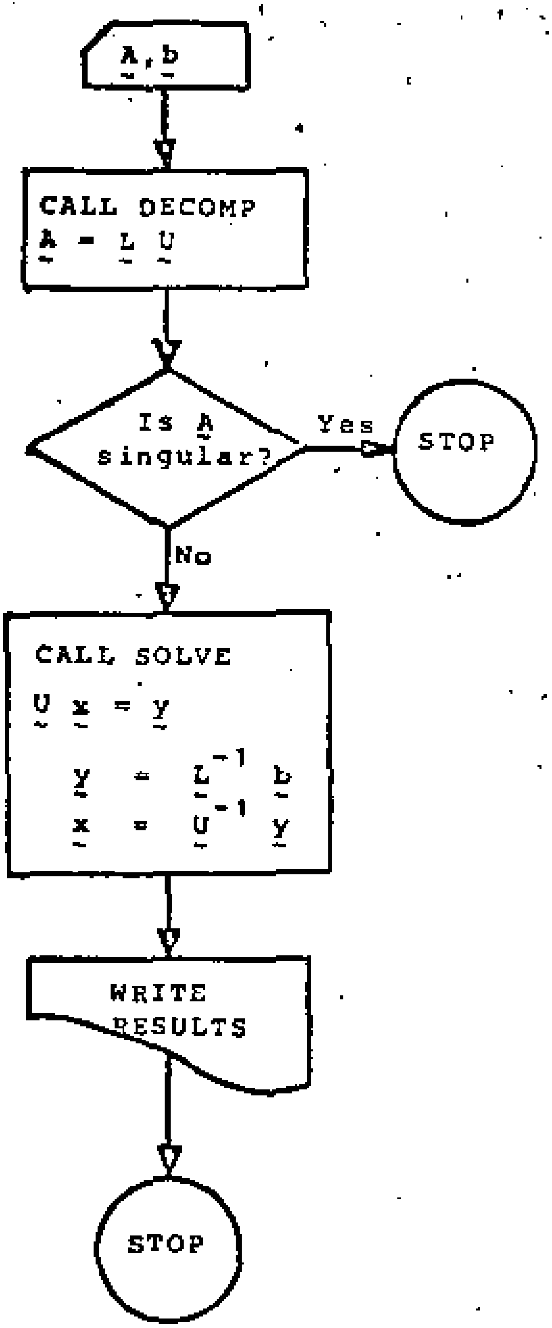
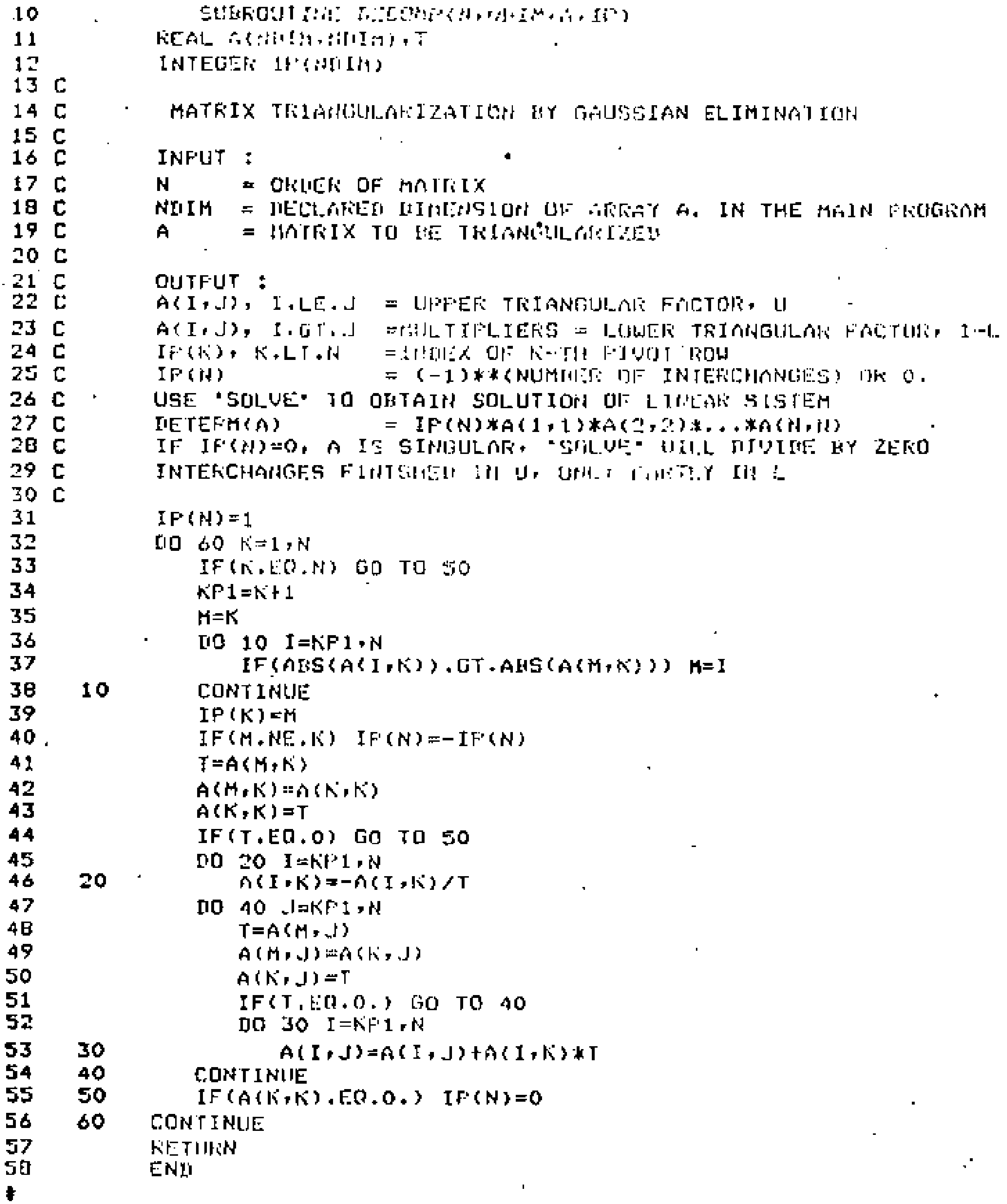
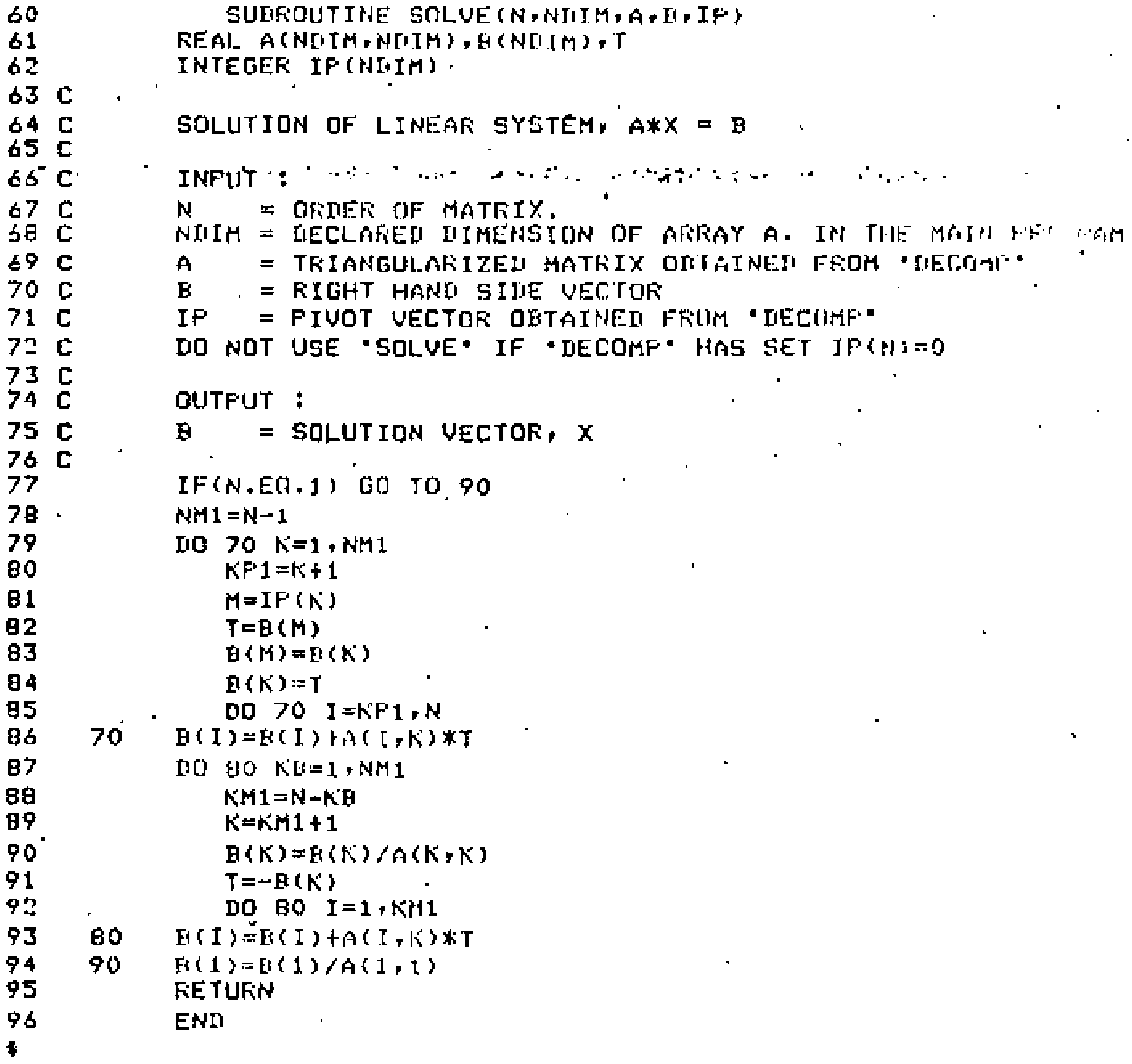


Fig 1.12.1 Flow diagram for the direct solution of a linear algebrai system with equal number of equations as unknowns.



```
           SUBROUTINE DECOMP(N,NDIM,A,IP)
          REAL A(NDIM,NDIM),T
          INTEGER IP(NDIM)
   C
   C       MATRIX TRIANGULARIZATION BY GAUSSIAN ELIMINATION
   C
   C      INPUT :
   C      N      = ORDER OF MATRIX
   C      NDIM   = DECLARED DIMENSION OF ARRAY A. IN THE MAIN PROGRAM
   C      A      = MATRIX TO BE TRIANGULARIZED
   C
   C      OUTPUT :
   C      A(I,J), I.LE.J  = UPPER TRIANGULAR FACTOR, U
   C      A(I,J), I.GT.J  =MULTIPLIERS = LOWER TRIANGULAR FACTOR, I-L
   C      IP(K), K.LT.N   =INDEX OF K-TH PIVOT ROW
   C      IP(N)           = (-1)**(NUMBER OF INTERCHANGES) OR 0.
   C      USE 'SOLVE' TO OBTAIN SOLUTION OF LINEAR SISTEM
   C      DETERM(A)       = IP(N)*A(1,1)*A(2,2)*...*A(N,N)
   C      IF IP(N)=0, A IS SINGULAR, 'SOLVE' WILL DIVIDE BY ZERO
   C      INTERCHANGES FINISHED IN U, ONLY PARTLY IN L
   C
          IP(N)=1
          DO 60 K=1,N
             IF(K.EQ.N) GO TO 50
             KP1=K+1
             M=K
             DO 10 I=KP1,N
                IF(ABS(A(I,K)).GT.ABS(A(M,K))) M=I
     10      CONTINUE
             IP(K)=M
             IF(M.NE.K) IP(N)=-IP(N)
             T=A(M,K)
             A(M,K)=A(K,K)
             A(K,K)=T
             IF(T.EQ.0) GO TO 50
             DO 20 I=KP1,N
     20         A(I,K)=-A(I,K)/T
             DO 40 J=KP1,N
                T=A(M,J)
                A(M,J)=A(K,J)
                A(K,J)=T
                IF(T.EQ.0.) GO TO 40
                DO 30 I=KP1,N
     30            A(I,J)=A(I,J)+A(I,K)*T
     40      CONTINUE
     50      IF(A(K,K).EQ.0.) IP(N)=0
     60   CONTINUE
          RETURN
          END
```

Fig. 1.12.2 Listing of SUBROUTINE DECOMP



```
           SUBROUTINE SOLVE(N,NDIM,A,B,IP)
        REAL A(NDIM,NDIM),B(NDIM),T
        INTEGER IP(NDIM)
C
C       SOLUTION OF LINEAR SYSTEM, A*X = B
C
C       INPUT :
C       N    = ORDER OF MATRIX.
C       NDIM = DECLARED DIMENSION OF ARRAY A. IN THE MAIN PROGRAM
C       A    = TRIANGULARIZED MATRIX OBTAINED FROM 'DECOMP'
C       B    = RIGHT HAND SIDE VECTOR
C       IP   = PIVOT VECTOR OBTAINED FROM 'DECOMP'
C       DO NOT USE 'SOLVE' IF 'DECOMP' HAS SET IP(N)=0
C
C       OUTPUT :
C       B    = SOLUTION VECTOR, X
C
        IF(N.EQ.1) GO TO 90
        NM1=N-1
        DO 70 K=1,NM1
           KP1=K+1
           M=IP(K)
           T=B(M)
           B(M)=B(K)
           B(K)=T
           DO 70 I=KP1,N
   70   B(I)=B(I)+A(I,K)*T
        DO 80 KB=1,NM1
           KM1=N-KB
           K=KM1+1
           B(K)=B(K)/A(K,K)
           T=-B(K)
           DO 80 I=1,KM1
   80   B(I)=B(I)+A(I,K)*T
   90   B(1)=B(1)/A(1,1)
        RETURN
        END
$
```

Fig. 1.12.3 Listing of SUBROUTINE SOLVE

$$\alpha = \text{sgn}\ (a_1)\,||\underset{\sim}{a}|| \qquad (1.12.5a)$$

$$\underset{\sim}{u} = \underset{\sim}{a}+\alpha\underset{\sim}{e}_1 \qquad (1.12.5b)$$

$$\beta = \alpha u_1 \qquad (1.12.5c)$$

$$\underset{\sim}{H} = \underset{\sim}{I}- \frac{1}{\beta}\underset{\sim}{u}\underset{\sim}{u}^T \qquad (1.12.5b)$$

Transforms $\underset{\sim}{a}$ into $-\alpha\underset{\sim}{e}_1$, and reflects any other vector $\underset{\sim}{b}$ about a hyperplane perpendicular to $\underset{\sim}{u}$.

On the other hand, if $\underset{\sim}{H}_k$ is defined as

$$\alpha_k = \text{sgn}(a_k)(a_k^2 + a_{k+1}^2+\ldots+a_m^2)^{\frac{1}{2}} \qquad (1.12.6a)$$

$$\underset{\sim}{u}_k = \left(0,\ldots,0,a_k+\alpha_k,\ a_{k+1},\ldots,a_m\right)^T \qquad (1.12.6b)$$

$$\beta_k = \alpha_k(\underset{\sim}{u}_k)_k \qquad (1.12.6c)$$

$$\underset{\sim}{H}_k = \underset{\sim}{I} - \frac{1}{\beta}\underset{\sim}{u}_k\underset{\sim}{u}_k^T \qquad (1.12.6d)$$

then $\underset{\sim}{H}_k\underset{\sim}{a}$ is a vector whose first k-1 components are identical to those of $\underset{\sim}{a}$, its kth component is $-\alpha_k$ and its remaining m-k components are all zero. Furthermore, if v is any other vector, then

$$\underset{\sim}{H}_k\underset{\sim}{v} = \underset{\sim}{v} - \gamma\underset{\sim}{u}$$

where

$$\gamma = \frac{\underset{\sim}{v}^T\underset{\sim}{v}}{\beta}$$

and if, in particular, $v_k = v_{k+1} = \ldots = v_m = 0$, then

$$\underset{\sim}{H}_k\underset{\sim}{v} = \underset{\sim}{v}$$

Let now $\underset{\sim}{H}_i$ be the Householder reflection wich cancels the last m-i components of the ith column of $\underset{\sim}{H}_{i-1}\underset{\sim}{A}$, while leaving its i-1 components unchanged and setting its ith component equal to $-\alpha_i$, for i=1,...,n. By application of the n Householder reflections thus defined, on $\underset{\sim}{A}$ and $\underset{\sim}{b}$ in the form

$$\underset{\sim}{H}_n\underset{\sim}{H}_{n-1}\cdots\underset{\sim}{H}_2\underset{\sim}{H}_1\underset{\sim}{A}\underset{\sim}{x}_0=\underset{\sim}{H}_n\underset{\sim}{H}_{n-1}\cdots\underset{\sim}{H}_2\underset{\sim}{H}_1\underset{\sim}{b} \qquad (1.12.7)$$

the original system is transformed into the following two systems

$$\underline{A}'_1\underline{x}_0 = \underline{b}'_1$$

$$\underline{A}'_2\underline{x}_0 = \underline{b}'_2$$

where $\underline{A}'_1$ is nxn and upper triangular, whereas $\underline{A}'_2$ is the (m-n)xn zero matrix and $\underline{b}'_2$ is of dimension m-n and diferent from zero. Once the system is in upper triangular form, it is a simple matter to find the values of the components of $\underline{x}_0$ by back substitution. Let a^*_{ij} and b^*_k be the values of the (i, j) element of $\underline{A}'_1$ and the k$\underline{\text{th}}$ component of $\underline{b}'_1$ respectively. Then, starting from the n$\underline{\text{th}}$ equation of system (1.12.7),

$$a^*_{nn}x_n = b^*_n$$

x_n is obtained as

$$x_n = \frac{b^*_n}{a^*_{nn}}$$

Substituting this value into the (n-1) st equation,

$$a^*_{n-1,n-1}x_{n-1} + a^*_{n-1,n}\frac{b^*_n}{a^*_{nn}} = b^*_{n-1}$$

from which

$$x_{n-1} = \frac{b^*_{n-1}}{a^*_{n-1,n-1}} \cdot \frac{b^*_n}{a^*_{nn}}$$

Proceeding similarly with the (n-2)nd,...,2nd and 1st equations, the n components of $\underline{x}_0$ are found. Clearly, then, $\underline{b}'_2$ is the error in the approximation and $||\underline{b}'_2|| = ||\underline{A}\ \underline{x} - \underline{b}||$.

The foregoing Householder reflection method can be readily implemented in a digital computer via the HECOMP and HOLVE subroutines appearing in (1.14), whose listings are reproduced in Figs 1.12.4 and 1.12.5.

Exercise 1.12.2 Show that, for any n-vector $\underline{x}$

$$\det(\underline{I}+\underline{x}\underline{x}^T) = 1+\underline{x}^T\underline{x}$$

```
       SUBROUTINE HECOMP(MDIM,M,N,A,U)
       INTEGER MDIM,M,N
       REAL A(MDIM,N),U(M)
       REAL ALPHA,BETA,GAMMA,SQRT
C
C   HOUSEHOLDER REDUCTION OF RECTANGULAR MATRIX TO UPPER
C   TRIANGULAR FORM.  USE WITH HOLVE FOR LEAST-SQUARE
C   SOLUTIONS OF OVERDETERMINED SYSTEMS.
C
C   MDIM= DECLARED ROW DIMENSION OF A
C   M   = NUMBER OF ROWS OF A
C   N   = NUMBER OF COLUMNS OF A
C   A   = M-BY-N MATRIX WITH M.>.N
C          INPUT :
C                 MATRIX TO BE REDUCED
C          OUTPUT:
C                 REDUCED MATRIX AND INFORMATION ABOUT REDUCTION
C   U   = M-VECTOR
C          INPUT :
C                 IGNORED
C          OUTPUT:
C                 INFORMATION ABOUT REDUCTION
C
C          FIND REFLECTION WHICH ZEROES A(I,K), I= K+1,...,M.
C
      DO 6 K= 1,N
         ALPHA= 0.0
         DO 1 I= K,M
            U(I)= A(I,K)
            ALPHA= ALPHA+U(I)*U(I)
 1       CONTINUE
         ALPHA= SQRT(ALPHA)
         IF(U(K).LT.0.0) ALPHA= -ALPHA
         U(K)= U(K)+ALPHA
         BETA= ALPHA*U(K)
         A(K,K)= -ALPHA
         IF(BETA.EQ.0.0.OR.K.EQ.N) GO TO 6
C
C   APPLY REFLECTION TO REMAINING COLUMNS OF A
         KP1= K+1
         DO 4 J= KP1,N
            GAMMA= 0.0
            DO 2 I= K,M
               GAMMA= GAMMA+U(I)*A(I,J)
 2          CONTINUE
            GAMMA= GAMMA/BETA
            DO 3 I= K,M
               A(I,J)= A(I,J)-GAMMA*U(I)
 3          CONTINUE
 4       CONTINUE
 6    CONTINUE
      RETURN
C
C   TRIANGULAR RESULT STORED IN A(I,J), I.LE.J
C   VECTORS DEFINING REFLECTIONS STORED IN U AND REST OF A
      END
```

Fig 1.12.4 Listing of SUBROUTINE HECOMP

```
          SUBROUTINE HOLVE(MDIM,M,N,A,U,B)
          INTEGER MDIM,M,N
          REAL A(MDIM,N),U(M),B(M)
          REAL BETA,GAMMA,T
C
C
C   LEAST-SQUARE SOLUTION OF OVERDETERMINED SYSTEMS
C   FIND X THAT MINIMIZES NORM(A*X- B)
C
C   MDIM,M,N,A,U.  RESULTS FROM HECOMP
C   B= M-VECTOR
C      INPUT :
C              RIGHT HAND SIDE
C      OUTPUT:
C              FIRST N COMPONENTS = THE SOLUTION, X
C              LAST M-N COMPONENTS= TRANSFORMED RESIDUAL
C   DIVISION BY ZERO IMPLIES A NOT OF FULL RANK
C
C   APPLY REFLECTIONS TO B
C
C
        DO 3 K= 1,N
           T= A(K,K)
           BETA= -U(K)*A(K,K)
           A(K,K)= U(K)
           GAMMA= 0.0
           DO 1 I= K,M
              GAMMA= GAMMA+A(I,K)*B(I)
 1         CONTINUE
           GAMMA= GAMMA/BETA
           DO 2 I= K,M
              B(I)= B(I)-GAMMA*A(I,K)
 2         CONTINUE
           A(K,K)= T
 3      CONTINUE
C
C   BACK SUBSTITUTION
C
        DO 5 KB= 1,N
           K= N+1-KB
           B(K)= B(K)/A(K,K)
           IF(K.EQ.1) GO TO 5
           KM1= K-1
           DO 4 I= 1,KM1
              B(I)= B(I)-A(I,K)*B(K)
 4         CONTINUE
 5      CONTINUE
        RETURN
        END
```

Fig 1.12.5 Listing of SUBROUTINE HOLVE

Exercise 1.12.3* Show that $\underset{\sim}{H}$, as defined in eqs. (1.12.5) is in fact a reflection, i.e. show that $\underset{\sim}{H}$ is orthogonal and the value of its determinant is -1. (Hint: Use the result of Exercise 1.12.2).

ii) m<n: Now, the linear system of equations $\underset{\sim}{A}\underset{\sim}{x}=\underset{\sim}{b}$ is studied when the number of unknowns is greater that the number of equations.

In this case, the system is underdetermined and has an infinity of solutions. However, as was discussed in Section 1.11, among those solutions, there is one, say $\underset{\sim}{x}_0$ whose Euclidean norm is a minimum. This is given by eq. (1.11.22), repeated here for ready reference.

$$\underset{\sim}{x}_0=\underset{\sim}{A}^T(\underset{\sim}{A}\underset{\sim}{A}^T)^{-1}\underset{\sim}{b} \tag{1.12.8}$$

One possible way of computing $\underset{\sim}{x}_0$ is given next:

a) Write eq. (1.11.20) in the form

$$\underset{\sim}{A}\underset{\sim}{A}^T\underset{\sim}{\lambda}=\underset{\sim}{b} \tag{1.12.9}$$

b) Using the LU decomposition method, find $\underset{\sim}{\lambda}$ from (1.12.9)

c) With $\underset{\sim}{\lambda}$ known from step ii), compute $\underset{\sim}{x}_0$ by matrix multiplication, as appearing in (1.11.19), i.e.

$$\underset{\sim}{x}_0=\underset{\sim}{A}^T\underset{\sim}{\lambda} \tag{1.12.10}$$

1.13 NUMERICAL SOLUTION OF NONLINEAR ALGEBRAIC SYSTEMS.

For several reasons, nonlinear systems are more difficult to deal with than are linear systems. Considering the simplest case of equal number of equations and unknowns, there is no guarantee that the nonlinear system has a uinque solution; in fact, there is no guarantee that the system has a solution at all.

* See Section 2.3 for more details on reflections.

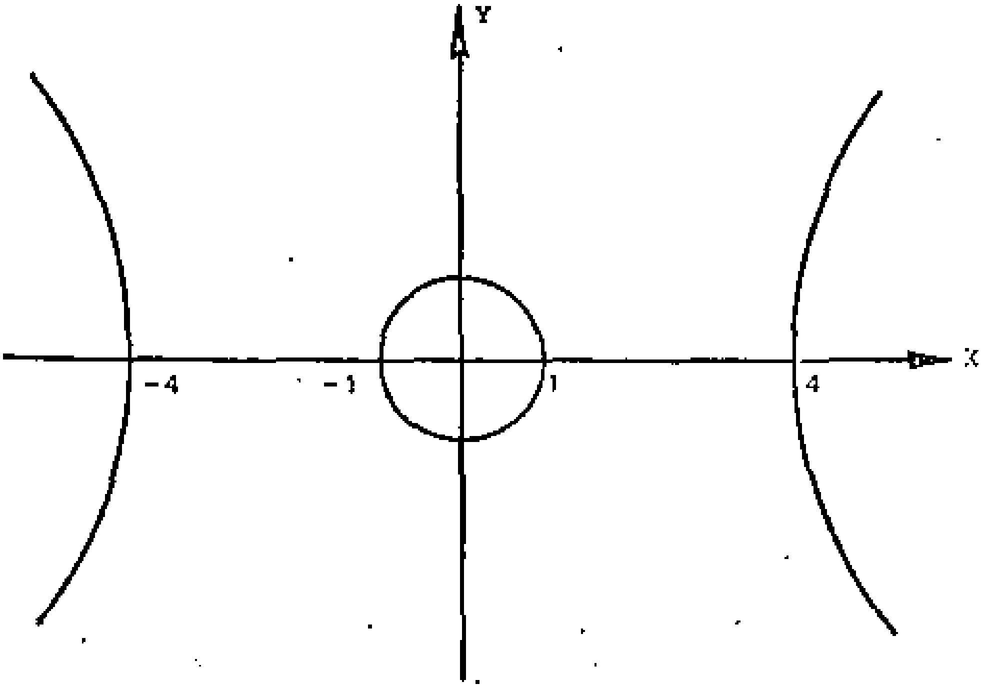


Fig 1.13.1 Non-intersecting hyperbola and circle

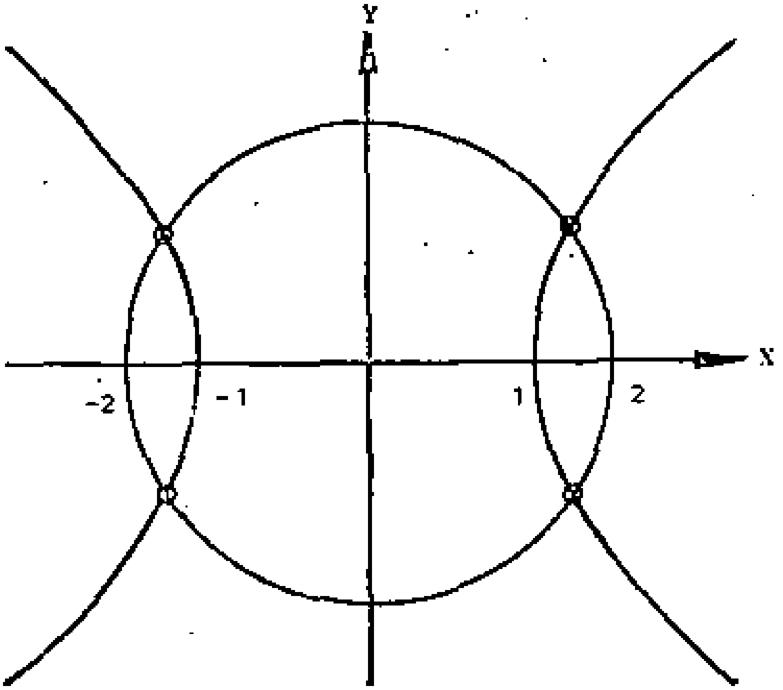


Fig 1.13.2 Intersections of a hyperbola and a circle

Example 1.13.1 The 2nd order nonlinear algebraic system

$$x^2 - y^2 = 16 \qquad \text{(a)}$$

$$x^2 + y^2 = 1 \qquad \text{(b)}$$

has no solution, for the hyperbola (a) does not intersect the circle (b), as is shown in Fig. 1.13.1

Example 1.13.2 The 2nd order linear algebraic system

$$x^2 - y^2 = 1 \qquad \text{(c)}$$

$$x^2 + y^2 = 4 \qquad \text{(d)}$$

has four solutions, namely

$$x_1 = \sqrt{\frac{5}{2}}, \quad y_1 = \sqrt{\frac{3}{2}}$$

$$x_2 = \sqrt{\frac{5}{2}}, \quad y_2 = -\sqrt{\frac{3}{2}}$$

$$x_3 = -\sqrt{\frac{5}{2}}, \quad y_3 = -\sqrt{\frac{3}{2}}$$

$$x_4 = -\sqrt{\frac{5}{2}}, \quad y_4 = \sqrt{\frac{3}{2}}$$

which are the four points where the hyperbola (c) intersects the circle (d). These intersections appear in Fig. 1.13.2

The most popular method of solving a nonlinear algebraic system is the so-called Newton-Raphson method. First, the system of equations has to be written in the form

$$\underline{f}(\underline{x}) = \underline{0} \qquad (1.13.1)$$

where $\underline{f}$ and $\underline{x}$ are m- and n- dimensional vectors. For example, system (a), (b) of Example 1.13.1 can be written in the form

$$f_1(x_1,x_2) = x_1^2 - x_2^2 - 16 = 0 \qquad \text{(a')}$$

$$f_2(x_1,x_2) = x_1^2 + x_2^2 - 1 = 0 \qquad \text{(b')}$$

Here f_1 and f_2 are the components of the 2-dimensional vectors $\underline{f}$ and x_1 and

x_2 (clearly, x and y have been replaced by x_1 and x_2, respectively) are the components of the 2-dimensional vector $\underline{x}$. Next, the three cases, m=n, m>n and<m n, are discussed

<u>First case: m=n</u>

Let $\underline{x}_0$ be known to be a "good" approximation to the solutions $\underline{x}_r$ or a "guess". The expansion of $\underline{f}(\underline{x})$ about $\underline{x}_0$ in a Taylor series yields

$$\underline{f}(\underline{x}_0+\Delta\underline{x}) = \underline{f}(\underline{x}_0) + \underline{f}'(\underline{x}_0)\Delta\underline{x} + \underline{R} \qquad (1.13.1)$$

If $\underline{x}_0 + \Delta\underline{x}$ is an even better approximation to $\underline{x}_r$, then $\Delta\underline{x}$ must be small and so, only linear terms could be retained in (1.13.2) and, of course, $\underline{f}(\underline{x}_0+\Delta\underline{x})$ must be closer to $\underline{0}$ than is $\underline{f}(\underline{x}_0)$. Under these assumptions, $\underline{f}(\underline{x}_0+\Delta\underline{x})$ can be assumed to be zero and (1.13.2) leads to

$$\underline{f}(\underline{x}_0) + \underline{f}'(\underline{x}_0)\Delta\underline{x} = \underline{0} \qquad (1.13.3)$$

In the above equation $\underline{f}'(\underline{x}_0)$ is the value of the gradient of $\underline{f}(\underline{x})$, $\underline{f}'(\underline{x})$ at $\underline{x} = \underline{x}_0$. This gradient is an nxn matrix, $\underline{J}$, whose (k,ℓ) element is

$$J_{k\ell} = \frac{\partial f_k}{\partial x_\ell} \qquad (1.13.4)$$

If the Jacobian matrix $\underline{J}$ is nonsingular, it can be inverted to yield

$$\Delta\underline{x} = - \underline{J}^{-1}(\underline{x}_0)\underline{f}(\underline{x}_0) \qquad (1.13.5)$$

Of course, $\underline{J}$ need not actually be inverted, for $\Delta\underline{x}$ can be obtained via the LU decomposition method from eq. (1.13.3) written in the form

$$\underline{J}(\underline{x}_0)\Delta\underline{x}=-\underline{f}(\underline{x}_0) \qquad (1.13.6)$$

With the value of $\Delta\underline{x}$ thus obtained, the improved value of $\underline{x}$, is computed as

$$\underline{x}_1 = \underline{x}_0 + \Delta\underline{x}$$

In general, at the <u>kth</u> iteration, the new value $\underline{x}_{k+1}$ is computed from the formula

$$\underline{x}_{k+1} = \underline{x}_k - \underline{J}(\underline{x}_k)^{-1}\underline{f}(\underline{x}_k) \qquad (1.13.7)$$

which is the Newton-Raphson iterative scheme. The procedure is stopped when a convergence criterion is met. One possible criterion is that the norm of $\underline{f}(\underline{x}_k)$ reaches a value below certain prescribed tolerance, i.e.

$$||\underline{f}(\underline{x}_k)|| \leq \varepsilon \tag{1.13.8}$$

where ε is the said tolerance. On the other hand, it can also happen that at iteration k, the norm of the increment becomes smaller than the tolerance. In this case, even if the convergence criterion (1.13.8) is not met, it is useless to perform more interations. Thus, it is more reasonable to verify first that the norm of the correction does not become too small before proceeding further, and stop the procedure if both $||\underline{f}(\underline{x}_k)||$ and $||\Delta\underline{x}_k||$ are small enough, in which case, convergence is reached.

If only $||\Delta\underline{x}_k||$ goes below the imposed tolerance, do not accept the corresponding $\underline{x}_k$ as the solution. The conditions under which the procedure converges are discussed in (1.15). These conditions, however, cannot be verified easily, in general. What is advised to do is to try different initial guesses $\underline{x}_0$ till convergence is reached and to stop the procedure if either

i) too many iterations have been performed

or

ii) $||\Delta\underline{x}_k|| \leq \varepsilon$ but $||\underline{f}(\underline{x}_k)|| > \varepsilon$

If the method of Newton-Raphson converges for a given problem, it does so quadratically, i.e. two digits are gained per iteration during the aproximation to the solution. It can happen, however, that the proceiure does not converge monotonically, in which case,

$$||\underline{f}(\underline{x}_{k-1})|| > ||\underline{f}(\underline{x}_k)||$$

thus giving rise to strong oscillations and, possibly, divergence. One way to cope with this situation is to introduce damping, i.e. instead of using

the whole computed increment $\Delta \underset{\sim}{x}_k$, use a fraction of it, i.e. at the kth iteration, for i=0,1,..., max, instead of using formula (1.13.7) to compute the next value $\underset{\sim}{x}_{k+1}$, use

$$\underset{\sim}{x}_{k+1} = \underset{\sim}{x}_k - \alpha^i \underline{J}(\underset{\sim}{x}_k)^{-1}\underline{f}(\underset{\sim}{x}_k) \qquad (1.13.9)$$

where α is a real number between 0 and 1. For a given k, eq. (1.13.9) represents the damping part of the procedure, which is stopped when

$$||\underline{f}(\underset{\sim}{x}_{k+1})_i)|| < ||\underline{f}(\underset{\sim}{x}_k)||$$

The algorithm is summarized in the flow chart of Fig 1.13.3 and implemented in the subroutine NRDAMP appearing in Fig 1.13.4

Second case: m>n

In this case the system is overdetermined and it is not possible, in general, to satisfy all the equations. What can be done, however, is to find that $\underset{\sim}{x}_0$ which minimizes $||\underline{f}(\underline{x})||$.

This problem arises, for example, when one tries to design a planar four-bar linkage to guide a rigid body through more than five configurations

To find the minimizing $\underset{\sim}{x}_0$, define first which norm of $\underline{f}(\underline{x})$ is desired to minimize. One norm which has several advantages is the Euclidean norm, already discussed in case i of Section 1.11, where the linear least-square problem was discussed. In the context of nonlinear systems of equations, minimizing the quadratic norm of $\underline{f}(\underline{x})$ leads to the nonlinear least-square problem. The problem is then to find the minimum of the scalar function

$$\phi(\underline{x}) = ||\underline{f}(\underline{x})||^2 = \underline{f}^T(\underline{x})\underline{f}(\underline{x}) \qquad (1.13.10)$$

As already discussed in Section 1.10, for this function to reach a minimum, it must first reach a stationary point, i.e. its gradient must vanish. Thus,

$$\underline{\phi}'(\underline{x}) = 2\,\underline{J}^T(\underline{x})\underline{f}(\underline{x}) \qquad (1.13.11)$$

where $\underline{J}(\underline{x})$ is the Jacobian matrix of $\underline{f}$ with respect to $\underline{x}$, i.e. an mxn matrix

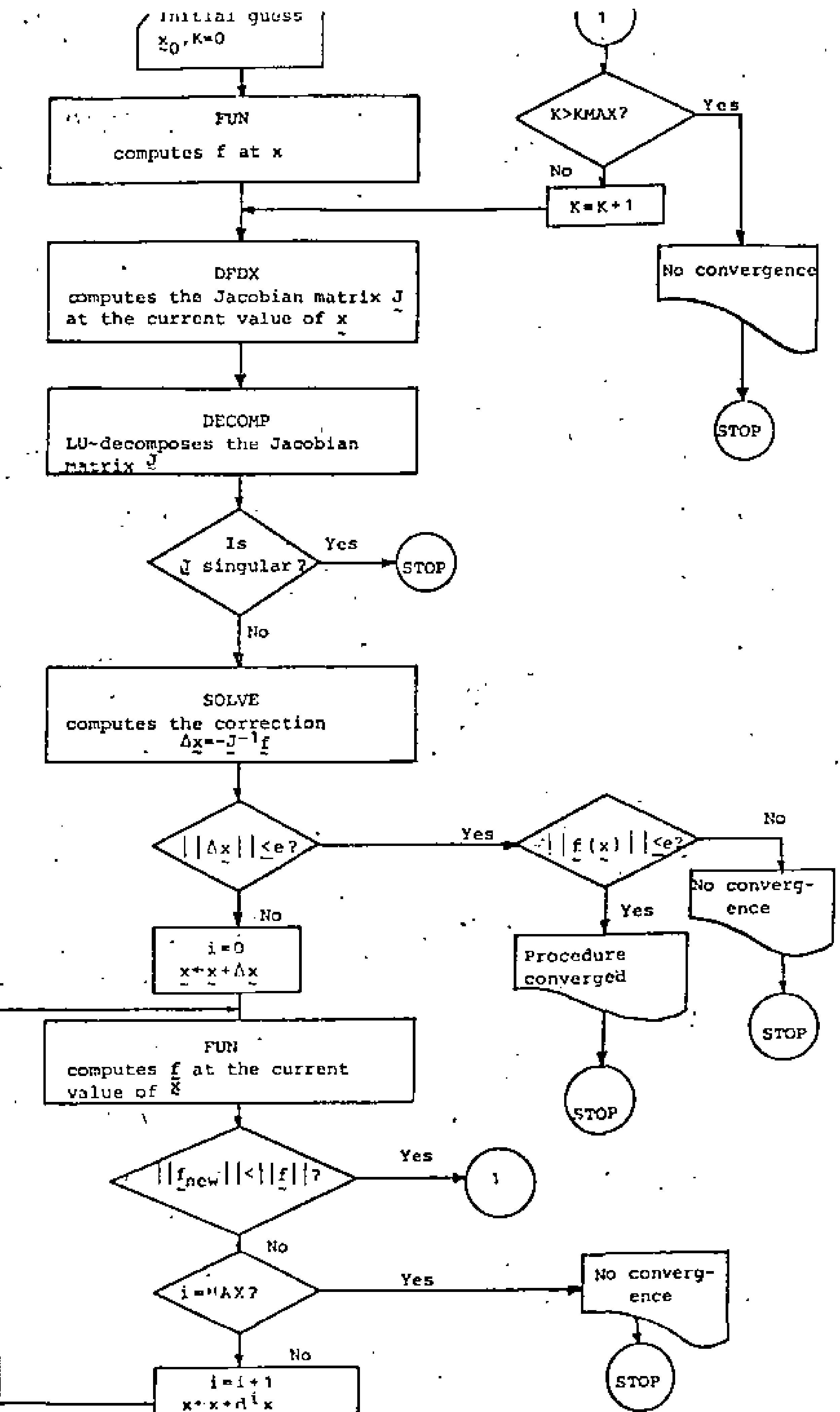


Fig. 1.13.3 Flow diagram to solve a nonlinear algebraic system with the same number of equations as of unknowns via the method of newton-Raphson with damping

```
       SUBROUTINE NRDAMP(X,FUN,DFDX,P,TOL,DAMP,N,ITER,MAX,KMAX)
        REAL X(N),F(N),DF(N,N),DELTA(N),P(20)
        INTEGER IP(N)
C
C
C   THIS SUBROUTINE FINDS THE ROOTS OF A NONLINEAR ALGEBRAIC SYSTEM OF
C   ORDER N, VIA NEWTON-RAPHSON METHOD(ISAACSON E. AND KELLER H.B.
C   ANALYSIS OF NUMERICAL METHODS, JOHN WILEY AND SONS, INC.-  NEW YORK,
C   1966,PP. 85-123) WITH DAMPING. SUBROUTINE PARAMETERS :
C   X            = N-VECTOR OF UNKNOWNS
C   FUN          = EXTERNAL SUBROUTINE WHICH COMPUTES VECTOR F,
C                  CONTAINING THE FUNCTIONS WHOSE ROOTS ARE TO BE OBTAINED
C   DFDX         = EXTERNAL SUBROUTINE WHICH COMPUTES THE JACOBIAN MATRIX
C                  OF VECTOR F WITH RESPECT TO X
C   P            = IS AN AUXILIARY VECTOR OF SUITABLE DIMENSION. IT
C                  CONTAINS THE PARAMETERS THAT EACH PROBLEM MAY REQUIRE
C   TOL          = POSITIVE SCALAR, TOLERANCE IMPOSED IN THE APPROXIMATION
C   DAMP         = THE DAMPING VALUE, PROVIDED BY THE USER SUCH THAT
C                  0.LT. DAMP .LE.1
C   ITER         = NUMBER OF ITERATION BEING EXECUTED
C   MAX          = MAXIMUM NUMBER OF ALLOWED ITERATIONS
C   KMAX         = MAXIMUM NUMBER OF ALLOWED DAMPINGS PER ITERATION. IT IS
C                  PROVIDED BY THE USER
C   FUN AND DFDX  ARE SUPPLIED BY THE USER
C   SUBROUTINES 'DECOMP' AND 'SOLVE' SOLVE THE NTH. ORDER LINEAR
C   ALGEBRAIC SYSTEM DF(X)*DELTA=F(X), DELTA BEING THE CORRECTION TO
C   THE K-TH ITERATION. THE METHOD USED IS THE LU DECOMPOSITION(MOLER
C   C.B. MATRIX COMPUTATIONS WITH FORTRAN AND PAGING. COMMUNICATIONS OF
C   THE A.C.M. VOLUME 15, NUMBER 4, APRIL 1972 ).
C
C
      ITER= 0
      CALL FUN(X,F,P,N)
  1   ITER= ITER+1
      IF(ITER.GT.MAX) GO TO 10
      FNOR1=FNORM(F,N)
      CALL DFDX(X,DF,P,N)
      CALL DECOMP(N,N,DF,IP)
C
C   IF THE JACOBIAN MATRIX IS SINGULAR, THE SUBROUTINE RETURNS TO THE
C   MAIN PROGRAM. OTHERWISE, IT PROCEEDS FURTHER.
      IF(IP(N).EQ.0) GO TO 11
      CALL SOLVE(N,N,DF,F,IP)
      DO 2 I=1,N
           DELTA(I)=F(I)
  2   CONTINUE
      DELNOR=FNORM(DELTA,N)
```

Fig 1.13.4 Listing of SUBROUTINE NRDAMP

```
      IF(DELNOR.LT.TOL) GO TO 8
      K=1
    3 DO 4 I=1,N
          X(I)=X(I)-DELTA(I)
    4 CONTINUE
      CALL FUN(X,F,P,N)
      FNOR2=FNORM(F,N)
C
C  TESTING THE NORM OF THE FUNCTION F AT CURRENT VALUE OF X. IF THIS
C  DOES NOT DECREASE,- THEN DAMPING IS INTRODUCED.
      IF(FNOR2.LT.TOL) GO TO 8
      IF(FNOR2.LT.FNOR1) GO TO 1
      IF(K.GT.KMAX) GO TO 7
      DO 6 I=1,N
          IF(K.GE.2) GO TO 5
          DELTA(I)=(DAMP-1.)*DELTA(I)
          GO TO 6
    5     DELTA(I)=DAMP*DELTA(I)
    6 CONTINUE
      K=K+1
      GO TO 3
    7 WRITE(6,101)
C
C  IT AT THIS ITERATION THE NORM OF THE FUNCTION CANNOT BE DECREASED
C  AFTER KMAX DAMPINGS, DAMP IS SET EQUAL TO -1 AND THE SUBROUTINE
C  RETURNS TO THE MAIN PROGRAM.
      DAMP=-1
      RETURN
    8 WRITE(6,102) FNOR2,ITER,K
      DO 9 I=1,N
          WRITE(6,103)I,X(I)
    9 CONTINUE
      RETURN
   10 WRITE(6,104)
      RETURN
   11 WRITE(6,105)
      DAMP=0
      RETURN
  101 FORMAT(10X,'NO CONVERGENCE WITH THIS DAMPING VALUE'/)
  102 FORMAT(2X,'CONVERGENCE REACHED. NORM OF THE FUNCTION',5X,
     -         6HFUN = F15.9//2X,'NUMBER OF ITERATIONS = ',I5,5X,
     -         'NUMBER OF DAMPINGS =',I3//5X,'THE SOLUTION IS:'/)
  103 FORMAT(5X,2HI=I6,10X,2HX=F10.4/)
  104 FORMAT(10X,'NO CONVERGENCE'/)
  105 FORMAT(10X,'JACOBI MATRIX IS SINGULAR'/)
      END
```

Fig 1.13.4 Listing of SUBROUTINE NRDAMP (Continued)

Exercise 1.13.1. Derive the expression (1.13.11)

In order to compute the value of $\underline{x}$ that zeroes the gradient (1.13.11) proceed iteratively, as next outlined. Expand $\underline{f}(\underline{x})$ around $\underline{x}_0$:

$$\underline{f}(\underline{x}_0+\Delta\underline{x}) = \underline{f}(\underline{x}_0) + \underline{f}'(\underline{x}_0)\Delta\underline{x} + \underline{R} \qquad (1.13.12)$$

If $\underline{x}_0+\Delta\underline{x}$ is a better approximation to the value that minimizes the Euclidean norm of $\underline{f}(\underline{x})$, and if in addition $||\Delta\underline{x}||$ is small enough, $\underline{R}$ can be neglected in eq. (1.13.12) and as trying to set the whole expression equal to zero, the following equation is obtained

$$\underline{f}'(\underline{x}_0)\Delta\underline{x} = -\underline{f}(\underline{x}_0)$$

or, denoting by $\underline{J}$ the Jacobian matrix $\underline{f}'(\underline{x})$,

$$\underline{J}(\underline{x}_0)\Delta\underline{x} = -\underline{f}(\underline{x}_0)$$

which is an overdetermined linear system. As discussed in Section 1.11, such a system has in general no solution, but a value of $\Delta\underline{x}$ can be computed which minimizes the quadratic norm of the error $J(\underline{x}_0)\Delta\underline{x} + \underline{f}(\underline{x}_0)$. This value is given by the expression (1.11.8) as

$$\Delta\underline{x} = -(\underline{J}^T\underline{J})^{-1}\underline{J}^T\underline{f}$$

In general, at the kth iteration, compute $\Delta\underline{x}_k$ as

$$\Delta\underline{x}_k = -\left(\underline{J}^T(\underline{x}_k)\underline{J}(\underline{x}_k)\right)^{-1}\underline{J}^T(\underline{x}_k)\underline{f}(\underline{x}_k) \qquad (1.13.13)$$

and stop the procedure when $||\Delta\underline{x}_k||$ becomes smaller than a prescribed tolerance, thus indicating that the procedure converged. In fact, if $\Delta\underline{x}_k$ vanishes, unless $(\underline{J}^T\underline{J})^{-1}$ becomes infinity, this means that $\underline{J}^T\underline{f}$ vanishes. But if this product vanishes, then from eq. (1.13.11), the gradient $\phi'(\underline{x})$ also vanishes, thus obtaining a stationary point of the quadratic norm of $\underline{f}(\underline{x})$.

In order to accelerate the convergence of the procedure, damping can also be introduced. This way, instead of computing $\Delta\underline{x}_k$ from eq. (1.13.13), compute it from

$$\Delta \underset{\sim}{x}_k = -\alpha^i \left(\underset{\sim}{J}^T(\underset{\sim}{x}_k)\underset{\sim}{J}(\underset{\sim}{x}_k)\right)^{-1}\underset{\sim}{J}^T(\underset{\sim}{x}_k)\underset{\sim}{f}(\underset{\sim}{x}_k) \qquad (1.13.14)$$

for i = 0, 1,..., max and stop the damping when

$$||\phi(\underset{\sim}{x}_{k+1,i})|| < ||\phi(\underset{\sim}{x}_{k,i})||$$

The algorithm is illustrated with the flow diagram of Fig 1.13.5 and implemented with the subroutine NRDAMC, appearing in Fig 1.13.6

Third case: m<n

The system, in this case, is underdeterminated and infinitely many solutions can be expected to exist. Out of these solutions, however, one can choose that with a minimum norm, thus converting the problem into a nonlinear quadratic programming problem, stated as

$$\text{Minimize } \underset{\sim}{x}^T\underset{\sim}{x} \qquad (1.13.15a)$$

$$\text{subject to } \underset{\sim}{f}(\underset{\sim}{x}) = \underset{\sim}{0} \qquad (1.13.15b)$$

One way to find the minimizing $\underset{\sim}{x}_0$, of problem (1.13.15) is via the method of Lagrange multipliers. Thus, define a new objective function

$$\phi(\underset{\sim}{x}) = \underset{\sim}{x}^T\underset{\sim}{x} + \underset{\sim}{\lambda}^T\underset{\sim}{f}(\underset{\sim}{x}) \qquad (1.13.16)$$

which is stationary at $\underset{\sim}{x}_0$ where its gradient vanishes. Thus,

$$\phi\,(\underset{\sim}{x}_0) = 2\underset{\sim}{x}_0 + \underset{\sim}{f}'^T(\underset{\sim}{x}_0)\underset{\sim}{\lambda} = \underset{\sim}{0} \qquad (1.13.17)$$

The systems of equations (1.13.15b) and (1.13.17) now represent a larger system of m+n equations (m in (1.13.10b) and n in (1.13.12)) in m+n unknowns (m components of $\underset{\sim}{\lambda}$ and n components of $\underset{\sim}{x}$). Hence, the problem now reduces to the first case and so can be solved by application of the subroutine NRDAMP.

Exercise 1.13.2 Let

$$f(\underset{\sim}{x}) = \frac{1}{2}x_1^2 + (x_1-x_2)^2 + (x_2-x_3)^2 + (x_3-x_4)^2 +$$
$$+\beta\left(\frac{7}{2}\cos x_1 + \frac{5}{2}\cos x_2 + \frac{3}{2}\cos x_3 + \frac{1}{2}\cos x_4\right)$$

be a scalar function of a vector argument $\underset{\sim}{x} = (x_1, x_2, x_3, x_4)^T$. Find its

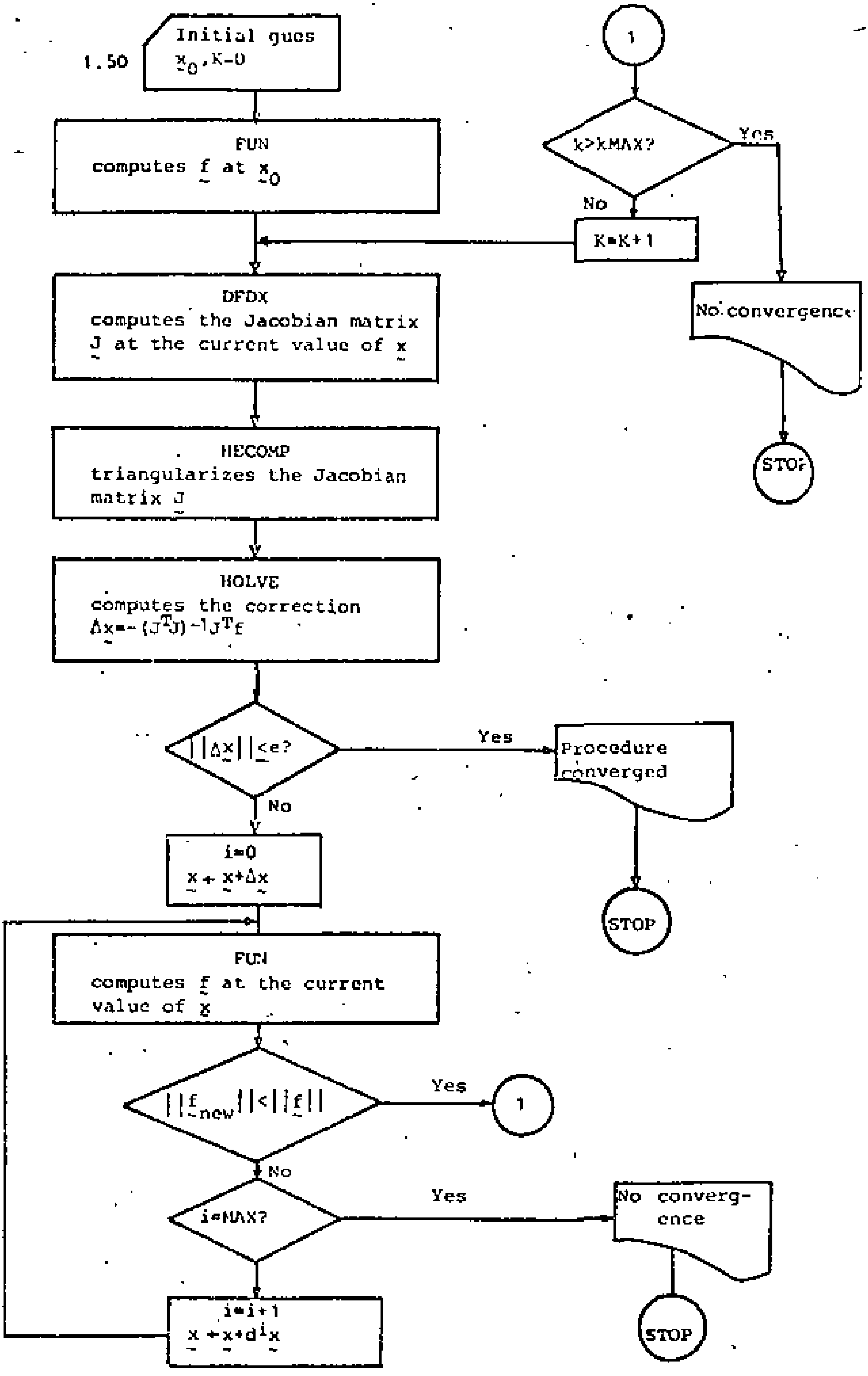
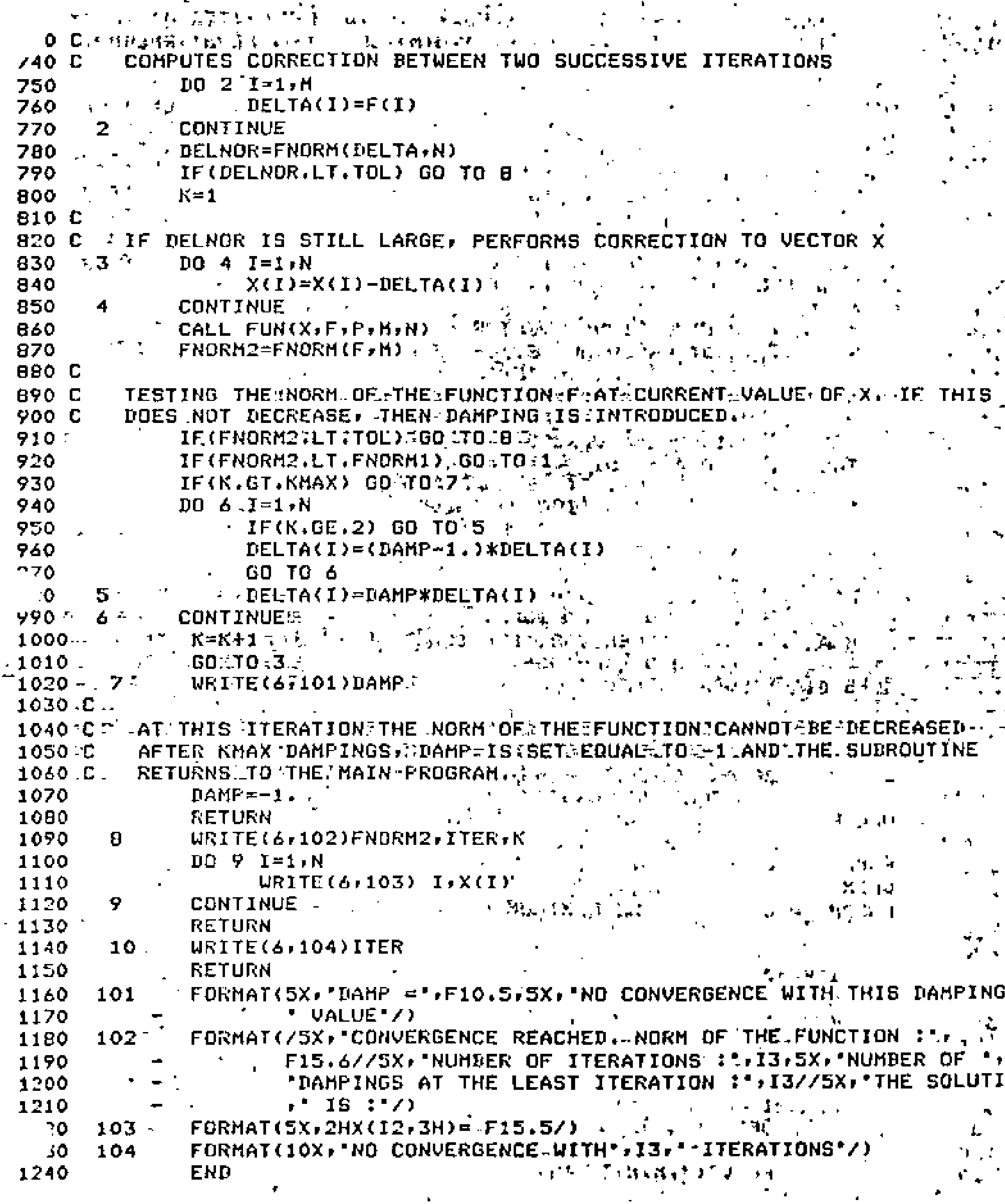


Fig 1.13.5 Flow diagram to compute the least-square solution to an overdetermined nonlinear algebraic system.

```
200             SUBROUTINE NRDAMC(X,FUN,DFDX,P,TOL,DAMP,N,M,ITER,MAX,KMAX)
210             REAL X(N),F(M),DF(M,N),P(1),U(M),DELTA(M),FNORM1,FNORM2,3
220      -      DELNOR
230 C
240 C   THIS PROGRAM OBTAINS THE LEAST SQUARE SOLUTION TO THE NONLINEAR
250 C   SYSTEM F(X)= 0, WHERE F AND X ARE M-AND N-DIMENSIONAL VECTORS, M
260 C   BEING GREATER THAN N. THE PROCEDURE IS ITERATIVE, AND AT EACH
270 C   ITERATION FINDS THE LEAST SQUARE SOLUTION TO THE LINEAR SYSTEM
280 C   DF*DELTA= -F, WHERE DF IS THE JACOBIAN M X N MATRIX OF THE ORIGINAL
290 C   SYSTEM, COMPUTED AT THE CURRENT VALUE OF X. THE LINEAR LEAST SQUARE
300 C   SOLUTION AT EACH ITERATION IS FOUND VIA HOUSEHOLDER REFLECTIONS (
310 C   BJOERCK A. AND G. DAHLQUIST, NUMERICAL METHODS, PRENTICE HALL,
320 C   ENGLEWOOD CLIFFS, N.J., 1974, PP. 201-206, 443-444).
330 C   PARAMETERS :
340 C       X     ..= N-DIMENSIONAL VECTOR OF UNKNOWNS.
350 C       F     ..= M-DIMENSIONAL VECTOR OF FUNCTIONS WHOSE EUCLIDEAN NORM
360 C                 IS TO BE MINIMIZED.
370 C       DF-   ..= M X N JACOBIAN MATRIX OF F WITH RESPECT TO X.
380 C       P     ..= VECTOR OF PARAMETERS APPEARING IN FUN AND/OR DFDX. ITS
390 C                 DIMENSION VARIES FROM PROBLEM TO PROBLEM.
400 C       TOL   ..= A REAL POSITIVE "SMALL" VARIABLE DENOTING THE IMPOSED
410 C                 TOLERANCE. IT IS SUPPLIED BY THE USER.
420 C       DAMP  ..= A REAL POSITIVE VARIABLE, 0.LT. DAMP .LT.1. IT DENOTES
430 C                 THE DAMPING FACTOR AND IS SUPPLIED BY THE USER.
440 C       ITER  ..= AN INTEGER VARIABLE DENOTING THE NUMBER OF THE CURRENT
450 C                 ITERATION.
460 C       MAX   ..= AN INTEGER VARIABLE DENOTING THE MAXIMUM NUMBER OF
470 C                 ALLOWED ITERATIONS.
480 C       KMAX  ..= AN INTEGER VARIABLE DENOTING THE MAXIMUM NUMBER OF
490 C                 ALLOWED DAMPINGS.
500 C   SUBSIDIARY SUBROUTINES :
510 C
520 C   HECOMP = TRIANGULARIZES A RECTANGULAR MATRIX BY HOUSEHOLDER
530 C            REFLECTIONS (MOLER C. B., MATRIX EIGENVALUE AND LEAST
540 C            SQUARE COMPUTATIONS, COMPUTER SCIENCE DEPARTAMENT,
550 C            STANFORD UNIVERSITY, MARCH, 1973)
560 C   HOLVE  = SOLVES TRIANGULARIZED SYSTEM BY BACK-SUBSTITUTION (MOLER
570 C            C. B., OP. CIT.)
580 C   FUN    = COMPUTES F.
590 C   DFDX   = COMPUTES DF.
600 C   FNORM  = COMPUTES THE MAXIMUM NORM OF A VECTOR.
610 C
620 C
630             ITER=0
640             CALL FUN(X,F,P,M,N)
650       1     ITER=ITER+1
660             IF(ITER.GT.MAX) GO TO 10
670 C
680 C   FORMS LINEAR LEAST SQUARE PROBLEM
690             FNORM1=FNORM(F,M)
700             CALL DFDX(X,DF,P,M,N)
710             CALL HECOMP(M,M,N,DF,U)
720             CALL HOLVE(M,M,N,DF,U,F)
```

Fig 1.13.6 Listing of SUBROUTINE NRDAMC



```
C
C    COMPUTES CORRECTION BETWEEN TWO SUCCESSIVE ITERATIONS
         DO 2 I=1,M
              DELTA(I)=F(I)
    2    CONTINUE
         DELNOR=FNORM(DELTA,N)
         IF(DELNOR.LT.TOL) GO TO 8
         K=1
C
C    IF DELNOR IS STILL LARGE, PERFORMS CORRECTION TO VECTOR X
    3    DO 4 I=1,N
              X(I)=X(I)-DELTA(I)
    4    CONTINUE
         CALL FUN(X,F,P,M,N)
         FNORM2=FNORM(F,M)
C
C    TESTING THE NORM OF THE FUNCTION F AT CURRENT VALUE OF X. IF THIS
C    DOES NOT DECREASE, THEN DAMPING IS INTRODUCED.
         IF(FNORM2.LT.TOL) GO TO 8
         IF(FNORM2.LT.FNORM1) GO TO 1
         IF(K.GT.KMAX) GO TO 7
         DO 6 I=1,N
              IF(K.GE.2) GO TO 5
              DELTA(I)=(DAMP-1.)*DELTA(I)
              GO TO 6
    5         DELTA(I)=DAMP*DELTA(I)
    6    CONTINUE
          K=K+1
          GO TO 3
    7     WRITE(6,101)DAMP
C
C     AT THIS ITERATION THE NORM OF THE FUNCTION CANNOT BE DECREASED
C      AFTER KMAX DAMPINGS, DAMP IS SET EQUAL TO -1 AND THE SUBROUTINE
C      RETURNS TO THE MAIN PROGRAM.
          DAMP=-1.
          RETURN
    8     WRITE(6,102)FNORM2,ITER,K
          DO 9 I=1,N
               WRITE(6,103) I,X(I)
    9     CONTINUE
          RETURN
   10     WRITE(6,104)ITER
          RETURN
  101     FORMAT(5X,'DAMP =',F10.5,5X,'NO CONVERGENCE WITH THIS DAMPING
     -               ' VALUE'/)
  102     FORMAT(/5X,'CONVERGENCE REACHED. NORM OF THE FUNCTION :',
     -               F15.6//5X,'NUMBER OF ITERATIONS :',I3,5X,'NUMBER OF ',
     -               'DAMPINGS AT THE LEAST ITERATION :',I3//5X,'THE SOLUTI
     -               ,' IS :'/)
  103     FORMAT(5X,2HX(I2,3H)= F15.5/)
  104     FORMAT(10X,'NO CONVERGENCE WITH',I3,' ITERATIONS'/)
          END
```

Fig 1.13.6 Listing of SUBROUTINE NRDAMC (Continued)

stationary points and decide whether each is either a maximum, a minimum or a saddle point, for β = 1,10,50.

Note: $f(\underline{x})$ could represent the potential energy of a mechanical system. In this case the stationary points correspond to the following equilib: ium states: minima yield a stable equilibrium state, whereas maxima and saddle points yield unstable states.

Example 1.13.3 Find the point closest to all three curves of Fig 1.13.7. These curves are the parabola(P), the circle(C) and the hyperbola(H) with the following equations:

$$y = \frac{1.}{2.4} x^2 \qquad (P)$$

$$x^2 + y^2 = 4 \qquad (C)$$

$$x^2 - y^2 = 1 \qquad (H)$$

From Fig 1.13.7 it is clear that no single pair (x,y) satisfies all three equations simultaneously. There exist points of coordinates x_0, y_0, however, that minimize the quadratic norm of the error of the said equations. These can be found with the aid of SUBROUTINE NRDAMC. A program was written that calls NRDAMC, HECOMP and HOLVE to find the least-square solution to eqs. (P), (C) and (H). The found solutions were:

First solution: x=-1.61537, y=1.17844

Second solution: x= 1,61537, y=1.17844

which are shown in Fig 1.13.7. These points have symmetrical locations, as expected, and lie almost on the circle at abount equal distances from A_i and C_i and B_i and D_i (i=1,2)

The maximum error of the foregoing approximation was computed as 0.22070

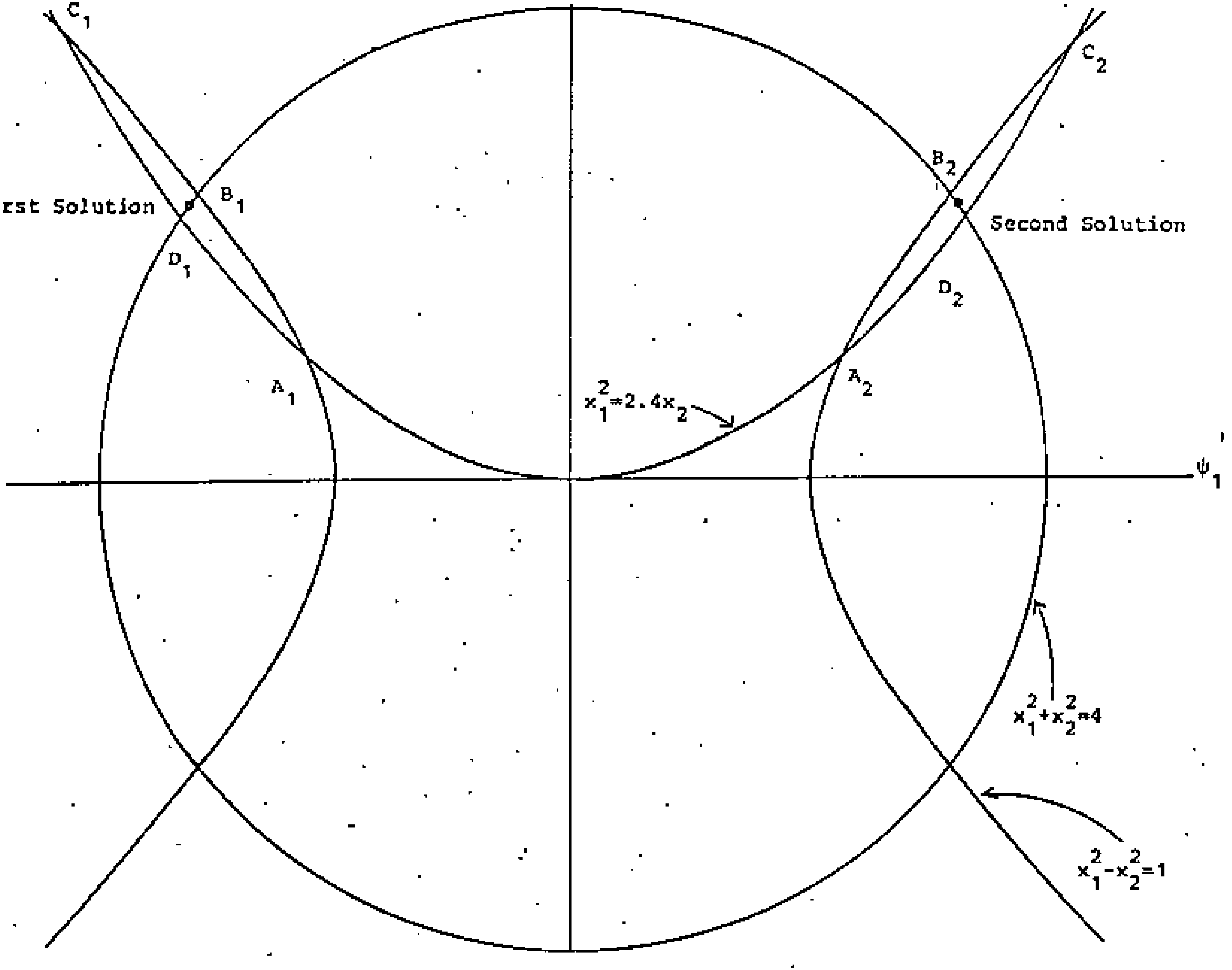


Fig 1.13.7 Location of the point closest to a parabola, a circle and a hyperbola.

REFERENCES

1.1 Lang S., Linear Algebra, Addison-Wesley Publishing Co., Menlo Park, 1970, pp. 39 and 40.

1.2 Lang S., op. cit., pp. 99 and 100

1.3 Finkbeiner, D.F., Matrices and Linear Transformations, W.H. Freeman and Company, San Francisco, 1960, pp. 139-142

1.4 Halmos, P.R., Finite-Dimensional Vector Spaces, Springer-Verlag, N. York, 1974.

1.5 Businger P. and G.H. Golub, "Linear Least Squares Solutions by Householder Transformations", in Wilkinson J.H. and C. Reinsch, eds., Handbook for Automatic Computation, Vol. II, Springer-Verlag, N. York, 1971, pp. 111-118

1.6 Stewart, G.W., Introduction to Matrix Computations, Academic Press, N.York, 1973, pp. 208-249.

1.7 Soderstrom T. and G.W. Stewart, "On the numerical properties of an iterative method for computing the Moore-Penrose generalized inverse", SIAM J. on Numerical Analysis, Vol. II, No. 1, March 1974.

1.8 Brand L., Advanced Calculus, John Wiley and Sons, Inc., N. York, 1955, pp. 147-197.

1.9 Luenberger, D.G., Optimization by Vector Space Methods, John Wiley and Sons, Inc., N. York, 1969, pp. 8, 49-52

1.10 Varga, R.S., Matrix Iterative Analysis, Prentice Hall, Inc., Englewood Cliffs, 1962, pp. 56-160

1.11 Forsythe, G.E. and C.B. Moler, Computer Solution of Linear Algebraic Systems, Prentice Hall, Inc., Englewood Cliffs, 1967, pp. 27-33

1.12 Moler C.B., "Algorithm 423. Linear Equation Solver (F 4)" Communications of the ACM, Vol. 15, Number 4, April 1973, p. 274.

1.13 Björck Å. and G. Dahlquist, Numerical Methods, Prentice-Hall, Inc., Englewood Cliffs, 1974, pp. 201-206.

1.14 Moler C.B., Matrix Eigenvalue and Least Square Computations, Computer Science Departament, Stanford University, Stanford, California, 1973 pp. 4.1-4.15

1.15 Isaacson, E. and H. B. Keller, Analysis of Numerical Methods, John Wiley and Sons, Inc., N. York, 1966, pp. 85-123

1.16 Angeles, J., "Optimal synthesis of linkages using Householder reflections", Proceedings of the Fifth World Congress on the Theory of Machines and Mechanisms, vol. I, Montreal, Canada, July 8-13, 1979, pp. 111-114.