



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**FACULTAD DE INGENIERÍA**

**LAS COMPUTADORAS EN LA INGENIERÍA CIVIL**

**DR. RODOLFO LUTHE G.**

G-602633

*notas sobre el sistema  
integrado de ingeniería civil*



FACULTAD DE INGENIERIA

# las computadoras en la ingeniería civil

*Dr. Rodolfo Luthe G.*

## LAS COMPUTADORAS EN LA INGENIERIA CIVIL

por el Dr. Rodolfo Luthe G.

### 1. Introducción.

Este artículo tiene por objeto indicar las aplicaciones principales de las computadoras en la ingeniería civil, en donde su uso es cada vez más frecuente. Las computadoras han ocasionado un cambio radical en la ingeniería civil, este cambio afecta tanto al análisis y diseño de las estructuras como a la enseñanza misma de la profesión.

La computadora representa la culminación de dispositivos de cálculo como el ábaco, regla de cálculo, tablas, nomogramas, calculadoras de escritorio, etc., y más que una nueva herramienta, es un enfoque completamente diferente en métodos, conceptos y educación ingenieriles. Ningún otro desarrollo ha tenido tanta influencia en la historia de la ingeniería.

En las aplicaciones se usa principalmente la computadora digital, siendo razones posibles la facilidad en la programación y el hecho de que las características de la computadora concuerdan con los aspectos de precisión y meticulosidad, presentes en la mayoría de los ingenieros. En las aplicaciones se observa también la influencia del desarrollo en los métodos de solución, en los lenguajes de programación y en el avance del equipo, que ha permitido la elaboración de sistemas integrados.

### 2. Métodos de Solución.

Los primeros métodos que se programaron fueron los mismos que se usa-

ban manualmente, con la ayuda de la regla de cálculo y de la sumadora de escritorio. Dichos métodos tradicionales son anticuados e inadecuados en general para usarse con computadoras, ya que se tenía el mismo proceso sólo que más rápido. Un ejemplo es la aplicación del método de H. Cross para el análisis de marcos con carga lateral, que son simplificaciones del problema como resultado de la experiencia en este campo, pero que no tiene sentido su programación debido a las características de las computadoras.

Los nuevos enfoques hacen uso de las características de las computadoras, que tienen capacidad para almacenar grandes cantidades de información, operar a alta velocidad, efectuar una serie de operaciones especificadas, realizar decisiones lógicas, y obtener resultados correctos. El álgebra matricial utiliza con ventaja estas características y de ahí la aplicación actual de los métodos matriciales. Esto ha producido un cambio en la enseñanza, que se ha concentrado en el enfoque matricial de los problemas, con peligro de perder el sentido físico de los mismos, elemento valioso en la ingeniería civil. La aplicación de las computadoras ha desarrollado también nuevos tópicos como los métodos numéricos para computadoras, en donde el análisis numérico y las matemáticas proporcionan métodos susceptibles para programarse en la máquina.

En lo que se refiere al diseño estructural, las computadoras se usan con ventaja en la enseñanza y en la práctica. En principio para diseñar se tie-

3.

ne que efectuar una serie de cálculos que pueden ser laboriosos, para obtener las dimensiones de las secciones. Esta cantidad de operaciones limita el número de diseños que pudiera ser deseable considerar. Con el auxilio de la computadora es factible considerar una infinidad de condiciones, de manera que se puede realizar un trabajo más eficiente e instructivo en el diseño estructural. Al hacer referencia al diseño estructural con métodos manuales, se pensaba únicamente en el análisis de los esfuerzos. Con la computadora se pueden considerar etapas más avanzadas, como la de optimización de las estructuras.

Es conveniente observar que existen muchos temas que son importantes por sí mismos, que se desarrollan y que posteriormente encuentran aplicación en la solución de problemas ingenieriles. Es el caso del álgebra matricial, de la solución de sistemas de ecuaciones, de la programación lineal, ecuaciones diferenciales ordinarias y parciales, investigación de operaciones, topología, etc.

La selección del método de solución implica necesariamente que se entienda completamente el problema que se desea resolver.

### 3. Lenguajes de Programación.

Las aplicaciones han seguido el progreso de los lenguajes de programación. El lenguaje más elemental es el de máquina o absoluto y su uso eficiente implica un programador con características de ingenio y buena memoria. Aun para problemas sencillos la programación es bastante laboriosa.

4.

La etapa siguiente la constituyen los lenguajes simbólicos, que permiten indicar las operaciones en forma abreviada y que tienen características que facilitan la programación, comparados con los lenguajes de máquina. Si los lenguajes no hubieran superado esta etapa, las aplicaciones no hubieran alcanzado el grado de desarrollo actual, debido a la dificultad de la programación y a la falta de programadores eficientes en dichos lenguajes.

Los lenguajes de procedimiento han facilitado enormemente la programación. Ejemplos de estos lenguajes son las distintas versiones de FORTRAN, ALGOL, COBOL, BASIC y PL-1, que permiten la especificación de operaciones en unidades de procesamiento denominadas proposiciones. La gran mayoría de las aplicaciones está relacionado con algunos de estos lenguajes de procedimiento, que han influido poderosamente en el uso cada vez más general de las computadoras.

Finalmente, existen los lenguajes orientados a problemas o superlenguajes, que utilizan en la programación la terminología establecida en ese tema.

De este manera, el problema se describe a la computadora básicamente en los mismos términos en que se le describiría a una persona conocedora del tema, haciendo más eficiente y sencilla la relación entre el hombre y la máquina. Dos lenguajes orientados a problemas que han tenido un gran impacto en ingeniería son COGO y STRESS: el primero se aplica a topografía y problemas geométricos y el segundo al análisis de estructuras. Se ha desarrollado con éxito un medio de comunicación con la máquina, en el lenguaje del ingeniero, estableciéndose un sistema eficiente en una gran

gama de problemas. Sin embargo, no se logró que estos sistemas fueran de fácil ampliación y han permanecido con algunas limitaciones, debido al equipo y a los sistemas operativos. STRESS tiene limitaciones en los datos de entrada cuando se trata de una serie de problemas que se repiten. Su aplicación es ideal para estructuras especiales. Un programa semejante que se ha ampliado y considera el diseño de los elementos de una estructura es el AMECO.

Se han desarrollado también programas de fines específicos, como FRAME para análisis de estructuras planas y STAIR para el análisis y diseño de torres de transmisión, considerándolas como estructuras espaciales.

Los lenguajes orientados a problemas se han aplicado a distintas disciplinas y están disponibles para problemas de ingeniería. Ejemplo de ello son GROPE para problemas de optimización, GISM para un sistema de operaciones matriciales, CSMP para ecuaciones diferenciales ordinarias y PDEL para problemas caracterizados por ecuaciones diferenciales parciales. Con las nuevas máquinas se proporciona una biblioteca que varios lenguajes orientados a problemas, que pueden aplicarse a la ingeniería. Por ejemplo, el MPS que usa la técnica de programación lineal para determinar la solución óptima de un sistema de desigualdades lineales, basándose en el método Simplex revisado. El PMS se aplica al control y planeación de proyectos; se usa el método de la ruta crítica, calculándose el costo del proyecto, controlándose el avance y preparando reportes. De manera semejante, el MSCS está formado por varios programas relacionados que permiten

la planeación calendarización y coordinación de grandes proyectos que varían desde la construcción hasta la planeación fiscal. El SSP consiste de una serie de subrutinas aplicables a problemas científicos y de ingeniería. En estadística comprende análisis de variancia, correlaciones, regresiones y números aleatorios; en matemáticas se incluye álgebra matricial, inversión de matrices, vectores y valores característicos, sistemas de ecuaciones, integración, ecuaciones diferenciales, raíces reales y complejas, mínimos e interpolación. El ECAP se aplica a problemas de ingeniería eléctrica.

Se observa que cada desarrollo de los lenguajes aumenta la facilidad de programación, advirtiendo que la preparación técnica de los ingenieros es adecuada para la programación y uso eficiente de las computadoras.

#### 4. Aplicaciones.

Las aplicaciones de las computadoras digitales son prácticamente ilimitadas, como puede observarse en las revistas, libros, congresos, etc., sobre este tema. En un principio existió una fuerte tendencia hacia los métodos de análisis de las estructuras, programándose desde los métodos tradicionales manuales, no adecuados en general para computadoras, hasta los nuevos métodos que usan eficientemente las computadoras.

A continuación se desarrollaron métodos para el diseño estructural, aprovechando la característica de poder tomar decisiones con la máquina. Con esta característica se desarrolló el diseño estructural para obtener la optimización de las estructuras, es decir, la combinación de propiedades geo-

7.

métricas y físicas, de manera que la estructura cumpla su finalidad y sea la más económica posible. Esta optimización puede automatizarse con la máquina y para problemas complejos es la única manera de resolverlos. Los criterios han sido el del mínimo peso de las estructuras y el del diseño al límite; si no son los apropiados se considera la estructura de costo mínimo, pudiendo tenerse también un criterio de deformaciones. Pueden utilizarse técnicas iterativas, de programación lineal y no lineal, añadiéndose otros conceptos. Por ejemplo, al concepto de diseño al límite se pueden añadir frecuencias resonantes alternativas, como es el caso de una armadura espacial que contiene varios satélites, los que se pondrán en órbita simultáneamente, investigándose distintas configuraciones de dicha armadura.

El diseño con computadoras se complementa con las graficadoras que pueden dibujar desde los diagramas de elementos mecánicos y secciones transversales hasta estructuras completas como edificios y puentes. En estos últimos, es frecuente que sólo se necesiten cambios pequeños en longitud, espesor, etc., de modelos estándar, pudiendo desarrollar la máquina el dibujo, si se le indican únicamente las modificaciones. Las pantallas de tubos de rayos catódicos, con posibilidad de usar plumas con punto sensible (haz luminoso) en la pantalla, permiten al calculista modificar el dibujo de la pantalla, almacenando la computadora dichas modificaciones y estableciéndose una relación eficiente entre el hombre y la máquina. De otra manera, el tiempo que necesita el dibujante en el restirador retrasa tanto el tiempo de entrega de los planos como el desarrollo del proyecto.

8.

Se han desarrollado también programas de simulación de sistemas. La técnica de simulación consiste esencialmente en la creación de un modelo matemático que represente una situación real, de manera que la experimentación del modelo permita investigar sistemas complejos, elaborando y probando teorías nuevas. Se simula el comportamiento dinámico de puentes y vehículos, presas, ríos, sistemas de riego e hidroeléctricos, etc., y en general cualquier problema ingenieril. Para simplificar los programas de simulación, se han desarrollado varios lenguajes orientados a este problema, siendo los más conocidos GPSS, SIMSCRIPT, GASP y CSMP.

Los programas estructurales de uso más frecuente son los siguientes:

Análisis de marcos de concreto reforzado y de acero

Análisis de puentes

Programa STRESS

Diseño de vigas compuestas

Análisis y diseño de vigas continuas

Geometría horizontal, vertical y esviada de puentes

Diseño de vigas soldadas

Análisis y diseño de estructuras articuladas

Líneas de influencia

Momentos, fuerzas cortantes y desplazamientos en vigas

Cargas lateral y vertical en edificios

Diseño de vigas de concreto presforzado

Análisis y diseño de losas

Muros de retención

Análisis y diseño elástico o al límite de columnas

Diseño y cargas en columnas metálicas

Elevaciones de losas de puentes

Envolvente de momentos en vigas contínuas

Diseño de cimentaciones y pilotes

Análisis de marcos rígidos de varios pisos

Análisis de armaduras contínuas

Análisis de vigas de concreto

Análisis de torres de transmisión

Esfuerzos y flexibilidad en tuberías

Análisis tridimensional de marcos

Análisis de vigas en balcón

Análisis de losas y cascarones

Esfuerzos y deformaciones de presas en arco

Análisis de arcos elásticos

Vigas en cimentación elástica

Análisis de subestructuras

Geometría estructural

Propiedades de secciones

Detalles de uniones metálicas

Escaleras en espiral

Análisis y diseño de silos y chimeneas

Análisis de estructuras subterráneas

Grupos de pilotes con carga lateral

Como puede observarse, la mayoría de los programas son de uso rutinario y frecuente en la ingeniería.

También se utilizan las computadoras analógicas, que simulan fácilmente el comportamiento dinámico de cualquier sistema, resolviendo la ecuación diferencial o sistemas de ecuaciones diferenciales que lo representan. Las soluciones se presentan usualmente en forma gráfica en un osciloscopio y el operador puede cambiar los valores de los parámetros, experimentando el comportamiento del modelo y observando instantáneamente el efecto del cambio de parámetros. Por esta razón, la analógica es básicamente una herramienta para la investigación y para la enseñanza.

En la computadora analógica se puede cambiar la escala del tiempo, haciendo lentas las soluciones rápidas o acelerando las soluciones lentas. Por ejemplo, el análisis de explosiones que ocurren en una fracción de segundo, tienen que hacerse más lentas en la computadora para poder estudiarlas. Otros procesos, como la fatiga en metales, que puede durar varios años, tienen que acelerarse en la máquina para poder analizarlos. En algunos sistemas es imposible o muy peligroso estudiar las condiciones críticas de operación, como en un reactor nuclear o en un sistema hidroeléctrico. Sin embargo, el modelo puede llevarse al límite de destrucción en la computadora y proporcionar el hecho la única manera segura de analizar con detalle el comportamiento del sistema en situaciones críticas.

La computadora analógica es una herramienta útil en la simulación y solución de sistemas de ecuaciones algebraicas y diferenciales no lineales, que describen comportamientos dinámicos de procesos, especialmente cuando

no se requiere una alta precisión en los cálculos. Se aplica en problemas de transferencia de calor, análisis de vibraciones, análisis dinámico de esfuerzos, control de cohetes y recientemente en bioingeniería.

Las personas que usan las computadoras digital y analógica se han especializado a tal grado que no hay comunicación entre ellas. Se admite en general que algunos problemas son más adecuados para alguno de los dos tipos de computadora. Sin embargo, para algunos problemas de ingeniería se hizo necesario utilizar las técnicas de computación digital y analógicas, dando como resultado la computadora híbrida.

La computadora híbrida es una combinación de computadoras analógica y digital, con un sistema adecuado de comunicación y operación, ya que la analógica opera en forma continua y en paralelo, de manera que el tiempo de solución es independiente de la complejidad del problema, mientras que la digital opera en forma discretizada y secuencial. El problema de comunicación no es sólo la conversión de analógica a digital y viceversa, sino también la solución de los complejos problemas de tiempo para asegurar que la transferencia de información entre los dos sistemas se realiza eficientemente.

La computadora híbrida se aplica en programación lineal y optimización, con los métodos de máxima pendiente, relajaciones y búsqueda aleatoria. Se aplica con ventaja en ecuaciones diferenciales ordinarias y parciales, especialmente si se aplica la generación de funciones de variables múltiples; sistemas de control con combinaciones de variables continuas y dis-

cretas; procesamiento de datos y análisis estadísticos. Específicamente en ingeniería civil se puede usar la computadora híbrida en diseño estructural, experimentando varios diseños posibles y determinando la magnitud y posición de las concentraciones de esfuerzos. Las estructuras se pueden representar con barras, articulaciones y masas concentradas con amortiguadores, sujetas a sistemas de cargas. En la computadora analógica se puede simular la variación de los esfuerzos con el tiempo en una estructura, debido a sistemas externos de carga como viento y sismo. Con la computadora digital se cambian las constantes de los resortes y los coeficientes de amortiguamientos, lo que equivale a cambiar los materiales de construcción y se cambian también las distancias entre masas concentradas, lo que corresponde a cambiar el diseño estructural, repitiéndose estos cambios para todas las condiciones de carga. De esta manera se puede considerar un gran número de problemas de diseño en un tiempo corto, utilizando la salida gráfica para mayor eficiencia.

De lo anterior se observa que el desarrollo del equipo permiten considerar problemas más complejos y que la computación es útil en la experimentación y en la investigación, estando limitado ésta última no por las computadoras sino por la capacidad humana.

##### 5. Sistema Integrado.

El avance en equipo y lenguajes ha permitido la elaboración del Sistema Integrado de Ingeniería Civil (1967), denominado ICES, y que consiste de una serie de subsistemas aplicable cada uno de ellos a una disciplina particular, con la característica de poder combinarlos. Está diseñado como un siste-



ma dinámico de módulos, en donde los subsistemas se pueden modificar, ampliar, añadir o sustituir por versiones mejoradas. Se necesitan estructuras dinámicas de arreglos para una mayor eficiencia del sistema, superándose el espacio disponible de memoria, reorganizándolo y creando archivos de datos en memoria secundaria si fuera necesario.

Los subsistemas disponibles son: STRUDL para el análisis y diseño de estructuras en dos y tres dimensiones, y es más flexible y poderoso que - - STRESS; COGO para problemas geométricos y de topografía; TABLE para manejo y almacenamiento de información tabulada, que puede usarse en combinación con cualquier otro subsistema; SEPOL para analizar los esfuerzos en suelos y los asentamientos de estructuras; SLOPE para estimar el factor de seguridad en la estabilidad de taludes; ROADS para la localización y diseño de carreteras y vías de ferrocarril; BRIDGE es aplicable al diseño de puentes, intersecciones de carreteras, pasos a desnivel y problemas similares; TRANSET para predicción y análisis de flujos en redes de transporte, aplicable también a redes eléctricas; PROJECT para ayudar en la planeación y control de proyectos de construcción; HYDRO para problemas de hidráulica; DYNAL para el análisis dinámico de estructuras complejas tridimensionales como edificios, tuberías, plataformas de perforación y varias estructuras aeroespaciales; OPTECH para técnicas de optimización.

#### 6. Ejemplos de Aplicación.

Como ejemplo de aplicación del subsistema STRUDL considérese el marco plano de la figura 1.

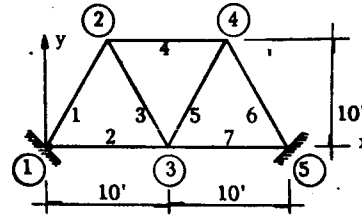


FIG. 1

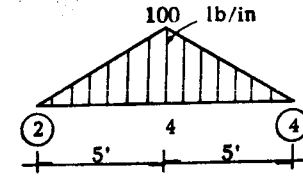


FIG. 2

Se desea determinar los elementos mecánicos, desplazamientos y reacciones en los apoyos, para los sistemas de carga:

- Sistema 1: Carga concentrada vertical hacia abajo de 10 Kips, aplicada en el nudo 3.
- Sistema 2: Carga distribuida lineal en el miembro 4, como se indica en la figura 2.
- Sistema 3: La mitad del sistema 1 más el sistema 2.

Repetir el problema si el apoyo de la derecha es un apoyo simple. Solicitar además las reacciones en los apoyos sólo para el sistema 3 de carga.

Para todos los miembros  $A = 10 \text{ in}^2$ ,  $I = 100 \text{ in}^4$  y  $E = 30 \times 10^6 \text{ lb/in}^2$ .

Los requisitos mínimos son los siguientes:

1. Subsistema 'identificación' 'título'
2. Tipo de estructura
3. Coordenadas de los nodos
4. Identificación de los miembros
5. Propiedades de los miembros

6. Sistemas posibles de carga
7. Sistemas de carga considerados
8. Tipo de análisis
9. Resultados deseados

El problema se especificaría de la manera siguiente:

```

STRUDL   'EJEM 1'   'MARCO PLANO'
$
PROGRAMADOR R LUTHE
TYPE PLANE FRAME
JOINT COORDINATES
1 X 0.0 Y 0.0 SUPPORT
2 X 60.0 Y 120.0
3 X 120.0 Y 0.0
4 X 180.0 Y 120.0
5 X 240.0 Y 0.0 SUPPORT
MEMBER INCIDENCES
1 1 2
2 1 3
3 2 3
4 2 4
5 3 4
6 4 5
7 3 5
MEMBERS 1 TO 7 PROPERTIES PRISMATIC
AX 10 0 IZ 100.0

```

```

CONSTANTS E 30000000.0 ALL
LOADING 1
JOINT 3 LOAD FORCE Y -10000.0
LOADING 2 'CARGA DISTRIBUIDA'
MEMBER 4 LOAD FORCE Y LINEAR FR WA 0.0 -
WB -100. LA 0.0 LB 0.5
MEMBER 4 LOAD FORCE Y LINEAR FR WA -100.0 -
WB 0.0 LA 0.5 LB 1.0
LOADING COMBINATION 3 COMBINE 1 0.5 2 1.0
LOADING LIST ALL
STIFFNESS ANALYSIS
LIST FORCES REACTIONS DISPLACEMENTS
CHANGES
JOINT 5 RELEASE FORCE X MOMENT Z
LOADING LIST ALL
STIFFNESS ANALYSIS
LIST FORCES REACTIONS DISPLACEMENTS
LOADING LIST 3
LIST REACTIONS ALL
FINISH

```

Observaciones: Las tarjetas con el símbolo \$ se usan para comentarios y se imprimen en los resultados. Una instrucción que termina con - continúa en la siguiente instrucción. Las unidades deben ser consistentes para todos los elementos y se usan pulgadas y libras, pero pueden especificar-

se pies, centímetros o metros para distancias y Kips, Kg. o ton. para --  
fuerzas, por medio de la instrucción UNITS.

Como ejemplo de aplicación del subsistema COGO, considérese el proble-  
ma de la figura 3, en la cual se localizan las coordenadas de puntos y se  
calcula el área de la sección sombreada.

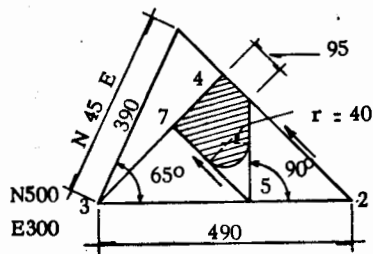


FIG. 3

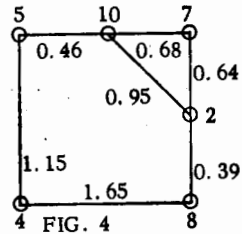


FIG. 4

El problema se especificaría de la manera siguiente:

```

COGO
STORE POINT 3 N 500 E 300
LOCATE POINT 6 FROM POINT 3 DISTANCE 390 AT
      BEARING N 45 00 E
LOCATE POINT 2 FROM POINT 3 3 DISTANCE 490 AT
      BEARING N 45 E PLUS ANGLE 65
STORE DISTANCE 'X' DISTANCE FROM POINT 2 TO POINT 6
      DIVIDED BY 2
LOCATE POINT 8 FROM POINT 2 DISTANCE 'X' BEARING FROM
      POINT 2 TO POINT 6
LOCATE POINT 4 FROM POINT 8 DISTANCE 95 AZIMUTH
      FROM POINT 2 TO POINT 6
    
```

```

LOCATE POINT 5 PROJECT POINT 8 ON LINE THRU POINT 3
      TOWARD POINT 2
LOCATE POINT 7 INTERSECT LINE THRU POINT 3 TOWARD POINT 4
      WITH LINE THRU POINT 5 AT AZIMUTH FROM POINT 2 TO
      POINT 6
STORE CURVE 1, PB AT 8, PIAT 5, RADIUS 40, PA AT 7
STORE PARCEL 'A1' POINT 7, POINT 4, POINT 8, CURVE 1, POINT 7
AREA PARCEL 'A1'
PRINT POINTS 2 TO 8
PRINT CURVE 1
FINISH
    
```

Existe una forma abreviada de la instrucciones anteriores. También, si  
la primera letra es una D indica que se repite la instrucción previa y só-  
lo se cambian los datos.

Como ejemplo de aplicación del subsistema TRANSET, considérese el  
problema de la figura 4, en donde se determinan las distancias mínimas  
de todos los nodos o zonas con respecto a la zona 5. Los números entre  
los nodos indican las distancias entre ellos. El problema se especificaría  
de la manera siguiente:

```

TRANSET
READ NETWORK 'RED-1', ZONES 6, VOLUME DELAY SET 'PRUEBA'
LINK 1 5 4 1.15 2 2 5 10 0.46 2 2 4 5 1.15 2 2 10 5 0.46 2 2
LINK 2 10 7 0.68 2 2 10 2 0.95 2 2 7 2 0.64 2 2 2 8 0.39 2 2
LINK 3 7 10 0.68 2 2 2 10 0.95 2 2 2 7 0.64 2 2 8 2 0.39 2 2
    
```

G.

19.

LINK 4 4 8 1.65 2 2 8 4 1.65 2 2

EDIT NETWORK

OUTPUT PRINTOUT OF MINIMUM TRACES

\$ SE PUEDEN OBTENER TRAYECTORIAS MINIMAS, RED DESCRITA.

SELECT MINIMUM DISTANCE TREES, ORIGIN 5

FINISH

Otra ventaja de los nuevos sistemas es que pueden ser accesibles a personas o lugares que no disponen de equipo suficiente, bastando una terminal para ellos, conectada con un equipo poderoso instalado en otro lugar. Este es el concepto de tiempo compartido y la comunicación es para cualquier tipo de problema, aun cuando es evidente la ventaja del uso de los lenguajes orientados. Algunas ventajas son el no tener acceso en general a una explicación detallada del método de análisis y la dificultad de detectar algunos errores al introducir información en la terminal, proceso que además puede ser laborioso e implicar bastante tiempo de máquina, según la complejidad del problema. La operación de los nuevos equipos se hace económica aplicando los conceptos de multiprogramación y multiprocesamiento, que permiten el procesamiento simultáneo de varios programas, aumentando la eficiencia del sistema.

#### 7. Conclusiones.

Las computadoras se usan cada vez con más frecuencia en todos los campos de la ingeniería civil y otras disciplinas técnicas, siendo el elemento de enlace entre ellas la universalidad de las matemáticas.

20.

Las computadoras son herramientas que no sustituyen al ingeniero, sino que le permiten mayor libertad en la solución de problemas, satisfaciendo su curiosidad y no restringiéndolo a conceptos anticuados limitados.

Los lenguajes orientados a problemas han influido incrementando el uso de las computadoras, ya que facilitan la relación entre el hombre y la máquina, además de permitir un uso más eficiente de esta última. Las salidas gráficas han mejorado también esta relación entre el hombre y la máquina.

Los nuevos conceptos de tiempo compartido y multiprogramación facilitan el acceso a computadoras de gran tamaño, lo que ligado al desarrollo de lenguajes orientados en sistemas integrados, han influido en el aumento del número de usuarios de computadoras.

Se vislumbra el conocimiento y uso de computadoras como ocurre con la regla de cálculo, debiendo subordinarse la enseñanza del uso de una herramienta a la enseñanza de los principios.

#### 8. Bibliografía.

First Conference on Electronic Computation, ASCE Structural Division, Kansas, City, Mo., Noviembre 20-21, 1958

Second Conference on Electronic Computation, ASCE Structural Division, Pittsburgh, Pa., Septiembre 8-9, 1960

Third Conference on Electronic Computation, ASCE Structural Division, Boulder, Colo., Junio 19-21, 1963



FACULTAD DE INGENIERIA

Fourth Conference on Electronic Computation, ASCE Structural Division  
Los Angeles, Cal., Septiembre 7-9, 1966.

Fifth Conference on Electronic Computation, ASCE Structural Division,  
Lafayette, Indiana, Agosto 31 - Septiembre 2, 1970

Fenves S.J., et al, Stress: A User's Manual, M.I.T. Press, Cambridge,  
Mass., 1964.

Fenves S.J., Métodos de Computación en Ingeniería Civil, Limusa-Wiley  
México, 1969.

Bashkow T. R. (ed), Engineering Applications of Digital Computers,  
Academic Press, Nueva York, 1968.

Naylor T.H., Balintfy J.L., Burdick D.S., y Chu K., Computer Simulation  
Techniques, John Wiley, Nueva York, 1968.

Noronha L.G., Hybrid Computation and its Engineering Applications, Elec\_  
tronic Associates, México, 1968.

ICES Systems General Description, Roos Daniel (ED), Civil Engineering  
Systems Laboratory, M.I.T., Cambridge, Mass., 1967.

ICES STRUDL, Engineering User's Manual, Civil Engineering Systems La\_  
boratory M.I.T., Cambridge, Mass., 1967.

ICES COGO, Engineers' Guide, Civil Engineering Systems Laboratory  
M.I.T., Cambridge, Mass., 1967.

ICES TRANSET, Engineering User's Manual, Civil Engineering Systems  
Laboratory M.I.T., Cambridge, Mass., 1968.

*notas sobre el sistema  
integrado de ingeniería civil*

**sistemas de información**

**en**

**ingeniería civil**

*Dr. Rodolfo Luthe G.*

## SISTEMAS DE INFORMACION EN INGENIERIA CIVIL

por el Dr. Rodolfo Luthé

### 1. Generalidades.

Los desarrollos actuales en equipo y lenguajes de programación han permitido un gran adelanto en los sistemas de información en Ingeniería Civil. El uso cada vez más generalizado de los lenguajes orientados a problemas específicos ha facilitado la transmisión masiva de información, por medio de una metodología sencilla, ya que no se requieren conocimientos previos de programación y técnicas de análisis, sino únicamente conocer la terminología del tema considerado y establecer una comunicación con la computadora de manera semejante a como se efectuaría con otra persona.

Se ha generado en el MIT un sistema de información denominado Sistema Integrado de Ingeniería Civil (ICES).

Este sistema consiste en una serie de subsistemas, con particularidad de que cada subsistema es aplicable a una área de la ingeniería y con la característica de poder combinar subsistemas, además de que cualquier subsistema se puede modificar, ampliar o sustituir por versiones mejoradas.

Inicialmente el sistema lo desarrollaron más de 50 personas, con un costo superior a los 2 millones de dólares.

Las características de un sistema integrado, como el Sistema ICES, son las siguientes:

2.

- 1) Un lenguaje poderoso y flexible, orientado a los problemas del ingeniero, para comunicarse con la computadora.
  - 2) Terminales para que el ingeniero tenga acceso a la computadora.
  - 3) Orientación hacia la solución total de problemas, no sólo la parte de computación.
  - 4) Interacción entre distintas disciplinas o subsistemas para la solución de problemas de más de una área ingenieril.
  - 5) Organización de datos y mecanismo de transferencia de datos para la transferencia de información entre los subsistemas.
  - 6) Estructura modular interna a base de bloques.
  - 7) Estructura interna de datos que sea eficiente.
  - 8) Arreglos dinámicos de datos y programas utilizados en el problema que se está resolviendo.
  - 9) Capacidad suficiente del equipo para los programas que se necesitan.
- Originalmente al construirse las computadoras, había cierta separación entre las personas que desarrollaban los lenguajes y las que los utilizaban; se generaban lenguajes donde se tenían en cuenta las necesidades de los usuarios pero no se satisfacían completamente ya que no se apreciaban esas necesidades en su totalidad. Para remediar esta situación, en el caso del sistema ICES, se tiene un sistema que utilizan ingenieros y que está desarrollado por ingenieros.

## 2. Lenguaje de programación.

Considerando que las características del FORTRAN son adecuadas para la solución de problemas técnicos, se emplearon todas las instrucciones de FORTRAN para formar parte de un lenguaje de programación que se utiliza en ICES para generar los subsistemas. Este lenguaje de programación se llama ICETLAN, (ICES - FORTRAN), el cual se hizo más flexible y poderoso que el FORTRAN original.

El lenguaje ICETLAN lo desarrollaron personas que tienen experiencia en ciencias de información y en ingeniería. Contiene todas las proposiciones FORTRAN más algunas otras características para hacerlos más flexible y poderoso, siendo una de las principales el aspecto dinámico del arreglo de la memoria.

## 3. Ejemplo de un programa en ICETLAN.

Para ejemplificar el arreglo dinámico de la memoria, considérese el programa ICETLAN siguiente:

```
SUBROUTINE ADD (I)
COMMON A, B, C, . . .
DYNAMIC ARRAYS A, B, C
DEFINE C, I, HIGH
DO 5 N = 1, I
5 C (N) = A (N) * B(N)
DESTROY A
```

3.

4.

Esta subrutina multiplica dos arreglos dinámicos A y B de una sola dimensión, formando un nuevo arreglo dinámico C. Se observa que asociada con cada arreglo dinámico existe una posición COMMON conocida como indicador de palabra codificada. Esta posición o localidad contiene la información siguiente sobre el arreglo dinámico:

- 1) El tamaño de ese arreglo.
- 2) La residencia del arreglo ( si es memoria primaria, secundaria, etc.)
- 3) Un indicador de la posición inicial de arreglo.
- 4) La jerarquía y prioridad del arreglo para usarse en la reorganización de memoria.

De esta manera un número suficiente de arreglos pueden transferirse a memoria secundaria para dejar espacio disponible para nuevos arreglos.

La proposición DYNAMIC ARRAYS se utiliza para especificar todos los arreglos dinámicos. La proposición DEFINE la usa el programador para especificar información acerca del arreglo dinámico que se almacenará en el indicador de palabra codificado. En el ejemplo se ocasiona que el tamaño (I) y la prioridad (HIGH) del arreglo dinámico C, se inserte en el indicador de palabra codificado del arreglo C.

El arreglo A, al ejecutarse la proposición DESTROY A, se borra o destruye y queda disponible el espacio correspondiente. La proposición RELEASE B significa que el arreglo B se aísla temporalmente pero que se tiene acce-

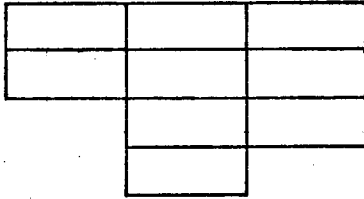


5.

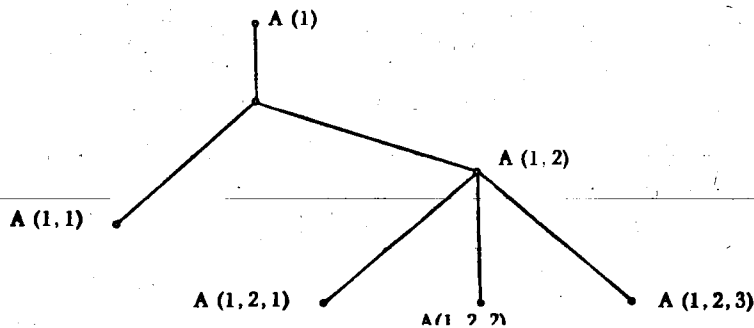
so a él si se desea usarlo nuevamente. De esta manera con los arreglos dinámicos, se aumenta la eficiencia cuando ocurre una reorganización de la memoria. Se pueden manejar arreglos de cualquier número de dimensiones y un arreglo se particiona internamente en un conjunto de subarreglos, pudiendo ser diferente el tamaño de cada subarreglo.

#### 4. Arreglos y su estructura.

La figura siguiente muestra un arreglo de dos dimensiones en donde el tamaño de cada subarreglo (por columnas), varía en 2, 3 y 1 unidades de bloque



La estructura del subarreglo puede variar de manera que las estructuras en forma de árbol se pueden representar utilizando la notación de los arreglos, como se muestra en la figura siguiente:



6.

Otras características adicionales a la de memoria dinámica son procesamiento de listas, organización de datos, transferencia de datos y manejo de matrices, disponibles todas ellas con ICETRAN.

La manera de proceder una vez que se tiene un programa o subrutina -- ICETRAN es usar el precompilador de ICES que traduce o convierte las proposiciones ICETRAN en proposiciones FORTRAN (es un primer paso de traducción o conversión). Después de FORTRAN puede pasarse a lenguaje de máquina en un compilador de FORTRAN usual y obtener el programa objeto correspondiente.

#### 5. Características del Sistema Operativo.

En el sistema operativo del sistema integrado hay otras características o componentes, siendo una de ellas el ICES-EXECUTIVE. Este ejecutor de ICES va a procesar las declaraciones del lenguaje orientado al problema, que proporciona el ingeniero y efectúa las operaciones siguientes:

- 1) Lectura y descifrado o interpretación de la declaración.
- 2) Análisis, conversión y almacenamiento de datos.
- 3) Efectuar verificación de errores o consistencia de datos.
- 4) Transferencia de control a la subrutina indicada para la ejecución de la declaración.

Existe la posibilidad de modificar nombres de declaraciones o sea, que se pueden considerar modificadores de nombre para una sola declaración. --

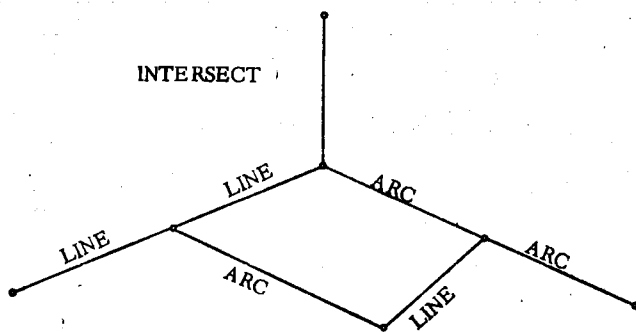
Considérese el subsistema COGO y la declaración INTERSECT que puede indicarse en las tres formas siguientes:

INTERSECT POINT 8 LINE 10 LINE 20

INTERSECT POINT 8 LINE 10 ARC 20 NEAR 4

INTERSECT POINT 8 ARC 10 ARC 20 NEAR 4

Los datos de la declaración son el número asignado al punto de intersección y los números de los dos elementos geométricos (arco-línea) que se intersectan. Si se usa arco y recta o dos arcos pueden existir dos intersecciones y el punto NEAR en las dos últimas declaraciones indica un punto conocido cercano a la intersección deseada. El ingeniero piensa todo esto como una sola declaración, pero el programador piensa en tres declaraciones distintas y visualiza una estructura como la mostrada, donde se tengan todas las combinaciones posibles.



### 6. Lenguaje de definición de declaraciones.

Existe en el sistema ICES el lenguaje de definición de declaraciones CDL, con el cual el programador establece las declaraciones de los subsistemas y genera tablas de declaraciones. Es un lenguaje orientado hacia el que desarrolla subsistemas y por eso se dice que ICES incluye un lenguaje - - orientado que genera lenguajes orientados a problemas. El lenguaje de definición de declaraciones se puede usar para añadir, modificar o borrar de declaraciones existentes en un subsistema.

Supóngase que en el subsistema COGO se desea añadir la declaración - - STORE para almacenar las coordenadas de un punto dado. Un ingeniero escribiría esta declaración en la forma

STORE POINT 5 X 110.30 Y 215.5

Para añadir esta declaración en el vocabulario o en la gramática de COGO se escribe el programa de definición o declaración siguiente:

```
SYSTEM 'COGO'
ADD 'STORE'
ID 'P' INTEGER 'NPOINT' REQUIRED
ID 'X' REAL 'XCOORD' STANDARD 0
ID 'Y' REAL 'YCOORD' STANDARD 0
EXECUTE 'STORE'
FILE
```

Si se desea el almacenamiento del dato usando el idioma castellano se cambiaría STORE por ALMAC. Las palabras subrayadas pertenecen al vocabulario del lenguaje de definición de declaraciones.

La cantidad P es un entero y se necesita, por lo que si se olvidara poner el número 5, el diagnóstico indicaría que falta ese punto porque se ha definido como "requerido" o existente. Si se escribiera 5.0 indicaría error porque se ha definido como cantidad entera.

La X y Y son cantidades reales y se van a almacenar bajo esta denominación; pueden ser no requeridas y si no se escriben se suponen igual a cero.

EXECUTE indica que se ejecute la instrucción y FILE que se archive.

7. Organización de datos.

En cuanto a la organización y localización de datos y programas hay programas del sistema de organización de datos que controlan los archivos que están en memoria secundaria y que están asociados con la solución del problema. Esto es importante porque estos programas permiten que el mismo arreglo sea usado por distintos subsistemas. Para esto se necesita un archivo de nombres de arreglos y variables que pueden tener distinta denominación o identificación para distintos subsistemas. Cada subsistema puede hacer referencia a él con nombre distinto, por lo que se necesita un control para llegar al mismo arreglo de datos. También se puede operar fácilmente con estos archivos codificándolos, solicitando su impresión o borrándolos. Además, lo anterior permite que distintos ingenieros

trabajando en distintos problemas tengan acceso a los mismos datos.

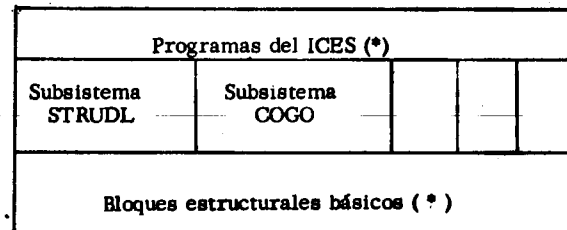
ICES funciona desde fines de 1965 y desde entonces se ha venido mejorando, pudiéndose hacer uso de graficadoras, pantallas, osciloscopios, etc., para hacer más flexible y útil el sistema. ICES se ha desarrollado en forma de módulos de manera que muchos de ellos se puedan cambiar, borrar, etc.

Existe también un supervisor del ICES que coordina a todos los programas del sistema y supervisa el uso de la computadora.

Para el desarrollo del sistema se ha tenido en cuenta que se usa bastante tiempo de máquina, pero se ha preferido minimizarlo permitiendo resolver totalmente el problema y proporcionar todas las alternativas para la solución de un problema. Es el caso de la memoria dinámica que utiliza más tiempo de máquina pero permite un estudio completo del problema al aumentar la capacidad disponible en la memoria en la solución de problemas.

8. Resumen del Sistema ICES.

El sistema ICES total puede presentarse en forma resumida como se indica en la figura siguiente:



(\*) Aplicados a todo el sistema.

El sistema ICES es un marco de referencia para que los ingenieros puedan desarrollar un trabajo personal y con esta experiencia ampliar la capacidad del ICES.

ICES implica varias relaciones o asociaciones: entre máquinas de tercera generación y los programas del sistema; entre el ICETLAN y los programadores, entre el lenguaje de definición de declaraciones y el ingeniero, y además, entre el sistema integrado de computación y la organización de ingenieros que trabajan en el sistema.

El ICES fue financiado hasta 1970 por nueve compañías:

Fundación Ford

Compañía I. B. M.

M. I. T. Lincoln Laboratories

Massachusetts Bay Transportation Authority

McDonnell Automation Company

Departamento de Obras Públicas de Massachusetts

National Science Foundation

Portland Cement Association

Sociedad de Transportes del Gobierno de Estados Unidos.

### 9. Subsistemas del ICES.

A continuación se presenta un resumen de los distintos subsistemas del ICES

### 9.1 Subsistema STRUDL (STRUctural Design Language)

El subsistema STRUDL está diseñado para auxiliar al ingeniero en el proceso de diseño. Se considera una amplia gama de tipos de estructuras 2 y 3 dimensiones, considerándose armaduras, marcos y elemento finito. Se pueden combinar estos componentes para problemas de análisis y diseño.

STRUDL se diseñó de manera que el usuario especifique datos y procedimientos de operación de estos mismos. La información de la estructura se puede almacenar, recuperar y manejar por medio de declaraciones. Los datos pueden arreglarse de manera que sólo parte de la información que se utiliza, pueda obtenerse en cualquier instante

STRUDL II se aplica tanto a estructuras porticadas como a medio continuo (aplicación del elemento finito). Para estructuras porticadas los datos pueden especificar miembros, nudos y condiciones de carga nombrándolos arbitrariamente. Los datos se pueden modificar continuamente en cualquier parte del problema.

Las posibilidades de análisis incluyen análisis indeterminado o hiperestático, análisis preliminar suponiendo sistemas de carga y un análisis isostático que no requiere propiedades de los componentes.

Los resultados pueden consistir en fuerzas y deformaciones de cualquier parte de la estructura, fuerzas y esfuerzos en los miembros, y combinaciones como máximos esfuerzos, fuerzas, envolventes de esfuerzos, etc.

Se pueden combinar distintas condiciones de carga.

*notas sobre el sistema  
integrado de ingeniería civil*

**el sistema integrado  
de  
ingeniería civil (ices)**

*Dr. Rodolfo Luthe G.*

## EL SISTEMA INTEGRADO DE INGENIERIA CIVIL (ICES)

### 1. Generalidades

El ICES ha sido desarrollado en el MIT por más de 50 personas, con un costo superior a los 2 millones de dólares.

Cualquier sistema integrado de ingeniería utilizando computadoras, como el Sistema ICES, debe tener las características siguientes:

- 1) Un lenguaje poderoso y flexible, orientado a los problemas del ingeniero, para comunicarse con la computadora.
- 2) Terminales para que el ingeniero tenga acceso a la computadora.
- 3) Orientación hacia la solución total de problemas, no sólo la parte de computación.
- 4) Interacción entre distintas disciplinas o subsistemas para la solución de problemas de más de una área ingenieril.
- 5) Organización de datos y mecanismo de transferencia de datos para la transferencia de información entre los subsistemas.
- 6) Estructura modular interna a base de bloques.
- 7) Estructura interna de datos que sea eficiente.
- 8) Arreglos dinámicos de datos y programas utilizados en el problema que se está resolviendo.

- 9) Capacidad suficiente del equipo para los programas que se necesitan.

Originalmente al construirse las computadoras, había cierta separación entre las personas que desarrollaban los lenguajes y las que los utilizaban; se generaban lenguajes donde se tenían en cuenta las necesidades de los usuarios pero no se satisfacían completamente ya que no se apreciaban esas necesidades en su totalidad. Para remediar esta situación, en el caso del sistema ICES, se tiene un sistema que utilizan ingenieros y que está desarrollado por ingenieros.

### 2. Lenguaje de programación

Considerando que las características del FORTRAN son adecuadas para la solución de problemas técnicos, se emplearon todas las instrucciones de FORTRAN para formar parte de un lenguaje de programación que se utiliza en ICES para generar los subsistemas. Este lenguaje de programación se llama ICETAN, (ICES - FORTRAN), el cual se hizo más flexible y poderoso que el FORTRAN original.

El lenguaje ICETAN lo desarrollaron personas que tienen experiencia en ciencias de información y en ingeniería. Contiene todas las proposiciones FORTRAN más algunas otras características para hacerlo más flexible y poderoso, siendo una de las principales el aspecto dinámico del arreglo de la memoria.

### 3. Ejemplo de un programa en ICETAN

Para ejemplificar el arreglo dinámico de la memoria, considérese el programa ICETAN siguiente:

```

SUBROUTINE ADD (1)
COMMON A, B, C, . . .
DYNAMIC ARRAYS A, B, C
DEFINE C, 1, HIGH
DØ 1 L = 1, 1
1 C (L) = A (L) + B(L)
DESTROY A
RELEASE B
RETURN
END

```

Esta subrutina suma dos arreglos dinámicos A y B de una sola dimensión, formando un nuevo arreglo dinámico C. Se observa que asociada con cada arreglo dinámico existe una posición COMMON conocida como indicador de palabra codificada. Esta posición o localidad contiene la información siguiente sobre el arreglo dinámico:

- 1) El tamaño de ese arreglo.
- 2) La residencia del arreglo (si es memoria primaria, secundaria, etc.)
- 3) Un indicador de la posición inicial del arreglo.
- 4) La jerarquía y prioridad del arreglo para usarse en la reorganización de memoria.

De esta manera un número suficiente de arreglos pueden transferirse a

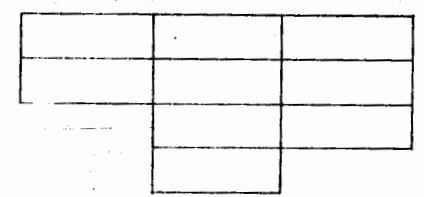
memoria secundaria para dejar espacio disponible para nuevos arreglos.

La proposición DYNAMIC ARRAYS se utiliza para especificar todos los arreglos dinámicos. La proposición DEFINE la usa el programador para especificar información acerca del arreglo dinámico que se almacenará en el indicador de palabra codificado. En el ejemplo se ocasiona que el tamaño (1) y la prioridad (HIGH) del arreglo dinámico C, se inserte en el indicador de palabra codificado del arreglo C.

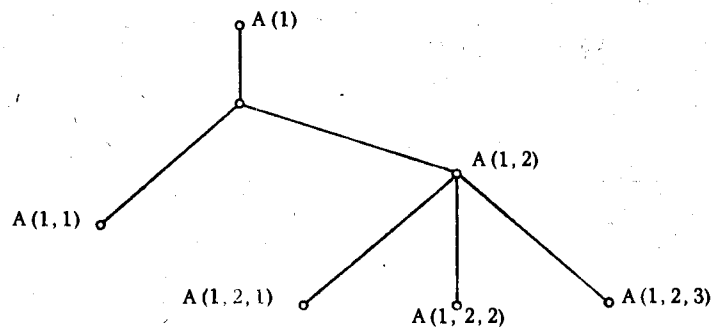
El arreglo A, al ejecutarse la proposición DESTROY A, se borra o destruye y queda disponible el espacio correspondiente. La proposición RELEASE B significa que el arreglo B se aísla temporalmente pero que se tiene acceso a él si se desea usarlo nuevamente. De esta manera con los arreglos dinámicos, se aumenta la eficiencia cuando ocurre una reorganización de la memoria. Se pueden manejar arreglos de cualquier número de dimensiones y un arreglo se particiona internamente en un conjunto de subarreglos, pudiendo ser diferente el tamaño de cada subarreglo.

4. Arreglos y su estructura

La figura siguiente muestra un arreglo de dos dimensiones en donde el tamaño de cada subarreglo ( por columnas ), varía en 2, 4 y 3 unidades de bloque



La estructura del subarreglo puede variar de manera que las estructuras en forma de árbol se pueden representar utilizando la notación de los arreglos, como se muestra en la figura siguiente:



Otras características adicionales a la de memoria dinámica son procesamiento de listas, organización de datos, transferencia de datos y manejo de matrices, disponibles todas ellas con ICETLAN.

La manera de proceder una vez que se tiene un programa o subrutina ICETLAN es usar el precompilador de ICES que traduce o convierte las proposiciones ICETLAN en proposiciones FORTRAN (es un primer paso de traducción o conversión). Después de FORTRAN puede pasarse a lenguaje de máquina en un compilador de FORTRAN usual y obtener el programa objeto correspondiente.

##### 5. Características del Sistema Operativo

En el sistema operativo del sistema integrado hay otras características o componentes, siendo una de ellas el ICES-EXECUTIVE. Este ejecutor

de ICES va a procesar las declaraciones del lenguaje orientado al problema, que proporciona el ingeniero y efectúa las operaciones siguientes:

- 1) Lectura y descifrado o interpretación de la declaración.
- 2) Análisis, conversión y almacenamiento de datos.
- 3) Efectuar verificación de errores o consistencia de datos.
- 4) Transferencia de control a la subrutina indicada para la ejecución de la declaración.

Existe la posibilidad de modificar nombres de declaraciones o sea, que se pueden considerar modificadores de nombre para una sola declaración. Considérese el subsistema COGO y la declaración INTERSECT que puede indicarse en las tres formas siguientes:

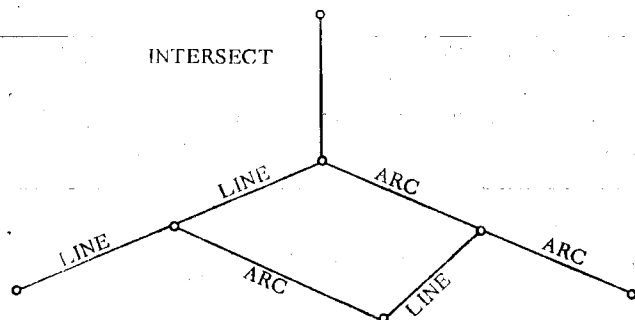
```

INTERSECT POINT 8 LINE 10 LINE 20
INTERSECT POINT 8 LINE 10 ARC 20 NEAR 4
INTERSECT POINT 8 ARC 10 ARC 20 NEAR 4
  
```

Los datos de la declaración son el número asignado al punto de intersección y los números de los dos elementos geométricos (arco-línea) que se intersectan. Si se usa arco y recta o dos arcos pueden existir dos intersecciones y el punto NEAR en las dos últimas declaraciones indica un punto conocido cercano a la intersección deseada. El ingeniero piensa todo esto como una sola declaración, pero el programador piensa en tres



declaraciones distintas y visualiza una estructura como la mostrada, - -  
donde se tengan todas las combinaciones posibles.



#### 6. Lenguaje de definición de declaraciones

Existe en el sistema ICES el lenguaje de definición de declaraciones -  
CDL, con el cual el programador establece las declaraciones de los sub-  
sistemas y genera tablas de declaraciones. Es un lenguaje orientado ha-  
cia el que desarrolla subsistemas y por eso se dice que ICES incluye un  
lenguaje orientado que genera lenguajes orientados a problemas. El len-  
guaje de definición de declaraciones se puede usar para añadir, modificar  
o borrar declaraciones existentes en un subsistema.

Supóngase que en el subsistema COGO se desea añadir la declaración - -  
STORE para almacenar las coordenadas de un punto dado. Un ingeniero  
escribiría esta declaración en la forma

```
STORE POINT 5 X 110.30 Y 215.5
```

Para añadir esta declaración en el vocabulario o en la gramática de - -  
COGO se escribe el programa de definición o declaración siguiente:

```
SYSTEM 'COGO'
ADD 'STORE'
ID 'P' INTEGER 'N POINT' REQUIRED
ID 'X' REAL 'X COORD' STANDARD 0
ID 'Y' REAL 'Y COORD' STANDARD 0
EXECUTE 'STORE'
FILE
```

Si se deseara el almacenamiento del dato usando el idioma castellano se  
cambiaría STORE por ALMAC. Las palabras subrayadas pertenecen -  
al vocabulario del lenguaje de definición de declaraciones.

La cantidad P es un entero y se necesita, por lo que si se olvidara po-  
ner el número 5, el diagnóstico indicaría que falta ese punto porque se  
ha definido como "requerido" o existente. Si se escribiera 5.0 indica-  
ría error porque se ha definido como cantidad entera.

La X y Y son cantidades reales y se van a almacenar bajo esta denomi-  
nación; pueden ser no requeridas y si no se escriben se suponen igual a  
cero.

EXECUTE indica que se ejecute la instrucción y FILE que se archive.

#### 7. Organización de datos

En cuanto a la organización y localización de datos y programas hay pro-  
gramas del sistema de organización de datos que controlan los archivos

que están en memoria secundaria y que están asociados con la solución del problema. Esto es importante porque estos programas permiten que el mismo arreglo sea usado por distintos subsistemas. Para esto se necesita un archivo de nombres de arreglos y variables que pueden tener distinta denominación o identificación para distintos subsistemas. Cada subsistema puede hacer referencia a él con nombre distinto, por lo que se necesita un control para llegar al mismo arreglo de datos. También se puede operar fácilmente con estos archivos codificándolos, solicitando su impresión o borrándolos. Además, lo anterior permite que distintos ingenieros trabajando en distintos problemas tengan acceso a los mismos datos.

ICES funciona desde fines de 1965 y desde entonces se ha venido mejorando, pudiéndose hacer uso de graficadoras, pantallas, osciloscopios, etc., para hacer más flexible y útil el sistema. ICES se ha desarrollado en forma de módulos de manera que muchos de ellos se puedan cambiar, borrar, etc.

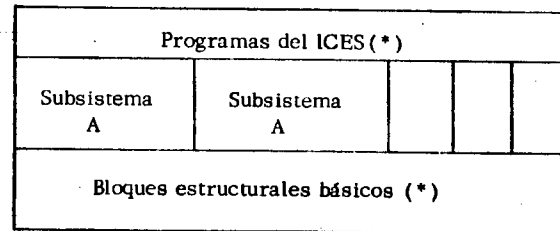
Existe también un supervisor del ICES que coordina a todos los programas del sistema y supervisa el uso de la computadora.

Para el desarrollo del sistema se ha tenido en cuenta que se usa bastante tiempo de máquina, pero se ha preferido minimizarlo permitiendo resolver totalmente el problema y proporcionar todas las alternativas para la solución de un problema. Es el caso de la memoria dinámica que utiliza más tiempo de máquina pero permite un estudio completo del proble-

ma al aumentar la capacidad disponible en la memoria en la solución de problemas.

#### 8. Resumen del Sistema ICES

El sistema ICES total puede presentarse en forma resumida como se indica en la figura siguiente:



(\*) aplicados a todo el sistema.

El sistema ICES es un marco de referencia para que los ingenieros puedan desarrollar un trabajo personal y con esta experiencia ampliar la capacidad del ICES.

ICES implica varias relaciones o asociaciones: entre máquinas de tercera generación y los programas del sistema; entre el ICETRAN y los programadores, entre el lenguaje de definición de declaraciones y el ingeniero, y además, entre el sistema integrado de computación y la organización de ingenieros que trabajan en el sistema.

El ICES fue financiado hasta 1970 por nueve compañías:

Fundación Ford

Compañía I. B. M.

M. I. T. Lincoln Laboratories

Massachusetts Bay Transportation Authority

McDonnell Automation Company

Departamento de Obras Públicas de Massachusetts

National Science Foundation

Portland Cement Association

Sociedad de Transportes del Gobierno de Estados Unidos.

#### 9. Subsistemas del ICES

A continuación se presenta un resumen de los distintos subsistemas del ICES.

##### 9.1 Subsistema STRUDL (STRUctural Design Language)

El subsistema STRUDL está diseñado para auxiliar al ingeniero en el proceso de diseño. Se considera una amplia gama de tipos de estructuras de 2 y 3 dimensiones, considerándose armaduras, marcos y elemento finito. Se pueden combinar estos componentes para problemas de análisis y diseño.

STRUDL se diseñó de manera que el usuario especifique datos y procedimientos de operación de estos mismos. La información de la estructura se puede almacenar, recuperar y manejar por medio de declaraciones. Los datos pueden arreglarse de manera que sólo parte de la información que se utiliza, pueda obtenerse en cualquier instante.

STRUDL II se aplica tanto a estructuras porticadas como a medio continuo (aplicación del elemento finito). Para estructuras porticadas los datos pueden especificar miembros, nudos y condiciones de carga nombrándolos arbitrariamente. Los datos se pueden modificar continuamente en cualquier parte del problema.

Las posibilidades de análisis incluyen análisis indeterminado o hiperestático, análisis preliminar suponiendo sistemas de carga y un análisis isostático que no requiere propiedades de los componentes.

Los resultados pueden consistir en fuerzas y deformaciones de cualquier parte de la estructura, fuerzas y esfuerzos en los miembros, y combinaciones como máximos esfuerzos, fuerzas, envolventes de esfuerzos, etc.

Se pueden combinar distintas condiciones de carga.

El problema en sí pueden almacenarse en discos y volver a utilizarse posteriormente. Se puede especificar una gran variedad de problemas y al especificar los miembros hay que incluir propiedades, identificar los extremos de los miembros, y las fuerzas que actúan en el miembro. Se pueden considerar miembros prismáticos o de sección variable, hacer referencia a una tabla generada en el subsistema TABLE, y especificar matrices de rigidez y de flexibilidad.

La información de nudos incluye la geometría, condiciones de apoyo,

y fuerzas que actúan sobre ellos. Pueden considerarse cargas uniformes, lineales o concentradas, variaciones de temperatura, y deformaciones de nudos (por asentamientos y por temperatura).

Los problemas de mecánica del medio continuo se resuelven con el método del elemento finito.

Se pueden considerar elementos de diversas formas.

En la situación actual del STRUDL II, se pueden considerar casos de esfuerzo plano, deformación plana, flexión de placas y análisis de cascarones.

Para el diseño de miembros, se incluyen las posibilidades de acero y concreto, de manera que los resultados puedan ser las dimensiones de miembros y el refuerzo longitudinal. Se incluye la posibilidad de dar el tamaño del miembro y verificar el diseño, por flexión, cortante, fricción, traslape y adherencia. El estudio se puede ligar con el manual de diseño de elementos metálicos.

También se tiene análisis no lineal de marcos, placas o cascarones.

Otras características incluyen: cargas de pandeo, análisis dinámico y optimización de marcos (en etapa experimental las dos últimas).

Finalmente, debe aclararse que STRUDL I está contenido totalmente dentro del subsistema STRUDL II.

## 9.2 Subsistema TABLE

Es un sistema usado para crear y manipular datos tabulados. Se diseñó para dar flexibilidad y especificar datos que puedan utilizarse en cualquier otro subsistema de ICES.

Produce archivos en disco, en arreglos identificados de manera que puede transmitirse información tabulada. Se pueden dar datos de entrada, actualizar, modificar y arreglar los datos tabulados en arreglos de dos dimensiones. Se pueden especificar arbitrariamente número de renglones y de columnas.

Las unidades de dimensión se pueden especificar para cada columna de manera que pueda hacerse la conversión a las mismas unidades. En cualquier instante pueden especificarse nuevos renglones y columnas y de igual forma se pueden borrar o cambiar.

Se pueden dar nombres a los órdenes utilizados en sucesiones de propiedades de la tabla, muy útil para el diseño estructural. Por ejemplo en la sucesión de intentos de diseño, el orden representa un valor que disminuye según el mérito del diseño.

Se puede poner un límite en cuanto al diseño y con qué precisión se puede lograr ese diseño, a través de dar indicadores de cómo proceder al diseño y cuántas veces rediseñar.

TABLE II es un poco más flexible pero en esencia es el mismo subsis -

tema que TABLE. En TABLE II se hacen cambios para hacerlo compatible con otros subsistemas.

TABLE I es compatible con STRUDL I y II, pero TABLE II no es compatible con ninguno.

### 9.3 Subsistema COGO (COordinated GeOMetry)

Es un subsistema utilizado para la descripción y procesamiento de problemas de diseño geométricos que incluyen diseños topográficos, de carreteras, intersecciones, pasos a desnivel, etc.

Opera en términos geométricos y variables que pueden definirse, almacenarse, recuperarse, calcularse, combinarse y manejarse por medio del lenguaje orientado al problema correspondiente.

Se pueden manipular puntos, rectas, curvas, distancias, direcciones, ángulos, alineamientos, perfiles, etc.

Se puede tener un máximo de 10 000 puntos, 1 000 rectas, 1000 curvas, etc. Es poderoso el sistema en cuanto a capacidad; todas las subrutinas son de precisión doble.

Se proporciona una verificación de errores muy extensa.

El sistema COGO se basa parcialmente en sistemas COGO previos, pero es un nuevo lenguaje más poderoso y completo, utilizando capacidades del sistema integrado y la tercera generación de máquinas.

### 9.4 Subsistema SEPOL (SETtlement Problem Oriented Language)

Es un subsistema para calcular esfuerzos y deformaciones en suelos, debido a cargas en cimentaciones. También calcula la magnitud y progreso de asentamientos de cimentaciones. Por lo tanto, se pueden calcular asentamientos para el diseño de edificios, presas, muros de contención y cualquiera otra carga que se distribuya en la superficie del suelo.

Se consideran cuatro áreas o funciones que son:

- 1) IN SITU. Es una función que describe el sistema del suelo y su estado natural antes de la construcción.
- 2) STRESS DISTRIBUTION. Función que describe la configuración del sistema de carga que permite el cálculo de esfuerzos y deformaciones producidos por la carga.
- 3) SETTLEMENT. Es una función que conduce al cálculo del asentamiento en cualquier punto de la superficie.
- 4) RATE. Función que conduce al cálculo del progreso del asentamiento y disipación de exceso de presión.

Las propiedades del suelo se pueden especificar en la forma en la que usualmente se obtienen de laboratorio y se pueden modificar durante el cálculo.

Se puede especificar cualquier número de capas de tierra con características diferentes.

El asentamiento se puede calcular para deformaciones iniciales consolidadas para cualquiera de las capas de tierra.

El asentamiento en función del tiempo se puede calcular para un sistema de capas con distintas permeabilidades y comprensibilidades y también con variación de carga en el tiempo (con base en la Teoría de Terzaghi.)

#### 9.5 Subsistema R O A D S (R Oadway Analysis and Design System)

Este subsistema se usa en la solución de problemas que implican la localización y diseño de casi cualquier tipo de camino, aunque fundamentalmente se pretendía utilizarlo para carreteras.

El marco de referencia básico es aplicable a varios problemas de ingeniería como excavación y relleno de materiales, para vía de ferrocarriles, canales y diques. Incluye una clasificación de los materiales superficiales e internos, geometría de perfiles y alineamientos, integrado con el subsistema COGO.

Permite la simulación de vehículos para determinar el efecto de la geometría de la carretera y del tráfico.

Se puede modificar la sección transversal del diseño de una carretera, los volúmenes de material cambiando a nuevas secciones, etc.

Se tiene la capacidad de guardar en memoria secundaria información tabulada de los perfiles y alineamientos.

Se puede hacer transferencia de datos entre los subsistemas ROADS y COGO de manera que los datos geométricos se puedan utilizar en ambos subsistemas.

#### 9.6 Subsistema T R A N S E T (TRANSPORTATION NETWORK ANALYSIS)

Permite un método conveniente y flexible para producción y análisis de flujos que ocurren en las redes de transporte.

Muchos problemas del proceso de planeación pueden auxiliarse con este subsistema

Se ha diseñado para representar carreteras o transporte masivo. Sin embargo puede utilizarse para otras redes como tuberías o redes eléctricas.

Se tienen dos grupos de posibilidades de aplicación:

- 1) En la planeación de transportación donde se pueden calcular trayectorias mínimas y asignar viajes entre distintas zonas.
- 2) Capacidad para la manipulación de archivos (en el análisis de redes se tienen que formar bastantes archivos).

Se pueden almacenar redes de transportación de gran tamaño en memoria secundaria y modificarse o pedir resúmenes de esos datos.

No hay limitación del tamaño de los problemas aunque si se usa mucha memoria secundaria, se aumenta el tiempo de máquina. La progra -

mación más que el tamaño de la máquina, limita el tamaño de las redes hasta 8,191 uniones (máximo número de uniones).

Se basa en programas ya conocidos de ruta crítica e investigación de tránsito, desarrollados en el M. I. T.

#### 9.7 Subsistema BRIDGE (BRIDGE Design System)

Permite que el ingeniero describa las etapas siguientes del proceso de diseño de puentes: el alineamiento horizontal, el alineamiento vertical y la sección transversal del puente propuesto.

Se pueden determinar y mostrar distintas posibilidades de arreglos claros combinando los requerimientos del puente con la intersección y descripción geométrica del pavimento o del terreno. Se efectúa el diseño o análisis de una losa de concreto con banqueta apoyada en un sistema reticular. El ingeniero puede proporcionar limitaciones en el diseño de la losa.

Se puede mostrar la geometría del puente, la configuración de la carga, el diseño de la losa y los esfuerzos de diseño. Se realiza el diseño preliminar de trabes simples o continuas con momento de inercia variable. Se pueden mostrar los valores de diseño de momento, fuerza cortante y reacciones.

El proyecto de diseño de puentes puede requerir hacer referencia a una o más de las etapas previas. La secuencia normal es como se ha indica

do en donde cada etapa después de la primera depende mucho de la información generada en la etapa previa.

El sistema de diseño de puentes permite que el ingeniero participe y dirija la solución del problema del proyecto del puente.

El formato y terminología que utiliza el ingeniero es el mismo que emplea al comunicarse con un colega conocedor de proyectos de puentes.

#### 9.8 Subsistema PROJECT (PROJECT Engineering Control)

Es un sistema de computadora creado para ayudar en la planeación y control de proyectos.

Cualquier trabajo que pueda representarse por una red de elementos y de interdependencias asociadas se puede procesar con PROJECT. El significado de las proposiciones es fácilmente entendible para cualquier usuario familiarizado con las técnicas de administración u organización de proyectos.

La parte más importante de PROJECT es que es un sistema de análisis de redes con el método de la ruta crítica, capaz de procesar tres tipos de redes: actividades en los nodos, actividades en las flechas y eventos orientados. El usuario puede especificar restricciones externas de tiempo, un punto de referencia del calendario y fechas no laborables que deban usarse en el cálculo del calendario de ruta crítica del proyecto. Cuando sea necesario, el usuario puede hacer referencia a días de calendario o a días de trabajo.

Permite el procesamiento y almacenamiento de cualquier número de redes independientes durante una sola corrida de máquina.

Una vez que se ha introducido una red, la información permanece en los discos como una red activa hasta que el usuario la borra. En cualquier instante se pueden especificar adiciones, sustracciones y modificaciones en los datos de cualquier red activa. El usuario puede pedir cualquier subarreglo de los datos calculados de la red que desee.

Los proyectos se pueden subdividir en grupos únicos o de traslape.

Algunos calendarios se pueden solicitar en forma de diagrama de barras además de la forma tabulada normal.

Aunque fundamentalmente es un sistema de calendarización, planeación o programación, se han incorporado varias capacidades financieras y de recursos de manera que el usuario pueda asignar costos a las actividades.

En PROJECT I se ha desarrollado un nuevo método conocido como sucesiones de pendientes repetitivas para el estudio de la lógica que normalmente se representa como actividades de la red cuando se consideran globalmente.

#### 9.9 Subsistema OPTTECH (OPTimization TECHniques System)

Se ha diseñado para resolver una variedad de problemas de optimización que ocurren en el diseño y análisis ingenieril y en la operación de elementos ingenieriles para dar una introducción a técnicas con computadora para resolver problemas de programación matemática.

Entre las áreas de aplicación se incluyen sistemas de aprovisionamiento de aguas, problemas de flujo en redes, en los campos hidráulico y de transporte, problemas de programación, en la administración de construcciones y en la operación de unidades de transporte y también en algunos aspectos del diseño estructural.

OPTTECH tiene cuatro características principales:

- 1) Programación lineal.
- 2) Optimización de flujos en redes.
- 3) Programación de búsqueda directa no lineal para problemas con restricciones y sin ellas.
- 4) Programación lineal entera

OPTTECH no proporciona archivos de datos para el usuario.

Algunos aspectos de las cuatro características son:

La programación lineal usa una técnica de solución basada en el método Simplex revisado, con selección de pivote compuesto (lo que permite variables reales negativas en la base de principio) y una característica de generación automática de holguras.

El flujo en redes usa el método generalizado primal-dual en la solución de cuatro clases de problemas: análisis, factibilidad o posibilidad, minimización de costos y síntesis de flujos independientes del tiempo. El



número de arcos o flechas se limita a 1,050 y el número de nodos a 400.

En búsqueda directa se usa la técnica desarrollada por Flood y León. Requiere una subrutina para evaluar el objetivo, en cada punto de la búsqueda. Usualmente es necesaria la intervención del usuario durante una serie de intentos de solución.

En programación entera se usa un algoritmo general primal (primario o de una alternativa), para la solución de problemas de programación entera. Sin embargo el costo resulta relativamente ineficiente comparándolo con otros algoritmos por lo que esta característica es esencialmente de interés experimental. Se proporcionan opciones para reiniciar intentos de solución que se han interrumpido.

#### 9.10 Subsistema LEASE ó SLOPE (SLOPE Stability System)

Es un subsistema orientado al problema de evaluación de la estabilidad de taludes y muros de contención.

La primera versión del sistema encuentra el factor de seguridad de taludes, de geometría arbitraria, formados de tantas clases de suelos como se desee. Se pueden efectuar análisis con drenes o sin ellos o una combinación de ambos.

El factor de seguridad se calcula para superficies de falla circulares cilíndricas, utilizando el método simplificado de Bishop y el método sueco o de Fellenius, modificado por Hvorslev.

El usuario puede solicitar soluciones para círculos específicos o puede dejar que la computadora busque el factor de seguridad mínimo. Esto se efectúa para cada centro de falla tentativo, variando el radio. Este centro se puede desplazar en una red o malla prescrita o la computadora puede encontrar un mínimo relativo con un procedimiento de búsqueda automática.

Se está trabajando en una nueva versión que incluya el método general de estabilidad de Morgenstern para superficies de falla arbitraria.

Posiblemente se incluye un método de elemento finito para resolver problemas de flujo de fluidos.

#### 9.11 Subsistema TRAVOL (Traffic VOLUME Data Subsystem)

Es un subsistema para el procesamiento, almacenamiento y aplicación de datos de volumen de tráfico con los fines de investigación y planeación de transporte en áreas urbanas o regionales.

Está diseñado como complemento de programas continuos o a corto plazo, de conteo de tráfico en donde se deban procesar rápida y eficientemente grandes cantidades de datos.

Consiste de registros de datos para tráfico, rutinas de análisis estadístico y un lenguaje orientado para usar archivos y subrutinas. Datos de tráfico, horarios y diarios, se pueden almacenar en memoria secundaria.

25.

El subsistema incluye un conjunto de rutinas para establecer un programa que estime el tránsito diario promedio a partir de pequeños muestreos.

Se pueden incorporar fácilmente extensiones a las características de análisis del subsistema.

*notas sobre el sistema  
integrado de ingeniería civil*

**subsistema  
ICES-STRUDL**

*Dr. Rodolfo Luths G.*

En el presente resumen se indican las declaraciones de STRUDL en forma compacta.

En primer lugar se presenta el formato del elemento "lista", que es

$$* \text{ lista} = \left\{ \begin{array}{l} \text{lista alfanumérica} \\ \text{lista de enteros} \\ n_1 \left\{ \begin{array}{l} \text{TO} \\ \text{THROUGH} \end{array} \right\} n_2 \end{array} \right\}$$

en donde

lista alfanumérica = 'a<sub>1</sub>' ('a<sub>2</sub>') . . .

lista de enteros = i<sub>1</sub> (i<sub>2</sub>) . . .

El modo de modificación de los datos de entrada, identificando la operación que ejecutará en los datos subsiguientes, se especifica con las declaraciones:

ADDITIONS  
CHANGES  
DELETIONS

Se tienen las declaraciones siguientes:

1. Declaración STRUDL

STRUDL ( 'a<sub>1</sub>' ('título')  
RESTORE 'a<sub>1</sub>' )

2. Declaración SAVE

SAVE ('a<sub>1</sub>')

3. Declaración CHANGE ID

CHANGE ID 'a<sub>1</sub>' ('título')

4. Declaración UNITS

\* UNITS {  
unidad de longitud  
unidad de fuerza  
unidad angular  
unidad de temperatura  
unidad de tiempo

unidad de longitud = {  
INCHES  
FEET  
FT  
CENTIMETERS  
CMS  
METERS  
M

unidad de fuerza = {  
POUNDS  
LBS  
KIPS  
TONS  
MTONS  
KILOGRAMS  
KGS

unidad angular = {  
RADIANS  
DEGREES

unidad de temperatura = {  
FAHRENHEIT  
CENTIGRADE

unidad de tiempo = {  
SECONDS  
MINUTES  
HOURS

5. Declaración TYPE

TYPE {  
PLANE {  
TRUSS {XY}  
FRAME {XZ}  
GRID {YZ}  
SPACE {  
TRUSS  
FRAME

6. Declaración SET ELEMENTS

SET ELEMENTS {  
INTEGER  
UNRESTRICTED

7. Declaración JOINT COORDINATES

Forma tabulada:

JOINT COORDINATES

{  
i<sub>1</sub>  
' a<sub>1</sub>' } ( [XCOORD] v<sub>x</sub> [YCOORD] v<sub>y</sub> [ZCOORD] v<sub>z</sub> ) {  
FREE  
SUPPORT

Forma individual:

JOINT {  
i<sub>1</sub>  
' a<sub>1</sub>' }

COORDINATES ( [XCOORD] v<sub>x</sub> [YCOORD] v<sub>y</sub> [ZCOORD] v<sub>z</sub> ) {  
FREE  
SUPPORT

8. Declaración JOINT RELEASES

Forma tabulada:

JOINT RELEASES ( {  
eliminar fuerzas  
eliminar momentos } ) ( [TH1] v<sub>1</sub> [TH2] v<sub>2</sub> [TH3] v<sub>3</sub> )

$$\text{lista} \left( \begin{array}{l} \text{eliminar fuerzas} \\ \text{eliminar momentos} \end{array} \right) \left( \begin{array}{l} [TH1] v_1 \\ [TH2] v_2 \\ [TH3] v_3 \end{array} \right)$$

Forma individual:

$$\text{JOINT lista RELEASES} \left( \begin{array}{l} \text{eliminar fuerzas} \\ \text{eliminar momentos} \end{array} \right) \left( \begin{array}{l} [TH1] v_1 \\ [TH2] v_2 \\ [TH3] v_3 \end{array} \right)$$

$$\text{eliminar fuerzas} = \text{FORCE} \begin{array}{l} X \\ Y \\ Z \end{array}$$

$$\text{eliminar momentos} = \text{MOMENT} \begin{array}{l} X \\ Y \\ Z \end{array}$$

9. Declaración MEMBER INCIDENCES

Forma tabulada:

MEMBER INCIDENCES

$$\left\{ \begin{array}{l} i_1 \\ a_1 \end{array} \right\} \text{ nudo inicial, nudo final}$$

Forma individual:

$$\text{MEMBER} \left\{ \begin{array}{l} i \\ a_1 \end{array} \right\} \text{ GOES (FROM) nudo inicial (TO) nudo final}$$

10. Declaración MEMBER RELEASES

Forma tabulada:

$$\text{MEMBER RELEASES} \left( \begin{array}{l} \text{START} \\ \text{END} \end{array} \right) \left( \begin{array}{l} \text{eliminar fuerzas} \\ \text{eliminar momentos} \\ \text{eliminar fuerzas} \\ \text{eliminar momentos} \end{array} \right)$$

$$\text{lista} \left( \begin{array}{l} \text{START} \\ \text{END} \end{array} \right) \left( \begin{array}{l} \text{eliminar fuerzas} \\ \text{eliminar momentos} \\ \text{eliminar fuerzas} \\ \text{eliminar momentos} \end{array} \right)$$

Forma Individual:

$$\text{MEMBER lista RELEASES} \left( \begin{array}{l} \text{START} \\ \text{END} \end{array} \right) \left( \begin{array}{l} \text{eliminar fuerzas} \\ \text{eliminar momentos} \\ \text{eliminar fuerzas} \\ \text{eliminar momentos} \end{array} \right)$$

$$\text{eliminar fuerzas} = \text{FORCE} \begin{array}{l} X \\ Y \\ Z \end{array}$$

$$\text{eliminar momentos} = \text{MOMENT} \begin{array}{l} X \\ Y \\ Z \end{array}$$

11. Declaración MEMBER PROPERTIES

Forma tabulada:

MEMBER PROPERTIES ( { PRISMATIC, (valores de la sección)  
 VARIABLE  
 TABLE 'nombre de la tabla' ('nombre de la sección')  
 FLEXIBILITY  
 STIFFNESS } )

lista ( (PRISMATIC), valores de la sección  
 especificaciones de variables  
 especificaciones de tablas  
 (FLEXIBILITY), especific. matricial  
 (STIFFNESS), especific. matricial )

Valores de sección =  $\begin{Bmatrix} [AX] v_1 & [AY] v_2 & [AZ] v_3 & [IX] v_4 & [IY] v_5 & [IZ] v_6 \\ [SY] v_7 & [SZ] v_8 \end{Bmatrix}$

especificaciones de variables = (VARIABLE)

SEGMENTS  $i_1$  (AND  $i_2$ ) { TABLE 'nombre de la tabla' 'nombre de la sección' { LENGTH  
 valores de sección { XC  
 YC  
 ZC } }  $v_1$

especificación matricial =

(MATRIX) COLUMNS ( $i_1$ ) ( $i_2$ ) ( $i_3$ ) ( $i_4$ ) ( $i_5$ ) ( $i_6$ )

{ ROW 1  $v_{11}$   $v_{12}$   $v_{13}$   $v_{14}$   $v_{15}$   $v_{16}$   
 :  
 ROW 6  $v_{61}$   $v_{62}$   $v_{63}$   $v_{64}$   $v_{65}$   $v_{66}$  }

Forma individual:

MEMBER lista PROPERTIES { PRISMATIC, valores de sección  
 especificaciones de variables  
 especificaciones de tablas  
 FLEXIBILITY, especific. matricial  
 STIFFNESS, especific. matricial }

12. Declaración MEMBER DEPTH

MEMBER DEPTH ( { PRISMATIC (altura)  
 VARIABLE } )

lista { (PRISMATIC) altura  
 especificaciones de variables }

especificaciones de variables = (VARIABLE)  
 SEGMENT  $i_1$  (AND  $i_2$ ) alturas

Forma individual:

MEMBER lista DEPTH { PRISMATIC alturas  
 especificaciones de variables }

13. Declaración CONSTANTS

CONSTANTS (descripción constante).

descripción constante

$$\text{descripción constante} = \begin{cases} \underline{E} \\ \underline{G} \\ \underline{CTE} \\ \underline{DENSITY} \\ \underline{BETA} \end{cases} \left\{ \begin{array}{l} v_1 \text{ lista, } v_2 \text{ lista, } v_3 \text{ lista } \dots \\ v_1 \underline{ALL} \\ v_1 \underline{ALL BUT} v_2 \text{ lista, } v_3 \text{ lista.} \end{array} \right.$$

14. Declaración LOADING

$$\underline{LOADING} \left\{ \begin{array}{l} i_1 \\ 'a_1' \end{array} \right\} \text{ (título)}$$

15. Declaración LOADING COMBINATION

$$\underline{LOADING COMBINATION} \left\{ \begin{array}{l} i_1 \\ 'a_1' \end{array} \right\} \text{ (título) (COMBINE especific. de carga)}$$

$$\text{especificaciones de carga} = \left\{ \begin{array}{l} i_2 \\ 'a_2' \end{array} \right\} v_2 \left( \left\{ \begin{array}{l} i_3 \\ 'a_3' \end{array} \right\} v_3 \right) \dots$$

16. Declaración JOINT LOADS

Forma tabulada:

$$\underline{JOINT LOADS} \left\{ \begin{array}{l} \underline{FORCE} \left[ \begin{array}{l} X \\ Y \\ Z \end{array} \right] v_1, \left[ \begin{array}{l} Y \\ X \\ Z \end{array} \right] v_2, \left[ \begin{array}{l} Z \\ X \\ Y \end{array} \right] v_3 \\ \underline{MOMENT} \left[ \begin{array}{l} X \\ Y \\ Z \end{array} \right] v_1, \left[ \begin{array}{l} Y \\ X \\ Z \end{array} \right] v_2, \left[ \begin{array}{l} Z \\ X \\ Y \end{array} \right] v_3 \end{array} \right\}$$

$$\text{lista} \left\{ \begin{array}{l} \underline{FORCE} \left[ \begin{array}{l} X \\ Y \\ Z \end{array} \right] v_1, \left[ \begin{array}{l} Y \\ X \\ Z \end{array} \right] v_2, \left[ \begin{array}{l} Z \\ X \\ Y \end{array} \right] v_3 \\ \underline{MOMENT} \left[ \begin{array}{l} X \\ Y \\ Z \end{array} \right] v_1, \left[ \begin{array}{l} Y \\ X \\ Z \end{array} \right] v_2, \left[ \begin{array}{l} Z \\ X \\ Y \end{array} \right] v_3 \end{array} \right\}$$

Forma individual:

$$\underline{JOINT \text{ lista } LOADS} \left\{ \begin{array}{l} \underline{FORCE} \left[ \begin{array}{l} X \\ Y \\ Z \end{array} \right] v_1, \left[ \begin{array}{l} Y \\ X \\ Z \end{array} \right] v_2, \left[ \begin{array}{l} Z \\ X \\ Y \end{array} \right] v_3 \\ \underline{MOMENT} \left[ \begin{array}{l} X \\ Y \\ Z \end{array} \right] v_1, \left[ \begin{array}{l} Y \\ X \\ Z \end{array} \right] v_2, \left[ \begin{array}{l} Z \\ X \\ Y \end{array} \right] v_3 \end{array} \right\}$$

17. Declaración JOINT DISPLACEMENTS

Forma tabulada:

$$\underline{JOINT DISPLACEMENTS} \left\{ \begin{array}{l} \underline{DISPLACEMENTS} \left[ \begin{array}{l} X \\ Y \\ Z \end{array} \right] v_1, \left[ \begin{array}{l} Y \\ X \\ Z \end{array} \right] v_2, \left[ \begin{array}{l} Z \\ X \\ Y \end{array} \right] v_3 \\ \underline{ROTATIONS} \left[ \begin{array}{l} X \\ Y \\ Z \end{array} \right] v_1, \left[ \begin{array}{l} Y \\ X \\ Z \end{array} \right] v_2, \left[ \begin{array}{l} Z \\ X \\ Y \end{array} \right] v_3 \end{array} \right\}$$

$$\text{lista} \left\{ \begin{array}{l} \underline{DISPLACEMENTS} \left[ \begin{array}{l} X \\ Y \\ Z \end{array} \right] v_1, \left[ \begin{array}{l} Y \\ X \\ Z \end{array} \right] v_2, \left[ \begin{array}{l} Z \\ X \\ Y \end{array} \right] v_3 \\ \underline{ROTATIONS} \left[ \begin{array}{l} X \\ Y \\ Z \end{array} \right] v_1, \left[ \begin{array}{l} Y \\ X \\ Z \end{array} \right] v_2, \left[ \begin{array}{l} Z \\ X \\ Y \end{array} \right] v_3 \end{array} \right\}$$

Forma individual:

$$\underline{JOINT \text{ lista } DISPLACEMENTS} \left\{ \begin{array}{l} \underline{DISPLACEMENTS} \left[ \begin{array}{l} X \\ Y \\ Z \end{array} \right] v_1, \left[ \begin{array}{l} Y \\ X \\ Z \end{array} \right] v_2, \left[ \begin{array}{l} Z \\ X \\ Y \end{array} \right] v_3 \\ \underline{ROTATIONS} \left[ \begin{array}{l} X \\ Y \\ Z \end{array} \right] v_1, \left[ \begin{array}{l} Y \\ X \\ Z \end{array} \right] v_2, \left[ \begin{array}{l} Z \\ X \\ Y \end{array} \right] v_3 \end{array} \right\}$$



18. Declaración MEMBER LOADS

Forma tabulada:

MEMBER LOADS, (especif. de dirección), (especif. de tipo)

Forma individual:

MEMBER lista LOADS, especific. de dirección, especific. de tipo

$$\text{especif. de dirección} \left\{ \begin{array}{l} \text{FORCE} \left( \begin{array}{c} X \\ Y \\ Z \end{array} \right) \\ \\ \text{MOMENT} \left( \begin{array}{c} X \\ Y \\ Z \end{array} \right) \end{array} \right\} \text{ (GLOBAL)}$$

$$\text{especif. de tipo} = \left\{ \begin{array}{l} \text{CONCENTRATED (FRACTIONAL)} [P] v_1, [L] v_2 \\ \text{UNIFORM (FRACTIONAL)} [W] v_3, ([LA] v_4, [LB] v_5) \\ \text{LINEAR (FRACTIONAL)} [WA] v_6, [WB] v_7, ([LA] v_8, [LB] v_9) \end{array} \right\}$$

19. Declaración MEMBER TEMPERATURE LOADS

Forma tabulada:

MEMBER TEMPERATURE (LOADS) (especif. de temperatura)

lista, (especif. de temperatura)

$$\text{especif. de temperatura} = (\text{FRACTIONAL}) ([LA] v_1, [LB] v_2) \left\{ \begin{array}{l} [AXIAL] v_3 \\ \\ \text{(BENDING)} [Y] \\ v_4, [Z] v_5 \end{array} \right\}$$

Forma individual

MEMBER lista TEMPERATURE (LOADS) especific. de temperatura

20. Declaración MEMBER DISTORTIONS

Forma tabulada:

MEMBER DISTORTIONS (tipos y lugar)

lista (tipos y lugar)

$$\text{tipos y lugar} = \left\{ \begin{array}{l} \text{CONCENTRATED (FRACTIONAL)} [L] v_1 \\ \text{UNIFORM (FRACTIONAL)} [LA] v_2 [LB] v_3 \end{array} \right\} \text{ datos de distancias}$$

$$\text{datos de distancias} = \left\{ \begin{array}{l} \text{DISPLACEMENT} [X] v_4, [Y] v_5, [Z] v_6 \\ \text{ROTATION} [X] v_4, [Y] v_5, [Z] v_6 \end{array} \right\}$$

Forma individual:

MEMBER lista DISTORTIONS tipos y lugar

21. Declaración MEMBER END LOADS

Forma tabulada:

$$\text{MEMBER END (LOADS)} \left( \begin{array}{l} \text{START} \\ \text{END} \end{array} \right) \left\{ \begin{array}{l} \text{FORCE} [X] v_1 [Y] v_2 [Z] v_3 \\ \text{MOMENT} [X] v_1 [Y] v_2 [Z] v_3 \end{array} \right\}$$

lista

$$\left( \begin{array}{l} \text{START} \\ \text{END} \end{array} \right) \left\{ \begin{array}{l} \text{FORCE} [X] v_1 [Y] v_2 [Z] v_3 \\ \text{MOMENT} [X] v_1 [Y] v_2 [Z] v_3 \end{array} \right\}$$

Forma individual:

$$\text{MEMBER lista END (LOADS)} \left( \begin{array}{l} \text{START} \\ \text{END} \end{array} \right) \left\{ \begin{array}{l} \text{FORCE} [X] v_1, [Y] v_2, [Z] v_3 \\ \text{MOMENT} [X] v_1, [Y] v_2, [Z] v_3 \end{array} \right\}$$

## 22. Declaración JOINT FORCES

Forma tabulada:

JOINT FORCES (especificaciones de fuerzas, (ángulos), carga)

lista (especificaciones de fuerzas, (ángulos), carga)

$$\text{especif. de fuerzas} = \left\{ \begin{array}{l} \underline{\text{FORCE}} \quad \underline{X} \quad \circ \quad \underline{\text{FX}} \\ \underline{\text{FORCE}} \quad \underline{Y} \quad \circ \quad \underline{\text{FY}} \\ \underline{\text{FORCE}} \quad \underline{Z} \quad \circ \quad \underline{\text{FZ}} \\ \underline{\text{MOMENT}} \quad \underline{X} \quad \circ \quad \underline{\text{MX}} \\ \underline{\text{MOMENT}} \quad \underline{Y} \quad \circ \quad \underline{\text{MY}} \\ \underline{\text{MOMENT}} \quad \underline{Z} \quad \circ \quad \underline{\text{MZ}} \end{array} \right\}$$

$$\text{especif. de ángulos} = \left[ \underline{\text{TH1}} \right] \theta_1 \left[ \underline{\text{TH2}} \right] \theta_2 \left[ \underline{\text{TH3}} \right] \theta_3$$

$$\text{especif. de carga} = \left\{ \begin{array}{l} v_1 \dots v_n \\ \text{'carga}_1 \text{' } v_1 \dots \text{'carga}_n \text{' } v_n \end{array} \right\}$$

Forma individual:

JOINT lista FORCES, especific. de fuerzas, (ángulos), carga

## 23. Declaración MEMBER FORCES

Forma tabulada:

MEMBER FORCES (especif. de fuerzas, distancias, carga)

lista, (especif. de fuerzas, distancias, carga)

$$\text{especif. de fuerzas} = \left\{ \begin{array}{l} \underline{\text{FORCE}} \quad \underline{X} \quad \circ \quad \underline{\text{FX}} \\ \underline{\text{FORCE}} \quad \underline{Y} \quad \circ \quad \underline{\text{FY}} \\ \underline{\text{FORCE}} \quad \underline{Z} \quad \circ \quad \underline{\text{FZ}} \\ \underline{\text{MOMENT}} \quad \underline{X} \quad \circ \quad \underline{\text{MX}} \\ \underline{\text{MOMENT}} \quad \underline{Y} \quad \circ \quad \underline{\text{MY}} \\ \underline{\text{MOMENT}} \quad \underline{Z} \quad \circ \quad \underline{\text{MZ}} \end{array} \right\}$$

$$\text{especif. de distancias} = \left\{ \begin{array}{l} \underline{\text{DISTANCE}} \quad v \\ \underline{\text{FRACTION}} \quad v \end{array} \right\}$$

$$\text{especif. de carga} = \left\{ \begin{array}{l} v_1 \dots v_n \\ \text{'carga}_1 \text{' } v_1 \dots \text{'carga}_n \text{' } v_n \end{array} \right\}$$

Forma individual:

MEMBER lista FORCES especific. de fuerzas, distancias, carga

## 24. Declaración LOADING LIST

$$\underline{\text{LOADING LIST}} \left\{ \begin{array}{l} \underline{\text{ALL}} \\ \underline{\text{ALL BUT}} \text{ lista} \\ \underline{\text{lista}} \end{array} \right\}$$

## 25. Declaración ACTIVE - INACTIVE

$$\left\{ \begin{array}{l} \underline{\text{ACTIVE}} \\ \underline{\text{INACTIVE}} \end{array} \right\} \left\{ \begin{array}{l} \underline{\text{MEMBERS}} \\ \underline{\text{JOINTS}} \end{array} \right\} \left\{ \begin{array}{l} \underline{\text{ALL}} \\ \underline{\text{ALL BUT}} \text{ lista} \\ \underline{\text{lista}} \end{array} \right\}$$

26. Declaración ANALYSIS

<u>STIFFNESS</u> <u>DETERMINATE</u> <u>PRELIMINARY</u>	( <u>ANALYSIS</u> )
--	---------------------

27. Declaración PRINT

PRINT 'tipo' (especif. de componentes)

tipo =	<u>DATA</u> <u>STRUCTURAL DATA</u> <u>LOADING DATA</u>	<u>LENGTH</u> <u>RELEASES</u> <u>CONSTANTS</u>
	<u>MEMBER</u>	<u>INCIDENCES</u> <u>PROPERTIES</u> <u>STATUS</u> <u>DEPTH</u>
	<u>JOINT</u>	<u>STATUS</u> <u>COORDINATES</u> <u>RELEASES</u>
	<u>APPLIED</u> <u>FORCE ASSUMPTIONS</u>	<u>MEMBER LOADS</u> <u>JOINT LOADS</u> <u>JOINT DISPLACEMENTS</u>

especif. de componentes =	<u>ALL</u> (activas e inactivas) (nudos y miembros)
	<u>JOINTS</u> lista
	<u>MEMBERS</u> lista

nudos y miembros =	<u>JOINTS</u>	( <u>AND</u> )	<u>MEMBERS</u>
	<u>MEMBERS</u>		<u>JOINTS</u>

activas e inactivas =	<u>ACTIVE</u>	( <u>AND</u> )	<u>INACTIVE</u>
	<u>INACTIVE</u>		<u>ACTIVE</u>

28. Declaración COMBINE

<u>COMBINE</u>	i  a	(especif. de carga)
----------------	------------	---------------------

29. Declaración OUTPUT DECIMAL

OUTPUT DECIMAL i

30. Declaración LIST

<u>LIST</u>	<u>FORCES</u>	} especific. de componentes
	<u>DISTORTIONS</u>	
	<u>LOADS</u>	
	<u>REACTIONS</u>	
	<u>DISPLACEMENTS</u>	

especif. de componentes =  $\left\{ \begin{array}{l} \underline{ALL} \text{ (activas e inactivas) (nudos y miembros)} \\ \underline{MEMBERS} \text{ lista} \\ \underline{JOINTS} \text{ lista} \end{array} \right\}$

activas e inactivas =  $\left\{ \begin{array}{l} \underline{ACTIVE} \\ \underline{INACTIVE} \end{array} \right\} \text{ (AND) } \left\{ \begin{array}{l} \underline{INACTIVE} \\ \underline{ACTIVE} \end{array} \right\}$

nudos y miembros =  $\left\{ \begin{array}{l} \underline{MEMBERS} \\ \underline{JOINTS} \end{array} \right\} \text{ (AND) } \left\{ \begin{array}{l} \underline{JOINTS} \\ \underline{MEMBERS} \end{array} \right\}$

resultado =

$\left\{ \begin{array}{l} \underline{SECTION FORCE} \\ \underline{SECTION STRESS} \\ \underline{FORCE ENVELOPE} \\ \underline{STRESS ENVELOPE} \\ \underline{MAXIMUM STRESS} \\ \underline{MAXIMUM STRESS EACH LOADING} \end{array} \right\}$

### 31. Declaración SECTION

SECTION especific. de sección

especif. de sección =  $(\underline{FRACTIONAL}) \left\{ \begin{array}{l} \underline{NS} \ i \ v_1 \dots v_i \\ \underline{DS} \ v_1 \ v_2 \ (\underline{NS} \ 1) \end{array} \right\}$

### 32. Declaración de resultados internos de miembros

$\underline{LIST}$  , resultado,  $\left\{ \begin{array}{l} \underline{ALL} \ (\underline{MEMBERS}) \\ \underline{MEMBER} \ \text{lista} \end{array} \right\} \text{ (SECTION especific. de sección)}$

*notas sobre el sistema  
integrado de ingeniería civil*

**subsistema**

**ICES-COGO**

ICES COGO

A continuación se presenta un resumen de las declaraciones utilizadas en el subsistema COGO.

1. Declaración SET SYSTEM

Permite establecer especificaciones del sistema y tiene la forma

SET (SYSTEM) especificaciones

en donde las especificaciones son:

NE ó XY  
NAZIMUTHS ó SAZIMUTHS  
AZIMUTHS ó BEARINGS  
DEC3 ó DEC2  
ASEC ó ADEC  
COMPUTE MODE ó CHECK MODE  
PRINT MODE ó NOPRINT MODE  
REDEFINE MODE ó NORDEFINE MODE

que indican:

NE, XY coordenadas horizontales, en entrada/salida, Norte - seguido de Este ó X seguida de Y, respectivamente.

NAZ, SAZ son los valores de azimut, en entrada/salida, medidas a partir del norte o del sur.

AZ, BEA son las direcciones impresas como azimut ó como - - orientación respectivamente.

DEC3, DEC2 indican las distancias y coordenadas considerando - - tres ó dos decimales.

ASEC, ADEC son ángulos y valores de dirección aproximados hasta segundos o redondeando con dos decimales, respectivamente.

COM, CHE indican entradas en tablas de datos o comparaciones - hechas con las entradas de tablas de datos.

PRI, NOP imprimen o no resultados intermedios después de cada declaración.

RED, NOR si se permite o no la redefinición de objetos.

2. Declaración SET CONSTANT

Esta declaración establece algunas constantes y tiene la forma

SET (CONSTANT)  $\left\{ \begin{array}{l} \text{DTOLERANCE} \quad v \\ \text{ATOLERANCE} \quad v \\ \text{MAXIMUM (ERRORS)} \quad v \end{array} \right\}$

en donde:

DTO es la tolerancia al verificar los valores numéricos de distancias y coordenadas, operando en el modo CHECK

de la declaración anterior. El valor de la tolerancia es v.

en donde

ATO

es la definición previa, pero aplicada a ángulos y direcciones, siendo v el valor de la tolerancia

'nombre archivo' indica el nombre del archivo cuyo contenido se almacenará en las tablas de datos.

MAX (ERR)

establece el número máximo de declaraciones con error, v, a partir de las cuales no se ejecutan más declaraciones. (las declaraciones de impresión no se incluyen).

Se intercalan los contenidos del archivo con los de las tablas de datos, teniendo prioridad los valores del archivo. El caso contrario ocurre cuando se usa el modo CHECK.

3. Declaración FILE SAVE

5. Declaración FILE DELETE

Se almacenan las tablas de datos como archivos permanentes en disco magnético. Tiene la forma

Con esta declaración se borran los archivos especificados.

Tiene la forma

FILE SAVE 'contraseña' 'nombre archivo'

FILE DELETE 'contraseña'

{ lista }  
{ ALL }

'contraseña' es el nombre, no excediendo ocho caracteres, que permite evitar que usuarios no autorizados generen archivos.

en donde

'nombre archivo' es el nombre asignado al archivo que se va a crear.

lista es la serie de nombres de los archivos que se van a borrar.

4. Declaración FILE RESTORE

ALL indica que todos los archivos se borran.

Tiene la forma

FILE RESTORE 'nombre archivo'

6. Declaración FILE LIST

Se listan los nombres de todos los archivos existentes en el conjunto de datos del usuario. Tiene la forma

FILE LIST

7. Declaración FILE PRINT

Se imprimen los archivos especificados. Tiene la forma

FILE PRINT  $\left\{ \begin{array}{l} \text{lista} \\ \text{ALL} \end{array} \right\}$

en donde los elementos son como se definieron en la declaración anterior.

8. Declaración PRINT ó DELETE

Se obtiene la impresión o se borran datos numéricos almacenados en tablas de datos. Tiene la forma

$\left\{ \begin{array}{l} \text{PRINT} \\ \text{DELETE} \end{array} \right\} \left\{ \begin{array}{l} \text{tipo lista} \\ \text{ALL tipos} \\ \text{ALL TABLES} \end{array} \right\}$

en donde

tipo =  $\left\{ \begin{array}{l} \text{POINTS} \\ \text{LINES} \\ \text{CURVES} \\ \text{AZIMUTHS} \\ \text{BEARINGS} \\ \text{ANGLES} \\ \text{DISTANCES} \\ \text{COORSES} \\ \text{CHAINS} \\ \text{PROFILES} \\ \text{TEXTS} \end{array} \right\}$

Con la primera alternativa -tipo- sólo se puede seleccionar un elemento. Se usa TABLES para incluir todos los tipos indicados. Cada uno de los elementos se explica en las declaraciones que se presentan a continuación.

Las declaraciones 9 a 11 se aplican a puntos.

9. Declaración STORE POINT

Se almacena la información de puntos. Tiene la forma

STORE (POINT) n  $\left\{ \begin{array}{l} \text{X} \ v_1 \ \text{Y} \ v_2 \ \text{STA} \ v_3 \ \text{Z} \ v_4 \\ \text{N} \ v_1 \ \text{E} \ v_2 \ \text{STA} \ v_3 \ \text{Z} \ v_4 \end{array} \right\}$

en donde

n es el número del punto que se almacena.

v<sub>1</sub>, v<sub>2</sub>, v<sub>3</sub>, v<sub>4</sub> son los valores numéricos de los elementos indicados.

Si un elemento se identifica, todos los que le siguen también tienen que identificarse.

10. Declaración PRINT ó DELETE POINTS

Sirve para imprimir o borrar la información sobre los puntos especificados. Tiene la forma



$\left. \begin{array}{l} \text{PRINT} \\ \text{DELETE} \end{array} \right\}$	$\left. \begin{array}{l} \text{POINTS} \\ \text{POINTS} \\ \text{ALL POINTS} \end{array} \right\}$	$p_i, p_j, p_k, \dots$
		$p_i, \text{ TO } p_j, (p_k \text{ TO } p_1), \dots$

en donde

$p_i, p_j, p_k$ , son números enteros que identifican a puntos.

### 11. Declaración EQUATE

Iguala valores numéricos de puntos almacenados con nuevos puntos. Tiene la forma

EQUATE  $n$  (TO)  $p_a$ , (impresión)

en donde

$n$  es el número del nuevo punto que se almacena.

$p_a$  es el número entero almacenado, cuyos valores numéricos de descripción se establecen en el punto  $n$ .

Las declaraciones 12 y 13 se aplican a distancias.

### 12. Declaración STORE DISTANCE

La declaración tiene la forma

STORE DISTANCE  $a$  distancia, (operador, modificador), (impresión)

en donde  $a$  es el nombre de la distancia que se almacena. Si se incluyen (operador, modificador), el valor de la distancia se modifica con el valor

del modificador para calcular el valor que debe almacenarse para el nombre  $a$  de la distancia. El operador puede ser: PLUS, MINUS, MULTIPLY (BY) ó DIVIDED (BY).

Si se desea calcular el radio de una curva, se usa

STORE DISTANCE a RADIUS of CURVE  $i$  at station, (impresión)

en donde se calcula y almacena el radio de la curva  $i$  para la estación dada. El radio de curva circular se almacena cuando se omite la estación y se trata de una curva circular con o sin espiral de transición.

Para calcular la distancia entre estaciones se tiene

STORE DISTANCE a STATION OF  $p_a$  MINUS STATION OF  $p_b$ , (impresión)

en donde  $a$  la estación del punto almacenado  $p_a$  se le resta la correspondiente del punto  $p_b$  y la diferencia en valor absoluto se almacena para el nombre  $a$  de la distancia. OF  $p_a$  y OF  $p_b$  pueden cambiarse por valores numéricos.

### 13. Declaración PRINT ó DELETE DISTANCES

Se imprimen o borran las magnitudes de las distancias especificadas. --

Tiene la forma

$\left. \begin{array}{l} \text{PRINT} \\ \text{DELETE} \end{array} \right\}$	$\left. \begin{array}{l} \text{DISTANCES} \\ \text{ALL DISTANCES} \end{array} \right\}$	$a, b, c, \dots$

En impresión, se tienen las variantes siguientes para la especificación de las distancias

$$\text{DISTANCES} \quad \left\{ \begin{array}{l} a, b, c, \dots \\ p_a \text{ TO } p_b, (p_c \text{ TO } p_d), \dots \\ p_a \text{ TO } p_b \text{ (TO } p_c \text{ TO } p_d \dots) \end{array} \right\}$$

Las declaraciones 14 a 16 se aplican a ángulos.

14. Declaración STORE ANGLE

Sirve para almacenar ángulos y tiene la forma

STORE ANGLE a ángulo (operador, modificador), (impresión)

en donde

a es el nombre del ángulo que se almacena. Los elementos restantes corresponden a los definidos en la declaración 12.

15. Declaración PRINT ó DELETE ANGLES

Se imprimen o borran las magnitudes de los ángulos especificados. Tiene la forma

$$\left\{ \begin{array}{l} \text{PRINT} \\ \text{DELETE} \end{array} \right\} \quad \left\{ \begin{array}{l} \text{ANGLES } a, b, c, \dots \\ \text{ALL ANGLES} \end{array} \right\}$$

16. Declaración PRINT ANGLE AT

En impresión también se puede calcular e imprimir un ángulo, usando la declaración

PRINT ANGLE AT p<sub>a</sub> from p<sub>b</sub> to p<sub>c</sub>, (AT p<sub>d</sub> from p<sub>c</sub> to p<sub>e</sub>) . . . .

Las declaraciones 17 a 20 se aplican a direcciones, pudiendo sustituirse la palabra AZIMUTH por la palabra BEARING.

17. Declaración STORE AZIMUTH

Tiene la forma

STORE AZIMUTH a dirección, (ROUND to v  $\left\{ \begin{array}{l} \text{MINUTES} \\ \text{SECONDS} \end{array} \right\}$ ) (impresión)

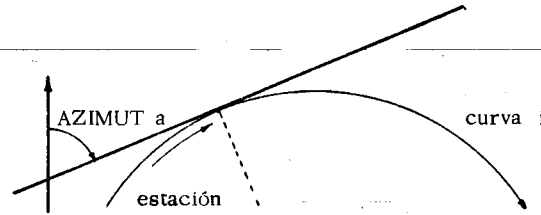
en donde

a es el nombre de la dirección que se almacena. El valor de la dirección se redondea a un número par de minutos o segundos antes de almacenarlo.

18. Declaración STORE AZIMUTH TANGENT

Se calcula y almacena la dirección de la tangente adelante del valor de la estación en la curva. Se imprimen mensajes para los casos en los que la estación se encuentra antes del principio de la curva, almacenándose la tangente hacia atrás y cuando la estación está después del fin de la curva, almacenándose la dirección de la tangente hacia adelante. Tiene la forma

STORE AZIMUTH a TANGENT to CURVE i at 'estación', (impresión)



19. Declaración PRINT ó DELETE AZIMUTHS

Se imprimen o borran las magnitudes especificadas. El valor de la dirección se imprime en grados, minutos y segundos. Tiene la forma

$\left\{ \begin{array}{l} \text{PRINT} \\ \text{DELETE} \end{array} \right\} \left\{ \begin{array}{l} \text{AZIMUTHS } a, b, c, \dots \\ \text{ALL AZIMUTHS} \end{array} \right\}$

20. Declaración PRINT AZIMUTHS TO

En impresión se puede calcular e imprimir una dirección, con

$\text{PRINT AZIMUTH} \left\{ \begin{array}{l} p_a \text{ TO } p_b, (\text{ROUND to } v \left\{ \begin{array}{l} \text{MINUTES} \\ \text{SECONDS} \end{array} \right\}) \\ p_a \text{ TO } p_b, (p_c \text{ TO } p_d), \dots \\ p_a \text{ TO } p_b (\text{TO } p_c \text{ TO } p_d \dots \\ \text{of LINE } i \end{array} \right\}$

Las declaraciones 21 a 24 se aplican a rectas.

21. Declaración STORE LINE

Esta declaración tiene la forma

$\text{STORE LINE } i \text{ thru } p_a \left\{ \begin{array}{l} \text{at dirección} \\ \text{TOWARD } p_b \end{array} \right\} \text{ (impresión)}$

en donde

i es el número de identificación que se va a almacenar,  $p_a$  y  $p_b$  son puntos almacenados y dirección indica la dirección de la recta.

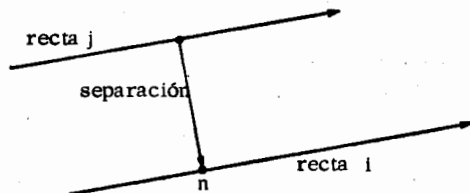
22. Declaración STORE LINE PARALLEL

Tiene la forma

$\text{STORE LINE } i \text{ thru } n \text{ PARALLEL to } \left\{ \begin{array}{l} \text{LINE } j \\ \text{línea} \end{array} \right\} \text{ (separación), (impresión)}$

en donde

n se localiza primero a la distancia de separación y sus coordenados se almacenan. Las rectas i y j tienen la misma dirección y se almacena también la recta i. En la segunda alternativa el equivalente de la recta j se determina con los elementos datos de línea. Gráficamente, se tiene



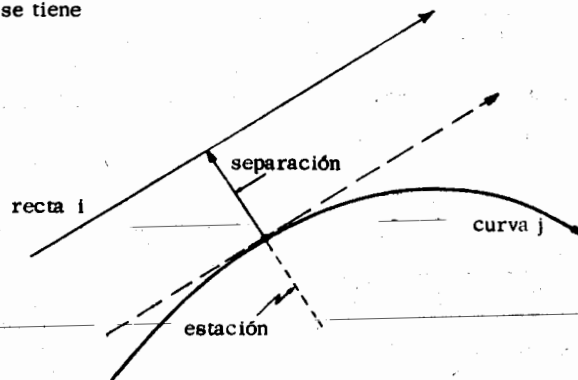
## 23. Declaración STORE LINE TANGENT

Tiene la forma

STORE LINE i thru n TANGENT to CURVE j at estación, (separación),  
(impresión)

en donde

el punto n se localiza primero en la curva almacenada j para el valor de la estación ( si se especifica la distancia, n se localiza sobre el radio a la curva y a la separación indicada ), almacenándose la recta i y las coordenadas de n en la tabla de puntos. Se calcula la dirección hacia adelante de la tangente a la curva en la estación indicada. Gráficamente, se tiene



## 24. Declaración PRINT 6 DELETE LINES

Se imprimen o borran las rectas especificadas. Tiene la forma

$$\left\{ \begin{array}{l} \text{PRINT} \\ \text{DELETE} \end{array} \right\} \left\{ \begin{array}{l} \text{LINES} \quad i, j, k, \dots \\ \text{LINES} \quad i \text{ TO } j, (k \text{ TO } l), \dots \\ \text{ALL} \quad \text{LINES} \end{array} \right\}$$

Las declaraciones 25 a 29 se aplican a cursos.

## 25. Declaración STORE COURSE

Con esta declaración se almacenan cursos y tiene la forma

$$\text{STORE COURSE } a \text{ } p_a \text{ to } \left\{ \begin{array}{l} p_b \\ n, \text{ distancia, dirección} \end{array} \right\} \text{ (Impresión)}$$

en donde

a es el nombre del curso que se almacena,  $p_a$  es el punto inicial,  $p_b$  el punto final. Las coordenadas del punto n se calculan a la distancia y dirección especificadas a partir del punto  $p_a$ . El punto n se almacena en la tabla de puntos y se utiliza posteriormente como punto  $p_b$ .

## 26. Declaración STORE COURSE PARALLEL

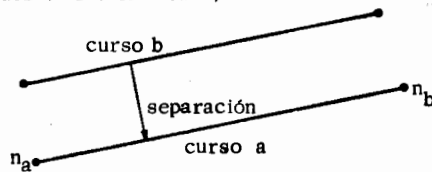
Tiene la forma

$$\text{STORE COURSE } a, n_a \text{ to } n_b, \text{ PARALLEL to } \left\{ \begin{array}{l} \text{COURSE } b, \\ \text{course } \quad \text{(separación)} \end{array} \right\}$$

(Impresión).

en donde

los puntos  $n_a$  y  $n_b$  se calculan a la separación especificada de los extremos inicial y final del curso b, respectivamente. Los puntos  $n_a$  y  $n_b$  se almacenan en la tabla de puntos y sirven posteriormente como puntos  $p_a$  y  $p_b$ . En la segunda alternativa, el equivalente al curso a y la separación se determinan a partir de cualquiera de las formas de los elementos dato del curso. Gráficamente,



### 27. Declaración TRANSPOSE COURSES

El orden de los puntos inicial y final del curso indicado se transponen en la tabla de cursos, es decir, el punto inicial se transforma en punto final y viceversa, tiene la forma.

TRANSPOSE COURSES a, b, c

### 28. Declaración STATION COURSE

Esta declaración usualmente sigue a la declaración 27 para proporcionar estaciones en el nuevo sentido hacia adelante. Se explica en la declaración de estación y tiene la forma

STATION COURSE a estación

### 29. Declaración PRINT & DELETE COURSES

Se usa para imprimir o borrar cursos y tiene la forma

$\left\{ \begin{array}{l} \underline{\text{PRINT}} \\ \underline{\text{DELETE}} \end{array} \right\} \left\{ \begin{array}{l} \underline{\text{COURSES}} \text{ a, b, c, } \dots \\ \underline{\text{ALL COURSES}} \end{array} \right\}$

en donde en impresión se tienen los variantes siguientes

COURSES  $\left\{ \begin{array}{l} \text{a, b, c, } \dots \\ p_a \text{ TO } p_b, (p_c \text{ TO } p_d), \dots \\ p_a \text{ TO } p_b \text{ TO } p_c (p_d \text{ TO } p_e \dots) \end{array} \right\}$

Las declaraciones 30 a 37 se aplican a cursos circulares.

### 30. Declaración STORE CURVE

Esta declaración tiene la forma

STORE CURVE i, referencia, elemento, a/tangente, c/estación) (impresión)

en donde

i es el número de identificación de la curva que se almacena en la tabla de curvas y

referencia = {   
PB at  $p_a$ , DB dirección, TL distancia   
PB at  $p_a$ , DB dirección, TTL distancia   
PC at  $p_a$ , DB dirección   
PI at  $p_a$ , DB dirección   
PB at  $p_a$ , PI at  $p_b$  }

elemento = {   
RADIUS distancia   
DEGREE ángulo   
LENGTH distancia   
TANGENT distancia   
LCHORD distancia   
EXTERNAL distancia   
CC at  $p_a$    
TL distancia -

pero sólo que en referencia se use el 2°

ó 5° renglón

a/tangente = {   
DA dirección   
p/m DEFLECTION ángulo   
p/m DELTA   
PA at  $p_a$    
LENGTH   
p/m TANGENT distancia   
LCHORD   
EXTERNAL   
DEGREE

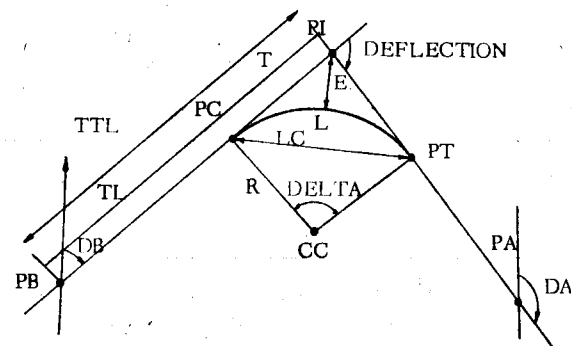
sólo si en referencia se usa el 2°, 4°, ó 5° renglones.

sólo si en elemento se utiliza RADIUS ó DEGREE

c/estación = STATION of identificación estación

en donde

identif es cualquier elemento de identificación de la curva excepto CC, pudiendo ser el punto PB. Si c/estación se omite, se supone que el valor de estación PC (ó TS) es 0+00. Gráficamente, se tiene



31. Declaración STORE CURVE con espirales de transición

Tiene la forma

STORE CURVE i, referencia, b/espiral, elemento, a/espiral, a/tangente, (c/estación), (impresión)

en donde

i es el número de identificación de la curva que va a almacenarse en la tabla de curvas; referencia es igual que en la declaración anterior, excepto que el 3er renglón es

TS at  $p_a$ , DB dirección

b/esprial = { SLB distancia }  
 (esprial hacia átras) { SAB ángulo }

Elemento = { RADIUS distancia }  
 (del segmento circular) { DEGREE ángulo }  
 { LENGTH }  
 { TANGENT } Sólo si se usa DELTA  
 { LCHORD } distancia en a/tangente  
 { EXTERNAL }

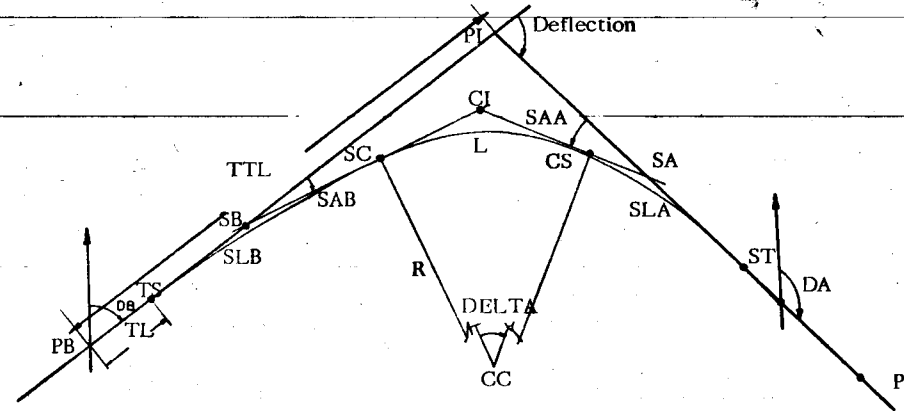
a/esprial = { SLA distancia }  
 (esprial hacia adelante) { SAA ángulo }

a/tangente = { DA dirección }  
 { p/m DEFLECTION ángulo (de curva total) }  
 { p/m DELTA ángulo (de curva circular) }  
 { PA at  $p_a$  (sólo si en referencia se usa el 2º, 4º o 5º renglón) }

Si las espirales de transición son iguales a ambos lados, se puede usar para b/esprial y/o a/esprial

{ SL distancia }  
 { SA ángulo }

Gráficamente, se tiene



32. Declaración MODIFY CURVE

Se usa para modificar una curva y tiene la forma

MODIFY CURVE i, (b/esprial), (elemento), (a/esprial), (c/impresión) estación

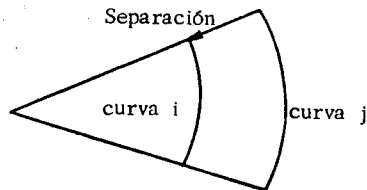
La curva almacenada i se calcula y redefine utilizando los nuevos valores de los elementos especificados. Los valores de referencia y a/tangente no se cambian, al igual que el punto PI y las dirección DA y DB. Si no se dan nuevos datos de espirales, las longitudes de esprial almacenadas no se cambian.

33. Declaración STORE CURVE CONCENTRIC

Tiene la forma

STORE CURVE i CONCENTRIC with CURVE j, separación, (c/estación),  
(impresión).

Se genera una nueva curva i con el mismo centro y tangentes paralelas a las de la curva almacenada j, desplazadas el valor de la separación. Esta declaración no se aplica a curvas con espiral. Gráficamente, se tiene

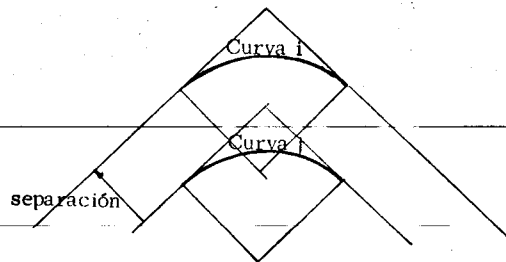


34. Declaración STORE CURVE PARALLEL

Tiene la forma

STORE CURVE i, PARALLEL to CURVE j, separación, (c/estación),  
(impresión)

Se genera una nueva curva i que tiene los mismos valores de los elementos y direcciones de tangentes que la curva almacenada j, desplazadas el valor de la separación. Gráficamente, se tiene



35. Declaración STATION CURVE

Tiene la forma

STATION CURVE i, identificación estación

en donde

identificación es cualquier elemento de identificación de la curva excepto el punto CC. La curva almacenada i se reestaciona con respecto a - - identificación que está dado por el valor de la estación.

36. Declaración TRANSPOSE CURVE

Esta declaración tiene la forma

TRASPOSE CURVE i

Se traspone la curva i, es decir, el punto PT anterior se convierte en el nuevo punto PC, invirtiéndose el sentido de recorrido, permaneciendo inalterada la posición y elementos de la curva.

La declaración se usa con ventaja para tener un solo sentido de recorrido en una cadena de curvas y rectas.

37. Declaración PRINT ó DELETE CURVES

Se usa para imprimir ó borrar curvas y tiene la forma



{	<u>PRINT</u>	}	*	<u>CURVES</u>	{	i, j, k, . . .	}
	<u>DELETE</u>			<u>ALL CURVES</u>		i TO j, (k TO 1), . . .	

Las declaraciones 38 a 43 se aplican a cadenas. Sin embargo, las declaraciones son válidas si la palabra CHAIN se reemplaza por cualquiera de: TRAVERSE, ALIGNMENT, PARCEL, BASELINE, CENTERLINE y LIST.

38. Declaración STORE CHAIN

Tiene la forma

STORE CHAIN a, lista

en donde

a es el nombre de la cadena que va a almacenarse y lista identifica a los objetos almacenados que definen a la cadena.

lista =

{	<u>(POINT)</u>	p <sub>i</sub>
	identif	i
	<u>COURSE</u>	a
	<u>CURVE</u>	i
	<u>CHAIN</u>	a
	<u>GAP</u>	

en donde identif se refiere a puntos de curvas y GAP a puntos que no están unidos por rectas ó curvas.

39. Declaración DESCRIBE CHAIN

Se dan todos los datos de los elementos de la cadena y tiene la forma

DESCRIBE CHAIN a

Si la cadena es una figura cerrada, o sea, el último punto es igual al primero, se imprime también el valor del área.

40. Declaración TRANSPOSE CHAIN

Tiene la forma

TRANSPOSE CHAIN a

Se transponen todos los objetos de la lista de la cadena a, es decir, el último objeto se convierte en el primero y así sucesivamente. También se transpone cada curva de la lista y usualmente la declaración va seguida de la declaración que se presenta a continuación.

41. Declaración STATION CHAIN

Esta declaración se explica en detalle en la declaración de estación y tiene la forma

STATION CHAIN a (FROM p<sub>a</sub>) estación (AHEAD to p<sub>b</sub>,) (BACK to p<sub>c</sub>)

42. Declaración AREA

Tiene la forma

AREA lista

en donde lista es como se definió en la declaración 38, y define el perímetro de la figura. Para una figura cerrada el último punto es igual al primero y de no ser así se establece una conexión entre los puntos extremos y se imprime un mensaje.

La figura no debe intersectarse a si misma, ya que el área calculada para figuras cerradas múltiples es incorrecta.

Para calcular e imprimir el área de una sola cadena almacenada, se usa la declaración

AREA CHAIN a

y para calcular e imprimir las áreas de segmento de una sola curva almacenada, se utiliza la declaración

AREA CURVE i

#### 43. Declaración PRINT ó DELETE CHAINS

Se imprimen ó borran cadenas y tiene la forma

PRINT CHAINS a, b, c, . . .

DELETE ALL CHAINS

Las declaraciones 44 y 45 se aplican a textos.

#### 44. Declaración STORE TEXT

Tiene la forma

STORE TEXT a, n

en donde

a es el nombre de un texto

n es el número de renglones ó tarjetas

que contienen el texto

Cuando se ejecuta la declaración se leen y almacenan las n tarjetas siguientes en la tabla de texto. El texto puede contener cualquier caracter ó símbolo legal.

#### 45. Declaración PRINT ó DELETE TEXT

Se imprimen o borran textos y tiene la forma

$\left\{ \begin{array}{l} \text{PRINT} \\ \text{DELETE} \end{array} \right\} \left\{ \begin{array}{l} \text{TEXTS } a, b, c, \dots \\ \text{ALL TEXTS} \end{array} \right\}$

Las declaraciones 46 a 51 se aplican a estaciones.

#### 46. Declaración STATION

Tiene la forma

STATION P<sub>a</sub> estación

en donde al punto P<sub>a</sub> almacenado se le da el valor de la estación.

#### 47. Declaración STATION CURVE POINTS

Esta declaración tiene la forma

STATION ON CURVE i (POINTS) P<sub>a</sub>, P<sub>b</sub>, P<sub>c</sub>, . . .

Se calculan las estaciones de los puntos enteros almacenados en la lista sobre la curva i almacenada, almacenándose en la tabla de puntos.

Si un punto no se encuentra en la curva dentro de la tolerancia  $\pm 0.10$ , se calcula la estación proyectando radialmente al punto, ó en la tangente más cercana a la curva, imprimiéndose un mensaje.

#### 48. Declaración STATION CURVE

Tiene la forma

STATION CURVE i identificación estación

en donde identificación es cualquier término perteneciente a una curva, exceptuando CC. La curva almacenada i se vuelve a estacionar con respecto a la identificación a la que se le da el valor de la estación.

#### 49. Declaración STATION CURVE POINTS

Tiene la forma

STATION ON CURVE a (POINTS) P<sub>a</sub>, P<sub>b</sub>, P<sub>c</sub>, . . .

Las estaciones de los puntos enteros almacenados de la lista se calculan con el curso almacenado a, almacenándose en la tabla de puntos. Si un punto no está en el rumbo con una tolerancia de  $\pm 0.10$ , la estación se calcula proyectando el punto sobre el rumbo ó sobre su prolongación, imprimiéndose un mensaje.

#### 50. Declaración STATION COURSE

Esta declaración tiene la forma

STATION COURSE a estación

en donde el curso almacenado a toma su valor de estación con respecto al punto inicial.

#### 51. Declaración STATION CHAIN

Tiene la forma

STATION CHAIN a (FROM P<sub>a</sub>) estación (AHEAD to P<sub>b</sub>) (BACK to P<sub>c</sub>)

La cadena almacenada a se estaciona o reestaciona continuamente, empezando en el punto P<sub>a</sub> al cual se le asigna el valor de la estación. El valor de la estación se calcula hacia adelante al punto P<sub>b</sub> o se calcula hacia atrás en el punto P<sub>c</sub>.

Las declaraciones 52 a 55 se aplican a perfiles.

#### 52. Declaración PROFILE

Esta declaración llama a un subprograma general de perfiles que procesa una serie de bloque de subdeclaraciones que definen y almacenan un perfil vertical de un alineamiento en la tabla de perfiles. La forma general es

PROFILE ' nombre '

VPI Subdeclaraciones (tres o más)

END of PROFILE, (impresión)

en donde 'nombre' es el nombre (1 a 8 caracteres) del perfil y las sub-declaraciones VPI se describen en la sección siguiente.

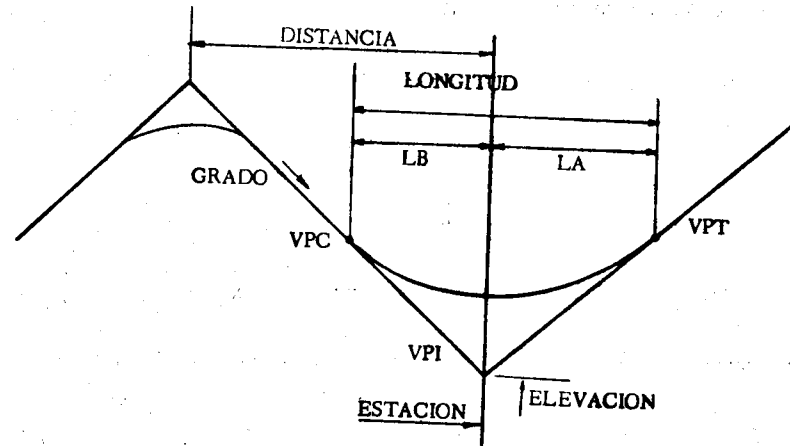
53. Subdeclaración VPI

Tiene la forma

VPI at P<sub>j</sub>, estación, elevación, longitud

en donde P<sub>j</sub> es el número del punto VPI. Si la estación no es dada está al macenada en la tabla de puntos, lo que también ocurre para elevación. Si no se da la longitud, esto significa que no se genera curva vertical en VPI.

Si se dan estos elementos que se presentan en la figura, se tiene



Otras variantes de la subdeclaración VPI son

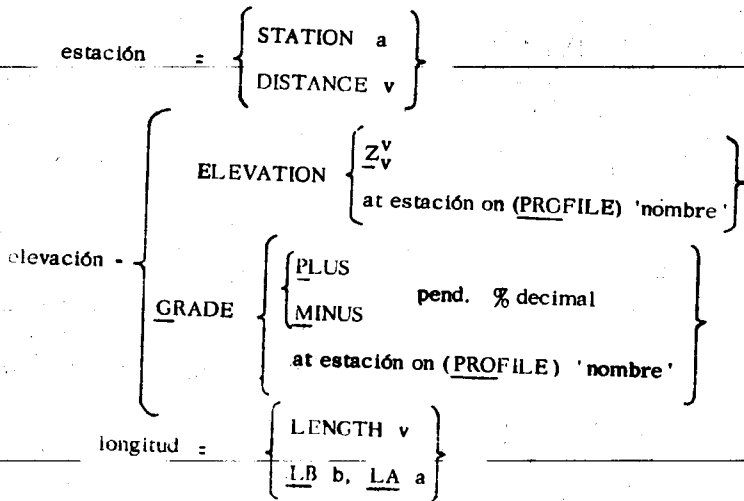
VPI at n { estación, GRADE UNKNOWN } (longitud)  
 { STATION UNKNOWN, grado }

VPI at n, estación, grado, (longitud)

VPI at P<sub>j</sub> estación, elevación, grado, (longitud)

Cuando a una curva vertical se le dan restricciones, el elemento longitud - puede ser cualquiera de las formas especiales.

longitud = { THRU P<sub>k</sub>  
THRU STATION v ELEVATION v  
CLEAR P<sub>k</sub> BY P/M v  
CLEAR STATION v, ELEVATION v BY P/M v  
FIXED (GRADE) v AT (STATION) v }



## 54. Declaración MODIFY PROFILE

Se cambia o almacena la longitud de la curva vertical y tiene la forma

MODIFY VC at (VPI)  $P_j$  (ON) (PROFILE) 'nombre' longitud

en donde el valor de longitud se cambia como se indica en la declaración 52.

## 55. Declaración PRINT ó DELETE PROFILE

Se imprime o borra la información de un perfil y tiene la forma

PRINT ELEVATIONS ON (PROFILE) 'nombre', lista

en donde

$$\text{lista} = \left\{ \begin{array}{l} a, b, c, \dots \\ \text{EVEN v (STATIONS) a TO b} \end{array} \right\}$$

Además, se tiene

$$\left\{ \begin{array}{l} \text{PRINT} \\ \text{DELETE} \end{array} \right\} \left\{ \begin{array}{l} \text{PROFILES a, b, c} \\ \text{ALL PROFILES} \end{array} \right\}$$

Las declaraciones 56 a 61 se aplican a la localización de un punto, o sea el cálculo y almacenamiento de sus coordenadas con respecto a uno o más elementos almacenados, y son las siguientes:

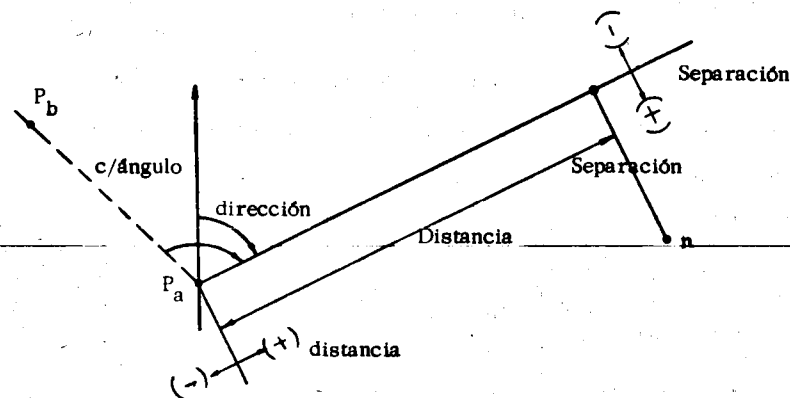
## 56. Declaración LOCATE FROM

Esta declaración tiene la forma

$$\text{LOCATE } n \text{ from p/m distancia} \left\{ \begin{array}{l} \text{dirección, separación, (impresión)} \\ \text{p/m ang. FROM } P_b \end{array} \right\}$$

y localiza al punto  $n$  a la distancia indicada y en la dirección especificada.

Gráficamente, se tiene



## 57. Declaración LOCATE ON

Tiene la forma

$$\text{LOCATE } n \text{ ON} \left\{ \begin{array}{l} \text{objeto p/m distancia from } p_a \\ \text{objeto at STATION estación, separación, (impresión)} \\ \text{CURVE i signo, at TANGENT definición} \end{array} \right\}$$

en donde

objeto proporciona la dirección sobre la que se mide la distancia para lo -

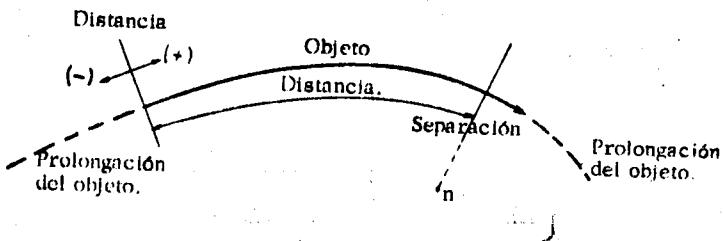
calizar n, prolongándose en caso necesario y puede ser

$$\text{objeto} = \left\{ \begin{array}{l} \text{línea} \\ \text{curso} \\ \text{curva} \end{array} \right\}$$

En la tercera alternativa el punto n se localiza en la curva en un punto de tangencia definido por

$$\begin{aligned} \text{definición} &= \left\{ \begin{array}{l} \text{dirección} \\ \text{THRU } p_b \\ \text{TO CURVE } j, \text{ signo} \end{array} \right\} \\ \text{signo} &= \left\{ \begin{array}{l} \text{CL (sentido manecillas reloj)} \\ \text{CC (sentido opuesto manecillas reloj)} \end{array} \right\} \end{aligned}$$

Se tiene la figura



58. Declaración LOCATE INTERSECT

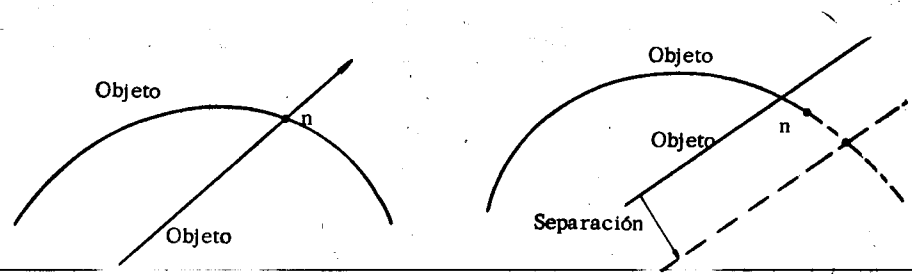
Se expresa como

LOCATE n INTERSECT objeto with objeto (NEAR p<sub>a</sub>), (impresión)

y localiza al punto n en la intersección del primer objeto con el segundo, siendo objeto como se definió en la declaración previa.

El dato opcional NEAR p<sub>a</sub> se usa cuando existe la posibilidad de dos o más intersecciones, localizándose el punto n en la intersección más cercana a p<sub>a</sub>.

Se tienen las figuras



59. Declaración LOCATE PROJECT

Tiene la forma

LOCATE n PROJECT p<sub>a</sub> con objeto, (impresión)

en donde el punto n se localiza proyectando p<sub>a</sub> en el objeto. Si el objeto es una línea o curso la proyección es perpendicular al objeto, y si es una curva la proyección es radial.

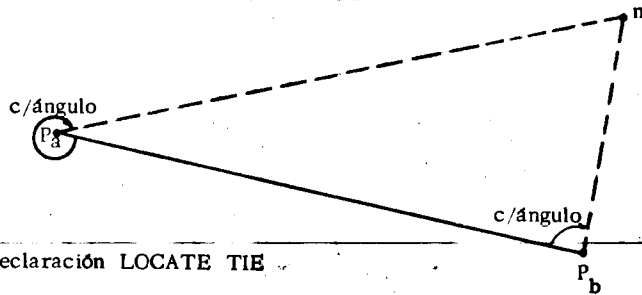
Si se da la separación con el objeto, n se localiza sobre dicha prolongación, la que se utiliza también en caso necesario para proyectar p<sub>a</sub> en el objeto.

60. Declaración LOCATE FORESECT

Esta declaración se escribe

LOCATE n FORESECT p<sub>a</sub> c/ángulo AND p<sub>b</sub> c/ángulo, (impresión)

El punto n se localiza en la intersección de dos rectas definidas por los puntos y las direcciones calculadas. La dirección puede darse como dato en lugar de los ángulos).



61. Declaración LOCATE TIE

Tiene las formas

LOCATE n TIE p<sub>a</sub> distancia, TIE p<sub>b</sub> { DISTANCE distancia, APPROX. direc.  
c/ángulo, (NEAR or FAR) }  
(Impresión)

Con la primera alternativa se intersectan dos círculos con centros en p<sub>a</sub> y p<sub>b</sub> y con radios iguales a las distancias especificadas respectivamente. Debido a que hay dos intersecciones posibles, se necesita la dirección aproximada de p<sub>a</sub> a n, para definir cual es la intersección n.

En la segunda alternativa se intersecta un círculo de centro p<sub>a</sub> y radio la distancia indicada, con una recta especificada que pasa por p<sub>b</sub>. Se usan

NEAR cuando n es la solución más cercana a p<sub>a</sub> y FAR en caso contrario.

En las declaraciones 62 a 74 se presentan las declaraciones y subdeclaraciones de alineamiento, que se pueden usar para resolver una gran variedad de problemas de alineamientos horizontales. Cada subdeclaración de alineamiento define un punto (POT) o curva en el alineamiento y adiciona un objeto a la lista de cadena para el alineamiento. El alineamiento se almacena automáticamente en la tabla de cadena para usarlo posteriormente.

61. Declaración ALIGNMENT

La forma general de la declaración es la siguiente:

ALIGNMENT 'nombre', tipo

en donde

'nombre' se refiere al nombre del alineamiento y

tipo = { OPEN  
INCOMPLETE  
INVERSE }

63. Declaración ALIGNMENT OPEN

En un alineamiento abierto, los datos para definir cada curva y cada POT están disponibles. Cada curva y POT son independientes de aquellos adelante de él, siendo un alineamiento abierto análogo a un transversal abierto.

Un alineamiento abierto siempre se puede definir por una serie de declaraciones de almacenamiento y localización de puntos, curvas y cadenas. La declaración de alineamiento abierto y subdeclaraciones asociadas proporcionan una manera más concisa de definir y almacenar el alineamiento y los puntos y curvas que lo definen.

La forma general para este caso es

ALIGNMENT 'nombre', OPEN (requerida)

Subdeclaración Punto inicial POT (requerida)

Subdeclaraciones Curva y POT (opcional)

Subdeclaración Punto Terminal POT (opcional)

END of ALIGNMENT, (DESCRIBE) (requerida)

en donde la primera declaración indica a la declaración ALIGNMENT OPEN, describiéndose a continuación las subdeclaraciones restantes.

Se define con POT a un punto entero que pertenece a un alineamiento, proporcionando las subdeclaraciones POT la información necesaria para determinar las coordenadas del punto.

#### 64. Subdeclaración punto inicial POT

Tiene la forma

POT p/dato, STATION estación

en donde

$$p/dato = \left\{ \begin{array}{l} n \text{ coordenadas} \\ n \text{ at } p_a \\ \text{at } p_j \\ \text{at } IP \end{array} \right\}$$

siendo  $n$  el número asignado a POT y usándose la última alternativa para un alineamiento incompleto.

#### 65. Subdeclaración puntos intermedios o final POT

Esta subdeclaración se usa en el caso de alineamiento abierto para puntos intermedios o final POT y tiene las formas

$$POT \left\{ \begin{array}{l} P/dato \\ n \text{ TL distancia (TD dirección c/ángulo)} \\ n \text{ TTL distancia} \end{array} \right\} (\text{STATION estación})$$

#### 66. Subdeclaración de curva

Se presenta como

$$\text{CURVE } 1 \left\{ \begin{array}{l} \text{referencia, b/espíral, elemento, a/espíral, a/tangente} \\ \text{STORED,} \end{array} \right\} c/estación$$

en donde los elementos de datos son los mismos que para la declaración 30 de STORE CURVE, excepto que



referencia =	}	<u>TL</u> distancia
		<u>TTL</u> distancia
		<u>DB</u> dirección, TL distancia
		<u>DB</u> c/ángulo, TL distancia
		<u>PI</u> at $p_a$
		<u>PI</u> coordenadas

elemento puede ser TL distancia si se da PI en referencia y la curva tiene espirales.

Si c/estación no se da, el cadenamamiento es llevado hacia adelante de la curva previa o POT.

#### 67. Declaración ALIGNMENT INCOMPLETE

Un alineamiento incompleto es aquel que tiene partes omitidas. El alineamiento debe ser procesado como un todo para determinar los valores omitidos antes de que las curvas y los POT individuales sean completamente definidos, calculados y almacenados.

Un alineamiento incompleto puede tratarse como un problema transversal. La declaración de alineamiento incompleto y subdeclaraciones asociadas permiten especificar el problema de una manera concisa y resolverlo en forma automática.

La forma general para este caso es

ALIGNMENT 'nombre', INCOMPLETE (requerida)  
 Subdeclaración Punto inicial POT (requerida)  
 Subdeclaración Curva y POT (requerida)  
 Subdeclaración Punto terminal POT (requerida)  
END of ALIGNMENT, (DESCRIBE) (requerida)

#### 68. Subdeclaración punto inicial POT

Tiene la forma

POT p/dato, TD dirección, STATION estación

en donde TD dirección proporciona la dirección de la tangente del punto inicial al POT o PI hacia adelante. p/dato es como se explica en la declaración 64.

#### 69. Subdeclaración punto intermedio POT

Se escribe como

POT n TL distancia, c/ángulo

en donde el dato c/ángulo se proporciona si el POT o PT previo es un punto en un ángulo.

#### 70. Subdeclaración punto terminal POT

Tiene la forma

POT p/dato, TL distancia, c/ángulo, TD dirección

en donde TD dirección da la dirección de la tangente del POT o PT previo al punto terminal. El dato c/ángulo se proporciona si el POT o PT previo es un punto en un ángulo.

### 71. Subdeclaración de curva

Esta subdeclaración se expresa con

CURVE 1 (DB c/ángulo),

TL distancia, b/espiral, elemento, a/espiral, a/tangente  
en donde

$$\text{espiral} = \left\{ \begin{array}{l} \text{SLA distancia, SLB distancia} \\ \text{SL distancia} \end{array} \right\}$$

$$\text{elemento} = \left\{ \begin{array}{l} \text{RADIUS distancia} \\ \text{DEGREE ángulo} \\ \text{LENGTH} \end{array} \right\}$$

$$\text{a/tangente} = \text{P/M DEFLECTION ángulo}$$

El dato DB c/ángulo se da si el POT o PT previo es un punto de un ángulo.

Para especificar las partes omitidas en un alineamiento incompleto, los datos pueden tener las siguientes formas adicionales.

$$\text{TL distancia} = \left\{ \begin{array}{l} \text{TL ?} \\ \text{TL ? APPROXIMATE distancia} \end{array} \right\}$$

$$\text{c/ángulo} = \left\{ \begin{array}{l} \text{P/M ?} \\ \text{P/M ? APPROXIMATE ángulo} \end{array} \right\}$$

$$\text{elementos} = \left\{ \begin{array}{l} \text{R ?} \\ \text{R ? APPROXIMATE distancia} \end{array} \right\}$$

$$\text{a/tangente} = \left\{ \begin{array}{l} \text{P/M DEFLECTION ?} \\ \text{P/M DEFLECTION ? APPROXIMATE ángulo} \end{array} \right\}$$

$$\text{TD dirección} = \left\{ \begin{array}{l} \text{TD (AZIMUTH) ?} \\ \text{TD (AZIMUTH) ? APPROXIMATE dirección} \end{array} \right\}$$

Cada distancia desconocida (TL y R) cuenta como una parte omitida. El número de partes omitidas atribuibles a incógnitas de ángulo y dirección es menor en uno que el número de incógnitas.

### 72. Declaración ALIGNMENT INVERSE

En un alineamiento inverso cada POT y cada PI de una curva está en un punto almacenado o de coordenadas dadas. Las direcciones y distancias entre los puntos se calculan internamente por inversión, proporcionándose facilidades para ajustar las coordenadas de los puntos para que las direcciones entre ellas tengan un valor par de minutos o segundos.

La forma general para este caso es como sigue:

ALIGNMENT 'nombre', INVERSE, (redondear) (requerida)

Subdeclaración punto inicial POT (requerida)  
 Subdeclaración curva y POT (requerida)  
 Subdeclaración punto terminal POT (requerida)  
END of ALIGNMENT, (DESCRIBE) (requerida)

en donde (redondear) = ROUND TO v  $\left\{ \begin{array}{l} \text{MINUTES} \\ \text{SECONDS} \end{array} \right\}$

Cuando se especifica redondear se ajustan los puntos POT y PI de manera que las direcciones de las tangentes a curvas tengan el valor par más cercano especificado en el dato de redondear.

#### 73. Subdeclaración puntos inicial, intermedios y final POT

La forma general para esta subdeclaración es:

POT p/data, (STATION estación), (HOLD)

La primera y la última subdeclaraciones deberán ser para definir los puntos inicial y terminal, respectivamente. El dato "station" se requiere necesariamente para el punto inicial.

p/data se explica en la declaración 64 y el uso de HOLD en la declaración siguiente.

#### 74. Subdeclaración de curva

Tiene la forma general

CURVE i, PI coord.,  
 (HOLD), b/esprial, elemento, a/esprial, c/station

Los términos de espirales y elemento se describen en la declaración 30 de STORE CURVE.

En las declaraciones 75 a 78 se presentan las declaraciones de localización que se utilizan para calcular e imprimir datos para usarse en el estacado del terreno, localización o gráfica de los objetos almacenados, particularmente curvas y alineamientos.

#### 75. Declaración LAYOUT TIES

Proporciona datos de localización que refieren puntos y curvas almacenadas a un punto almacenado y a una dirección de referencia. Tiene la forma

LAYOUT TIES  $\left\{ \begin{array}{l} \text{TRANSIT at } P_a, \text{ (SIGHT on) } P_b, \text{ TIE} \\ P_a \text{ TO} \end{array} \right\}$  lista

en donde lista toma la forma de una lista de puntos almacenados o de una cadena almacenada.

#### 76. Declaración LAYOUT CURVE

Proporciona datos para la localización y trazo de curvas almacenadas y tiene las formas

LAYOUT  $\left\{ \begin{array}{l} \text{CURVE } i, \text{ tránsito, par, desplazamiento, más} \\ \text{CURVES, ALIGNMENT } a, \text{ par, desplazamiento} \\ \text{ALL CURVES, par, desplazamiento} \end{array} \right\}$

en donde,

tránsito = (TRANSIT at)  $P_a$ , (SIGHT on)  $P_b$

par = (EVEN) v (STATIONS)

desplazamiento = (OFFSETS)

más =  $\left\{ \begin{array}{l} \text{PLUS STATION } v_a \text{ (STATION } v_b), \dots \\ \text{PLUS } P_a, P_b, P_c, \dots \end{array} \right\}$

Si no se proporciona tránsito,  $P_a$  se considera el PC de la curva i y  $P_b$  el PI de la curva, lo que ocurre siempre con la segunda y tercera alternativa.

#### 77. Declaración LAYOUT OFFSET

Proporciona distancias para localización de una cadena almacenada (línea base) a otra cadena almacenada (alineamiento). Tiene la forma

LAYOUT OFFSETS, (BASELINE) a to (ALIGNMENT) b, list

en donde

lista =  $\left\{ \begin{array}{l} \text{(STATIONS) } v_a, v_b, v_c, \dots \\ \text{EVEN v (STATIONS) } v_a \text{ TO } v_b \end{array} \right\}$

#### 78. Declaración LAYOUT ALIGNMENT

Esta declaración se usa para calcular e imprimir las coordenadas de estaciones pares sobre un alineamiento almacenado y tiene la forma

LAYOUT ALIGNMENT a, COORDINATES at (EVEN) v (STATIONS)

Además de las coordenadas de cada estación par, se imprimen la estación y coordenadas de cada punto y cada punto de curva en la lista de cadena.

En las declaraciones 79 a se presentan la declaración de poligonal y subdeclaraciones asociadas, que pueden utilizarse para resolver una gran variedad de problemas.

La poligonal puede ser abierta o cerrada, y en este último caso completa ó incompleta. Si es incompleta puede tener cualquier combinación de una ó dos partes omitidas. Si es cerrada con una o sin ninguna parte omitida, se rá ajustada de acuerdo a las reglas de ajustes. Puede ser en un ángulo, una dirección o una combinación, y se pueden mantener fijos cursos individuales durante el ajuste. La poligonal es almacenada automáticamente en la tabla de cadenas para utilización posterior.

#### 79. Declaración TRAVERSE

La forma general de la declaración y el orden normal de las subdeclaraciones son

<u>TRAVERSE</u>	'nombre'	(requerida)
Subdeclaración de ajuste		(opcional)
Subdeclaración de cierre		(opcional)
Subdeclaración de retroceso		(opcional)
Subdeclaraciones de curso		(se requieren 2 o más)
Subdeclaración hacia adelante		(opcional)

END of TRANVERSE, (REPORT)(PRINT)(SKETCH) (requerida)

en donde 'nombre' es el nombre de la poligonal que se almacena en la tabla de cadenas. El bloque de las subdeclaraciones de curso debe estar en el orden de los cursos, pero las otras pueden estar en cualquier orden, siempre y cuando vengan después de TRAVERSE y antes de END.

Si se da REPORT, se imprime un reporte completo de correcciones para poligonales ajustadas. También da los valores sin ajustar, la corrección y los valores ajustados para todas las distancias, direcciones, ángulos, latitudes, salidas y coordenadas de puntos. Con PRINT se imprimen tablas de valores ajustados de cursos. Con SKETCH se imprime un croquis de la poligonal en la impresora y permite distinguir errores importantes.

#### 80. Subdeclaración ADJUST

Tiene la forma

$$\text{ADJUST BY } \left\{ \begin{array}{l} \text{LINEAR} \\ \text{TRANSIT} \\ \text{COM PASS} \\ \text{CRANDALL} \end{array} \right\} \text{ (RULE)} \quad \left\{ \begin{array}{l} \text{TAPE} \\ \text{DISTANCE} \end{array} \right\} \text{ (WEIGHT) } v$$

LEAST SQUARES

Siendo  $v$  el valor numérico del peso que se asignará a cada 100 pies de cinta (si TAPE es dado) ó a cada longitud de curso (si DISTANCE es dado) en relación al peso del ángulo unitario. Si no se especifica el peso, a las distancias y a los ángulos se les da el mismo peso.

#### 81. Subdeclaración CLOSURE

Tiene la forma

CLOSURE ONE (PART IN)  $v_1$ , PER (ANGLE)  $v_2$  (SECONDS)

Los valores de cierre se usan como especificación contra la cual se compara el valor real de cierre de la poligonal. Si el cierre de la poligonal excede al doble de la especificación, el procesamiento continúa pero no puede entrarse a las tablas de datos. El valor  $v_1$  se usa para un ajuste angular preliminar y la especificación  $v_2$  se utiliza para un ajuste general.

#### 82. Subdeclaración BACK

Se escribe

BACK dirección

Esta subdeclaración se usa para dar la dirección hacia atrás al punto inicial de la poligonal para calcular las direcciones de los cursos de los ángulos de entrada y/o para proporcionar las bases para hacer un ajuste preliminar de ángulos. La subdeclaración puede omitirse si no es necesaria para una computación de curso hacia adelante ó si no se va a hacer un ajuste

angular preliminar.

### 83. Subdeclaración COURSE

Esta subdeclaración se escribe

COURSE ('nombre') (TO) n (FIXED) c/distancia, c/dirección

Si se proporciona el nombre del curso, éste será almacenado en la tabla de cursos. Si se da la palabra opcional FIXED la longitud y la dirección del curso serán mantenidos fijos y no serán cambiados durante el ajuste general.;

El dato j es el punto inicial del curso y n es el final.

$$\text{distancia} = \left\{ \begin{array}{l} \text{distancia} \\ (\underline{\text{DISTANCE}}) \text{ ? } (\underline{\text{APPROXIMATE}}) \text{ distancia} \\ (\underline{\text{DISTANCE}}) \underline{\text{SAME}} \end{array} \right\}$$

La segunda forma se usa cuando la longitud del curso es desconocida y se trata como una parte omitida o faltante.

El dato opcional APPROXIMATE se usa para proporcionar un valor aproximado para la distancia desconocida en dos soluciones. La forma SAME se usa cuando la longitud del curso tiene el mismo valor que la distancia desconocida previa, considerándose las dos como una sola parte omitida en la poligonal.

$$\text{c/dirección} = \left\{ \begin{array}{l} \text{dirección} \\ (\underline{\text{AZIMUTH}}) \text{ ? } (\underline{\text{APPROXIMATE}}) \text{ dirección} \\ \text{c/ángulo ?} \\ \underline{\text{ANGLE}} \text{ ? } (\underline{\text{APPROXIMATE}}) \text{ P/M ángulo} \end{array} \right\}$$

La primera forma se usa cuando la dirección del curso se da como dato de entrada y la segunda cuando la dirección del curso es desconocida. La tercera alternativa ocurre cuando el ángulo en j del curso previo a n se da como dato de entrada y va a ser usado internamente para calcular la dirección del curso y la cuarta forma es usada cuando el c/ángulo es desconocido. El dato opcional APPROXIMATE en la segunda y cuarta formas es usado para proporcionar un valor aproximado a la incógnita para usarse en los dos casos de solución.

La palabra UNKNOWN puede usarse en lugar del símbolo ? y deberá utilizarse cuando el problema va a ser corrido en una máquina que no tenga el signo ? en su impresora.

### 84. Subdeclaración AHEAD

Tiene la forma

AHEAD dirección CANGLE c/ángulo

Esta subdeclaración se usa para dar una dirección hacia adelante y un ángulo de cierre en el punto final de la poligonal, para usarse en el cálculo de las

direcciones de los cursos a partir de los ángulos de entrada y/o proporcionar las bases para un ajuste preliminar de los ángulos. La dirección se mantiene fija para hacer el ajuste. El c/ángulo es para dar el ángulo de cierre en el punto terminal desde el punto penúltimo hacia el punto final.

Puede tomar cualquiera de las formas de ángulo para el elemento c/dirección incluso la forma del ángulo incógnitas. La subdeclaración puede ser omitida si no es necesaria para el cálculo de cursos o si no se hace el ajuste angular preliminar.

*notas sobre el sistema  
integrado de ingeniería civil*

**subsistema  
ICES-ROADS**

*Dr. Rodolfo Luthe G.*



Este subsistema del ICES permite la solución de problemas de diseño y localización de carreteras. El orden de presentación de las declaraciones corresponde al que ocurre en un problema real, o sea, entrada de datos del terreno y subsuelo, alineamiento y perfiles, requerimientos de volúmenes y diseño de la carretera, así como comportamiento de los vehículos.

Los tópicos anteriores constituyen las fases o aspectos principales del subsistema ROADS.

El subsistema tiene un marco de referencia básico que es aplicable a una gran gama de problemas de transporte y de Ingeniería Civil, que implica el relleno y excavación de material, y que incluye carreteras, canales y diques. El usuario debe comprender siempre la relación especial que existe entre los subsistema ROADS y COGO, habiéndose diseñado el primero de manera que es completamente compatible con el segundo. Por facilidad se incluyen las declaraciones de COGO que se usan generalmente para alineamientos y perfiles geométricos. Entre ROADS y COGO existe transferencia de datos geométricos de manera que los datos desarrollados en un sistema se pueden utilizar en el otro subsistema, directamente y de una manera eficiente.

Los problemas se pueden resolver en forma incremental, seleccionando operaciones y declaraciones que se basan en datos de entrada y resultados de diseño previos pudiéndose obtener únicamente los resultados que se deseen.

\* El material que se presenta se ha tomado del manual R68-9, ICES ROADS I, Roadway Analysis and Design System, de John H. Suhrbier, John C. Prokopy y Edward C. Sullivan, publicado por Civil Engineering Systems Laboratory and Transportation Systems Division, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, Mass.

Las declaraciones que pueden utilizarse en el subsistema ROADS son:

1. - Declaración ROADS

Tiene la forma

ROADS

y le indica al sistema que se resolverán problemas del subsistema indicado.

2. - Declaración FINISH

Se utiliza para indicar la terminación de un problema o de una corrida y se escribe

{  
FINISH ROADS  
FINISH  
}

en donde la primera alternativa indica la terminación del subsistema y prepara para la lectura de otra declaración del subsistema ROADS o el nombre de otro subsistema. La segunda alternativa indica la terminación normal de una corrida.

3. - Declaración PROBLEM NAME

Permite especificar el nombre de cada uno de los problemas considerados y tiene la forma

PROBLEM NAME ' nombre del problema '

en donde el nombre del problema puede tener un máximo de 72 caracteres, encerrados dentro de comillas.

Las declaraciones 4 a 6 proporcionan la capacidad de mantenimiento de archivos para todos los archivos de información que pueden guardarse permanentemente entre las corridas de máquina y son

4. - Declaración LIST

Se expresa en la forma

LIST tipo

en donde

- tipo =
- ALIGNMENTS
  - PROFILES
  - TYPICALS
  - TEMPLATE (CHANGE) (SETS)
  - ROADWAY (DESCRIPTIONS)
  - TRAFFIC (DATA) (SETS)
  - ALTERNATIVES
  - MATERIALS (DEFINED)
  - BORINGS
  - GROUPS (IN) SOUNDINGS (FILE)
  - CONTOURS
  - SUBSYSTEM (UNAMED) (FILES)
  - ALL (FILE) (NAMES)

y permite obtener el listado de los nombres de objetos definidos en un problema específico, de un tipo particular o de todos los tipos.

5. - Declaración DELETE

Tiene la forma

DELETE {  
 tipo<sub>1</sub>, 'nombre',.... 'nombre'  
ALL tipo<sub>1</sub>  
 tipo<sub>2</sub> FULLY

Con la alternativa primera se pueden borrar objetos o archivos selectos de un tipo particular y con la alternativa segunda todos los objetos o archivos de un tipo particular. Con la alternativa tercera se pueden borrar selectivamente elementos de datos individuales o un archivo de entrada completo del modelo de terreno multimaterial. Además, se tiene

- tipo<sub>1</sub> =
- ALIGNMENTS
  - OFFSETS
  - PROFILES
  - TYPICALS
  - TEMPLATE (CHANGE) (SETS)
  - ALTERNATIVES
  - ROADWAY (DESCRIPTIONS)
  - TRAFFIC (DATA) (SETS)

5.

tipo2 = {  
MODEL  
SURFACE (DATA)  
(SURVEY) NOTEBOOK  
BORINGS  
SOUNDING  
CONTOURS

'nombre' es el nombre de un archivo y tiene ocho o menos caracteres alfanuméricos.

6. - Declaración PRINT

La declaración de impresión existe para todos los tipos indicados en las declaraciones previas, anteponiendo la palabra PRINT. Esta declaración se considera en detalle en cada uno de las secciones del subsistema.

7. - Declaración ALTERNATIVE

Se presenta en la forma

ALTERNATIVE 'nombre' (EQUALS) (ALIGNMENT 'nombre'), (PROFILE 'nombre')  
 (TYPICAL 'nombre'), (TEMPLATE (SET) 'nombre'),  
 (ROADWAY (DESCRIPTION) 'nombre', (TRAFFIC (SET) 'nombre')

determinándose una alternativa de diseño única con el conjunto de elementos indicados de alineamiento, perfil, sección transversal típica y datos de

6.

plantilla. Cada declaración de ejecución requiere en general el nombre de alguna alternativa.

8. - Declaración LIST, PRINT ó DELETE ALTERNATIVES

Se tienen las expresiones

LIST ALTERNATIVES

{  
PRINT  
DELETE } ALTERNATIVE 'nombre',...('nombre')  
DELETE ALL ALTERNATIVES

para listar los nombres de todas las alternativas definidas, imprimir o borrar los objetos componentes de las alternativas especificadas.

9. - Declaración FILE SAVE

Permite la transmisión de información entre los subsistemas ROADS y COGO, generándose y almacenándose los datos en un subsistema y transmitiéndose al otro. Por ejemplo, se tiene

COGO

FILE SAVE 'identif' 'titulo'

RCADS

FILE RESTORE 'titulo'

pudiéndose intercambiar los subsistemas COGO y ROADS

### Modelo multimaterial del terreno

Con este modelo se puede representar adecuadamente la superficie y sub-suelo del terreno para la localización y diseño de carreteras. Los datos del modelo permiten considerar de manera eficiente un gran número de posiciones de alineamientos posibles.

Los datos se proporcionan con respecto a una línea base estándar (eje x) a la que se hace referencia con "linbest". Antes de considerar los datos, el usuario selecciona la franja de terreno de su interés, definiendo geométricamente la línea base en esta franja, pudiendo definirse el sistema con segmentos lineales o segmentos lineales conectados con curvas. La línea base se considera como una cadena COGO y se define geoméricamente como una serie de puntos, curvas, cursos y subcadenas.

Se puede imprimir cualquier porción de los datos de entrada o del modelo construido, pudiéndose efectuar modificaciones por medio de las declaraciones de modificación.

Las declaraciones 10 a 23 se aplican a datos superficiales y son:

#### 10. - Declaración ASSUMED MATERIAL

Tiene la forma

ASSUMED (MATERIAL) (NAME) (IS) 'nombre del material'

#### 11. - Declaración SURFACE DATA

Se utiliza para la entrada de datos topográficos como distancias y

elevaciones. Se escribe

SURFACE (ELEVATION) (DATA), [BASELINE] 'nombre', (HORIZONTAL)  
(MULTIPLICATION) (FACTOR) v), ( [VERTICAL] (MULTIPLICATION)

(FACTOR v) {  $\left. \begin{array}{l} \text{FREE} \\ \text{FIXED} \end{array} \right\}$  (FORMAT)

linbest dist elev, dist elev, dist elev, . . .

END

Una forma especial consiste en escribir DTM en lugar de SURFACE (ELEVATION), anteponiéndose en el tercer renglon y siguientes, los elementos

identif terreno, identif linbest

#### 12. - Declaración CROSS SECTION PARTIALLY

Esta declaración representa una forma adicional para la entrada de secciones transversales parcialmente reducidas de notas topográficas, proporcionándose lecturas de elevación de instrumento y rod en lugar de las elevaciones absolutas. Tiene la forma

CROSS (SECTION) (NOTES), (PARTIALLY REDUCED, BASELINE 'nombre'  
 (HORIZONTAL (MULTIPLICATION) (FACTOR) y), (VERTICAL (MULTIPLICATION  
(FACTOR) y)

linbest elev instr. dist lect. rod dist lect. rod dist lect. rod . . .

END

13. - Declaración EDIT DATA SURFACE

Los datos superficiales se pueden editar para verificar errores de registro y en los reportes. Se escribe

EDIT DATA SURFACE { ALL  
FROM linbest TO linbest  
AT linbest }

pudiendo efectuarse las pruebas siguientes

EDIT SPECIFICATIONS

CONTOUR (INTERVAL) y  
OFFSET (SPACING) ( [MAXIMUM] y ) ( [MINIMUM] y )  
POINT (DENSITY) ( [MAXIMUM] y ) ( [MINIMUM] y )  
GROUND ELEVATION ( [MAXIMUM] y ) ( [MINIMUM] y )  
GROUND ELEVATION CHANGE ( [MAXIMUM] y ) ( [MINIMUM] y )  
GROUND SLOPE ( [MAXIMUM] y ) ( [MINIMUM] y )

END

14. - Declaración PRINT SURFACE DATA

Se genera un reporte en forma de tabla con un listado de distancias y elevaciones superficiales para cada una de las líneas de base solicitadas. Tiene la forma

ALL  
PRINT SURFACE (DATA) FROM linbest TO linbest  
AT linbest

15. - Declaración ADD SURFACE DATA

Se añaden datos y tiene la expresión

ADD TO SURFACE (DATA)  
linbest dist elev dist elev dist elev . . .

END

16. - Declaración CHANGE SURFACE DATA

Permite cambiar los datos almacenados y se escribe

linbest dist elev dist elev dist elev . . .

END

17. - Declaración DELETE SURFACE DATA

Se utiliza para borrar información y tiene la forma

DELETE SURFACE (DATA) { ITEMS }  
 { FULLY }

linbest  
 linbest dist dist dist . . .  
 linbest FROM dist TO dist  
FROM linbest TO linbest  
FROM linbest TO linbest, FROM dist TO dist

END

18. - Declaración CROSS SECTION NOTES

Con esta declaración se almacenan libros de registro de campo de secciones transversales y tiene la forma

CROSS (SECTION) (NOTES), [BASELINE] 'nombre'

(BOOK i) (PAGE j) (LINE k)

INSTRUMENT (BACK (SIGHT) y), (HI y) (AHEAD) (SIGHT y)

(ELEVATION y)

linbest dist lec. rod dist lect. rod dist lect. rod . . .

END

19. - Declaración EDIT DATA NOTEBOOK

Se usa para editar los datos de libro de secciones transversales y tiene la forma

EDIT DATA NOTEBOOK { ALL }  
 { FROM localidad TO localidad }  
 { AT localidad }

en donde

localidad = (BOOK i), (PAGE j), (LINE k)

Se pueden efectuar las pruebas siguientes:

EDIT SPECIFICATIONS

HEIGHT (OF)(INSTRUMENT) (CHANGE) ([MAXIMUM] y) ([MINIMUM] y)

ROD (READING) CHANGE ([MAXIMUM] y) ([MINIMUM] y)

ROD (READING) OFFSET ([MAXIMUM] y) ([MINIMUM] y)

ORDERED (FIELD) (NOTEBOOK) OFFSETS (REQUIRED)

ORDERED (FIELD) (NOTEBOOK) STATIONING (REQUIRED)

END

20. - Declaración PRINT SURVEY NOTEBOOK

Sirve para hacer un listado por libro, página y renglón de las localidades especificadas del libro de campo de secciones transversales. Se escribe con

PRINT (SURVEY) NOTEBOOK  $\left. \begin{array}{l} \text{ALL} \\ \text{FROM localid ad TO localid ad} \\ \text{AT} \end{array} \right\} 13.$

21. - Declaración ADD NOTEBOOK

Modifica los datos almacenados y tiene la forma

ADD TO NOTEBOOK

(BOOK i), (PAGE j) (LINE k)

INSTRUMENT ([BACK] (SIGHT) v), ([HI] v), ([AHEAD] (SIGHT) v),

linbest dist lect. rod dist lect. rod dist lect. rod . . .

OFFSETS dist lect. rod dist lect. rod dist lect. rod . . .

END

22. - Declaración CHANGE NOTEBOOK

Se usa para cambiar información almacenada y se escribe con

CHANGE (POINT) (IN) NOTEBOOK

(BOOK i), (PAGE j), (LINE k)

INSTRUMENT ([BACK] (SIGHT) v), ([HI] v) ([AHEAD] (SIGHT) v),

([ELEVATION] v)

linbest dist lect. rod dist lect. rod dist lect. rod . . .

14. ROD (READINGS) dist lect. rod dist lect. rod . . .

END

23. - Declaración DELETE NOTEBOOK

Se utiliza para borrar la información especificada y se expresa con

DELETE (SURVEY) NOTEBOOK  $\left. \begin{array}{l} \text{ITEMS} \\ \text{FULLY} \end{array} \right\}$

$\left. \begin{array}{l} \text{localidad} \\ \text{localidad dist, dist, dist . . .} \\ \text{FROM localid ad TO localid ad} \end{array} \right\}$

END

Las declaraciones 24 a 32 se aplican a datos referencados al subsuelo y son las siguientes:

24. - Declaración BORINGS.

Sirve para la entrada de datos de subsuelo y perforaciones o taladros de reportes de campo y tiene la forma

BORINGS, [BASELINE] 'nombre <sub>1</sub>'

ORDER (OF) (MATERIALS) nombre mat<sub>1</sub> nombre mat<sub>2</sub> nombre mat<sub>3</sub>

GROUP (NAME) 'nombre<sub>2</sub>' [DENSITY] n

STATION linbest

OFFSET dist (BORING) (NAME) 'id' (DEPTHS) dist sup<sub>1</sub>, dist sup<sub>2</sub>,...

END

en donde dist. sup indica la distancia entre la parte superior de la capa del subsuelo a la superficie y siempre es positiva.

25. - Declaración SOUNDINGS

Esta declaración describe un solo material del subsuelo en el caso de que la declaración previa describiera una o más capas del subsuelo. Se expresa con

SOUNDINGS, [BASELINE] 'nombre<sub>1</sub>'

MATERIALS nombre mat<sub>1</sub> (OVER) nombre mat<sub>2</sub>

GROUP (NAME) nombre<sub>2</sub> [DENSITY] n

linbest dist dist. sup dist dist. sup...

END

26. - Declaración DEPTH OF COVER CONTOURS

La declaración se utiliza para proporcionar información adicional de la forma de las capas del subsuelo, cuando se tiene la forma de rectas de igual distancia a la superficie, como por ejemplo, de fotografías aéreas. Tiene la forma

DEPTH (OF) (COVER) (CONTOURS) [BASELINE] nombre<sub>1</sub>

MATERIALS nombre mat<sub>1</sub> (OVER) nombre mat<sub>2</sub> [MAXIMUM] (DEPTH) y<sub>1</sub>,

( MINIMUM (DEPTH) v<sub>2</sub>)

GROUP (NAME) nombre<sub>2</sub>

linbest dist dist. sup dist dist. sup...

END

27. - Declaración DATA

La edición de archivos de datos del subsuelo se inicia con la expresión

EDIT DATA {  
BORINGS  
SOUNDINGS  
CONTOURS}

puediendo efectuarse las pruebas siguientes:



EDIT SPECIFICATIONS

BORING DENSITY ( [MAXIMUM] y ) ( [MINIMUM] y )

BORING SPACING ( [MAXIMUM] y ) ( [MINIMUM] y )

MAXIMUM BORING (OFFSET) { BOTH  
LEFT } ( [MAXIMUM] y ) ( [MINIMUM] y )  
RIGHT }

SOUNDING DENSITY ( [MAXIMUM] y ) ( [MINIMUM] y )

SOUNDING SPACING ( [MAXIMUM] y ) ( [MINIMUM] y )

MAXIMUM SOUNDING (OFFSET) { BOTH  
LEFT } ( [MAXIMUM] y ) ( [MINIMUM] y )  
RIGHT }

END

28. - Declaración LIST GROUPS

Se imprimen los nombres de todos los grupos designados en el archivo de entrada de subsuelo y tiene la forma

LIST GROUPS (IN) { BORINGS  
SOUNDINGS } ( FILE )  
CONTOURS }

Cuando se desea la impresión de nombres de materiales de la superficie y subsuelo se utiliza la expresión

LIST MATERIALS (DEFINED) (IN) (THE) (SUBSYSTEM)

29. - Declaración PRINT GROUPS

Con esta declaración se genera un reporte que describe el contenido del archivo especificado y se escribe.

PRINT { BORINGS  
SOUNDINGS } { ALL  
CONTOURS }  
 GROUP 'nombre' { ALL  
FROM linbest TO linbest  
AT linbest }

30. - Declaración ADD GROUP

Esta declaración añade información a los datos almacenados y tiene la forma

ADD TO { BORINGS  
SOUNDINGS }  
CONTOURS }  
 GROUP 'nombre'  
 { linbest (NAME) dist dist sup dist. sup . . . }  
 { linbest dist dist. sup dist dist. sup . . . }

END

31. - Declaración CHANGE GROUP

Sirve para cambiar información almacenada y se expresa con

```

CHANGE (POINT) (IN) {
    BORINGS
    SOUNDINGS
    CONTOURS
}

GROUP 'nombre'
{
  Linbest (NAME) 'id' dist dist. sup dist. sup . . .
  Linbest list. dist. sup dist dist. sup dist dist. sup . . .
}

END
    
```

32. - Declaración DELETE GROUP

Se utiliza para borrar información almacenada y se escribe como

```

DELETE {
    BORINGS
    SOUNDINGS
    CONTOURS
} {
    ITEMS
    FULLY
}

GROUP 'nombre' {
    ITEMS
    FULLY
}

{
  linbest
  linbest dist dist dist . . .
  linbest FROM dist TO dist
  FROM linbest TO linbest
  FROM linbest TO linbest FROM dist TO dist
}

END
    
```

Las declaraciones 33 a 56 se utilizan para editar selectivamente archivos y son las siguientes:

33. - Declaración EDIT DATA

Los archivos de datos del terreno tanto de la superficie como del subsuelo se editan para verificar su lógica y valores. La forma general de la declaración es

EDIT DATA, tipo de archivo, límites estación

en donde

- tipo de archivo =
- SURFACE
  - SUBSURFACE
  - NOTEBOOK
  - BORINGS
  - SOUNDINGS
  - CONTOURS
  - RAW (FILES)
  - FULL (MODEL)
  - ALL

Para SURFACE, RAW FILES, FULL MODEL Y ALL

- límites estación =
- ALL
  - FROM linbest TO linbest
  - AT linbest

Para NOTEBOOK

límites estación =  $\left\{ \begin{array}{l} \underline{ALL} \\ \underline{FROM} \text{ localidad } \underline{TO} \text{ localidad} \\ \underline{AT} \text{ localidad} \end{array} \right\}$

Las declaraciones 43 a 36 se usan para pruebas de datos de la superficie como distancias y elevaciones o para secciones transversales parcialmente reducidas.

34. - Declaración CONTOUR INTERVAL

Tiene la forma

$\underline{CONTOUR} \ (\underline{INTERVAL}) \ \left\{ \begin{array}{l} \underline{ON} \\ \underline{OFF} \end{array} \right\} \vee$

e indica cuando la diferencia en elevación entre dos puntos adyacentes de una sección transversal no es igual al intervalo de contorno especificado.

35. - Declaración OFFSET SPACING

Se escribe como

$\underline{OFFSET} \ (\underline{SPACING}) \ \left\{ \begin{array}{l} \underline{ON} \\ \underline{OFF} \end{array} \right\} \ (\underline{MAXIMUM}) \vee \ (\underline{MINIMUM}) \vee$

informa siempre que se viola el criterio de la distancia entre dos elevaciones adyacentes de una sección transversal.

36. - Declaración POINT DENSITY

Se expresa con

$\underline{POINT} \ (\underline{DENSITY}) \ \left\{ \begin{array}{l} \underline{ON} \\ \underline{OFF} \end{array} \right\} \ (\underline{MAXIMUM}) \vee \ (\underline{MINIMUM}) \vee$

e indica siempre que se viola el criterio del número de puntos en una sección transversal.

Las declaraciones 37 a 39 se utilizan para pruebas de la entrada de datos de superficie cuando se pide la edición de

SURFACE, RAW FILES y FULL MODEL

37. - Declaración GROUND ELEVATION

Tiene la forma

$\underline{GROUND} \ \underline{ELEVATION} \ \left\{ \begin{array}{l} \underline{ON} \\ \underline{OFF} \end{array} \right\} \ (\underline{MAXIMUM}) \vee \ (\underline{MINIMUM}) \vee$

e informe cuando no se cumple el criterio de elevación de un punto que representa a la superficie del terreno.

38. - Declaración GROUND ELEVATION CHANGE

Se escribe como

$\underline{GROUND} \ \underline{ELEVATION} \ \underline{CHANGE} \ \left\{ \begin{array}{l} \underline{ON} \\ \underline{OFF} \end{array} \right\} \ (\underline{MAXIMUM}) \vee \ (\underline{MINIMUM}) \vee$

e indica cuando se viola la especificación de la diferencia de elevación en valor absoluto entre dos puntos adyacentes de la superficie de una sección transversal.

## 39. - Declaración GROUND SLOPE

Se expresa con

$$\underline{\text{GROUND SLOPE}} \left\{ \begin{array}{c} \underline{\text{ON}} \\ \underline{\text{OFF}} \end{array} \right\} ( [\underline{\text{MAXIMUM}}] v ) ( [\underline{\text{MINIMUM}}] v )$$

e informe siempre que la pendiente en valor absoluto de la superficie entre dos puntos adyacentes de una sección transversal viola el criterio establecido.

Las declaraciones 40 a 43 son pruebas aplicables sólo a modelos de terreno multimateriales totalmente construidos.

## 40. - Declaración STATIONING INTERVAL

Tiene la forma

$$\underline{\text{STATIONING (INTERVAL)}} \left\{ \begin{array}{c} \underline{\text{ON}} \\ \underline{\text{OFF}} \end{array} \right\} ( [\underline{\text{MAXIMUM}}] v ) ( [\underline{\text{MINIMUM}}] v )$$

y avisa cuando se viola el criterio especificado de la distancia entre estaciones de líneas base de dos secciones transversales consecutivas.

## 41. - Declaración MAXIMUM POINT OFFSET

Se escribe como

$$\underline{\text{MAXIMUM POINT (OFFSET)}} \left\{ \begin{array}{c} \underline{\text{BOTH}} \\ \underline{\text{LEFT}} \\ \underline{\text{RIGHT}} \end{array} \right\} \left\{ \begin{array}{c} \underline{\text{ON}} \\ \underline{\text{OFF}} \end{array} \right\} ( [\underline{\text{MAXIMUM}}] v ) ( [\underline{\text{MINIMUM}}] v )$$

e informa cuando la cola del punto extremo derecho y/o izquierdo de la sec

ción transversal excede la tolerancia indicada.

## 42. - Declaración DEPTH OF COVER OVER

Se expresa con

$$\text{DEPTH (OF) (COVER) (OVER)} \left\{ \begin{array}{c} \underline{\text{ALL}} \\ \text{'nombre mat'} \end{array} \right\} \left\{ \begin{array}{c} \underline{\text{ON}} \\ \underline{\text{OFF}} \end{array} \right\} ( [\underline{\text{MAXIMUM}}] v ) ( [\underline{\text{MINIMUM}}] v )$$

y reporta cuando las distancias verticales de una capa a otra o a la superficie excede los límites especificados.

## 43. - Declaración THICKNESS

Esta declaración tiene la expresión

$$\text{THICKNESS} \left\{ \begin{array}{c} \underline{\text{ALL}} \\ \text{'nombre mat'} \end{array} \right\} \left\{ \begin{array}{c} \underline{\text{ON}} \\ \underline{\text{OFF}} \end{array} \right\} ( [\underline{\text{MAXIMUM}}] v ) ( [\underline{\text{MINIMUM}}] v )$$

e indica cuando el espesor de una capa viola el criterio especificado.

Las declaraciones 44 a 48 se usan para pruebas opcionales de entrada de datos de superficie como notas de campo de secciones transversales no reducidas.

## 44. - Declaración HEIGHT OF INSTRUMENT CHANGE

Tiene la forma

$$\text{HEIGHT (OF) (INSTRUMENT) (CHANGE)} \left\{ \begin{array}{c} \underline{\text{ON}} \\ \underline{\text{OFF}} \end{array} \right\} ( [\underline{\text{MAXIMUM}}] v ) ( [\underline{\text{MINIMUM}}] v )$$

y reporta cuando se viola el criterio para la diferencia en valor absoluto de

elevaciones entre dos alturas de instrumento en dos rectas sucesivas.

#### 45. - Declaración ROD READING CHANGE

Esta declaración se expresa con

ROD (READING) CHANGE  $\left\{ \begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right\} ([\text{MAXIMUM}] v) ([\text{MINIMUM}] v)$

y avisa cuando se excede la especificación de la diferencia en valor absoluto entre lecturas rod de dos puntos adyacentes en una recta de datos de una sección transversal.

#### 46. - Declaración ROD READING OFFSET

Esta declaración se escribe con

ROD (READING) OFFSET  $\left\{ \begin{array}{c} \text{BOTH} \\ \text{LEFT} \\ \text{RIGHT} \end{array} \right\} \left\{ \begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right\} ([\text{MAXIMUM}] v) ([\text{MINIMUM}] v)$

y reporta cuando se viola la tolerancia para el valor absoluto de la distancia asociada con una lectura rod.

#### 47. - Declaración ORDERED OFFSETS

Tiene la forma

ORDERED (FIELD) NOTEBOOK) OFFSETS (REQUIRED)  $\left\{ \begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right\}$

e indica cuando las lecturas en dos lecturas rod adyacentes no se incrementan positivamente.

#### 48. - Declaración ORDERED STATIONING

Se expresa con

ORDERED (FIELD) (NOTEBOOK) STATIONING (REQUIRED)  $\left\{ \begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right\}$

e informa siempre que las estaciones de líneas base en líneas consecutivas de secciones transversales son decrecientes.

Las declaraciones 49 a 51 se utilizan para pruebas de perforación y son las siguientes:

#### 49. - Declaración BORING DENSITY

Esta declaración se escribe

BORING DENSITY  $\left\{ \begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right\} ([\text{MAXIMUM}] v) ([\text{MINIMUM}] v)$

y reporta cuando se viola la densidad de entrada de un grupo de perforaciones.

#### 50. - Declaración BORING SPACING

Tiene la forma

BORING SPACING  $\left\{ \begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right\} ([\text{MAXIMUM}] v) ([\text{MINIMUM}] v)$

e indica cuando la diferencia en distancia entre dos perforaciones adyacentes en una sección transversal excede la tolerancia especificada.

## 51. - Declaración MAXIMUM BORING OFFSET

Esta declaración se expresa con

$$\text{MAXIMUM BORING (OFFSET)} \left\{ \begin{array}{l} \text{BOTH} \\ \text{LEFT} \\ \text{RIGHT} \end{array} \right\} \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\} ([\text{MAXIMUM}] v) ([\text{MINIMUM}] v)$$

e informa cuando el valor absoluto de la distancia de un punto extremo en una sección transversal de perforaciones no cumple con el criterio especificado.

Las declaraciones 52 a 54 se aplican a pruebas con sonido y son las siguientes:

## 52. - Declaración SOUNDING DENSITY

Tiene la forma

$$\text{SOUNDING DENSITY} \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\} ([\text{MAXIMUM}] v) ([\text{MINIMUM}] v)$$

e indica siempre que la densidad de entrada de un grupo de pruebas con sonido no cumple el criterio especificado.

## 53. - Declaración SOUNDING SPACING

Esta declaración se escribe

$$\text{SOUNDING SPACING} \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\} ([\text{MAXIMUM}] v) ([\text{MINIMUM}] v)$$

y reporta cuando se excede la tolerancia en la diferencia de distancias de dos pruebas de sonido adyacentes en una sección transversal.

## 54. - Declaración MAXIMUM SOUNDING OFFSET

Se expresa con

$$\text{MAXIMUM SOUNDING (OFFSET)} \left\{ \begin{array}{l} \text{BOTH} \\ \text{LEFT} \\ \text{RIGHT} \end{array} \right\} \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\} ([\text{MAXIMUM}] v) ([\text{MINIMUM}] v)$$

y avisa cuando no se cumple la especificación del valor absoluto de la distancia de un punto extremo en una prueba de sonido en una sección transversal.

## 55. - Declaración PRINT SPECIFICATIONS

Se genera un reporte por medio de un listado de todas las pruebas opcionales consideradas. Tiene la forma general

PRINT (EDIT) SPECIFICATIONS

## 56. - Declaración DELETE SPECIFICATIONS

Con esta declaración se borran todas las especificaciones del tipo designado y tiene la forma

$$\text{DELETE (EDIT) SPECIFICATIONS (FOR)} \left\{ \begin{array}{l} \text{COMPLETELY} \\ \text{SURFACE (DATA)} \\ \text{NOTEBOOK} \\ \text{SUBSURFACE (DATA)} \\ \text{FULL (MODEL)} \end{array} \right\}$$

Las declaraciones 57 a 69 sirven para generar el modelo de terreno multi-material y son las siguientes:

57. - Declaración REDUCE NOTEBOOK

Tiene la forma

REDUCE (NOTEBOOK) { ALL  
FROM localidad TO localidad  
AT localidad }

58. - Declaración ADD SUBSURFACES

Se expresa con

ADD SUBSURFACES { ALL  
FROM linbest TO linbest  
AT linbest }

59. - Declaración CONSTRUCT MODEL

Se escribe como

CONSTRUCT (MODEL) { ALL  
FROM linbest TO linbest  
AT linbest }

en donde esta declaración es equivalente a ejecutar sucesivamente las declaraciones 57 y 58.

60. - Declaración ENABLE MODEL

Esta declaración se usa para regresar un modelo existente a su esta

do actualizado previo y tiene la forma

ENABLE (MODEL)

61. - Declaración MULTI MATERIAL TERRAIN MODEL

Esta declaración permite la entrada directa como datos tabulados de secciones transversales de modelos construidos de terreno multimaterial y zonas de materiales. Tiene la forma general

MULTI(MATERIAL) (TERRAIN) (MODEL), [BASELINE] 'nombre'

ZONE linbest linbest n 'nombre mat<sub>1</sub>' nombre mat<sub>2</sub>' . . . . (opcional)

linbest dist<sub>1</sub> elev<sub>2</sub> elev<sub>3</sub> . . . dist<sub>1</sub> elev<sub>2</sub> . . . dist<sub>n</sub> elev<sub>1</sub> . . . (opcional)

END

62. - Declaración EDIT FULL MODEL

Permite la edición del modelo construido de terreno multimaterial para la verificación de errores. Se expresa con

EDIT DATA, FULL (MODEL) { ALL  
FROM linbest TO linbest  
AT linbest }

en donde se tienen disponibles las pruebas opcionales siguientes:

EDIT SPECIFICATIONS

GROUND ELEVATION ( [MAXIMUM] v ) ( [MINIMUM] v )

GROUND ELEVATION CHANGE ( [MAXIMUM] v ) ( [MINIMUM] v )

GROUND SLOPE ( [MAXIMUM] v ) ( [MINIMUM] v )

STATIONING (INTERVAL) ( [MAXIMUM] v ) ( [MINIMUM] v )

MAXIMUM POINT (OFFSET) { BOTH  
LEFT  
RIGHT } ( [MAXIMUM] v ) ( [MINIMUM] v )

DEPTH (OF) (COVER) (OVER) { ALL  
'nombre mat' } ( [MAXIMUM] v ) ( [MINIMUM] v )

THICKNESS { ALL  
'nombre mat' } ( [MAXIMUM] v ) ( [MINIMUM] v )

END

Las declaraciones 63 a 66 se utilizan para la salida de información o datos y son

## 63. - Declaración PRINT MODEL

Se genera un reporte de distancias y elevaciones de todos los puntos de la superficie y subsuelo definidos en el modelo completo. Tiene la forma

PRINT MODEL { ALL  
FROM linbest TO linbest  
AT linbest }

## 64. - Declaración PRINT MATERIALS MODEL

Se describen las zonas de materiales y la configuración cualitativa del subsuelo asociadas con el modelo. Se expresa con

PRINT MATERIALS (IN) (MODEL) { ALL  
FROM linbest TO linbest  
AT linbest }

## 65.-- Declaración LIST MATERIALS

Con esta declaración se imprime una lista de los nombres de materiales asignados a la superficie y subsuelo. Se escribe con

LIST MATERIALS (DEFINED) (IN) (THE) (SYSTEM)

## 66. - Declaración DUMP MATERIAL MODEL

Con esta declaración se tiene salida por tarjeta perforada del modelo de terreno multimaterial completo dentro de los límites de líneas base designadas.

Tiene la forma

DUMP (FULL) (MULTI) (MATERIAL) (TERRAIN) (MODEL) { ALL  
FROM linbest TO linbest }

Las declaraciones 67 a 69 sirven para corregir de una manera selectiva el modelo construido y son

## 67. - Declaración ADD MODEL

Permite añadir información y tiene la forma



ADD TO MODEL

linbest dist<sub>1</sub> elev<sub>1</sub> elev<sub>2</sub> . . . dist<sub>2</sub> elev<sub>1</sub> . . . dist<sub>n</sub> elev<sub>1</sub> . . .

⋮

END

## 68. - Declaración CHANGE MODEL

Sirve para cambiar datos del modelo y se expresa con

CHANGE (POINT) (IN) MODEL

linbest dist<sub>1</sub> elev<sub>1</sub> elev<sub>2</sub> . . . dist<sub>2</sub> elev<sub>1</sub> . . . dist<sub>n</sub> elev<sub>1</sub> . . .

⋮

END

## 69. - Declaración DELETE MODEL

Se utiliza para borrar información del modelo y se escribe

DELETE MODEL { ITEMS }  
                          { FULLY }

{  
  linbest  
  linbest FROM dist TO dist  
  linbest dist dist dist . . .  
  FROM linbest TO linbest, (FROM dist TO dist)  
}

END

*notas sobre el sistema  
integrado de ingeniería civil*

**subsistema**

**ICES-SLOPE**

*Dr. Rodolfo Luths G.*

Este subsistema permite la solución de problemas de estabilidad de taludes o pendientes. El factor de seguridad se calcula por un método aproximado de la técnica de equilibrio límite desarrollada por Bishop y también utilizando el método de franjas de Fellenius.

Con este subsistema se puede considerar cualquier sección transversal de suelo, aunque esta limitado a arcos circulares de falla. Sin embargo, no hay límite realmente al número de puntos, rectas y suelos que pueden usarse para definir un problema.

Se analizan círculos de prueba con centro abajo de la cresta y con radios de gran dimensión tales que el círculo de falla "cuelga", produciéndose en estos casos una grieta en donde el círculo corta la vertical. Cuando se analiza una pendiente irregular, es posible que un círculo de prueba defina una masa deslizante compuesta de dos o más partes no conectadas, suponiéndose y calculándose el factor de seguridad que corresponde a la parte de mayor tamaño.

La experiencia más reciente justifica al método de Bishop como el mejor de los procedimientos sencillos para superficies circulares de falla.

Las declaraciones que se utilizan en este subsistema son las siguientes:

1. Declaración SLOPE

Tiene la forma SLOPE

\*El material que se presenta se ha tomado del manual R69-22, Slope Stability Analysis, de William A. Barley y John T. Christian, publicado por Soil Mechanics Division and Civil Engineering Systems Laboratory, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, Mass.

y se usa al principio de cada problema en el que se aplique este subsistema.

2. Declaración SAVE

Se presenta en la forma

SAVE 'nombre archivo'

en donde

'nombre de archivo' es la identificación del archivo no pudiendo exceder de 8 caracteres.

Con esta declaración se almacenan datos de problemas para poder usarlos nuevamente en el futuro.

3. Declaración RESTORE

Se expresa con

RESTORE 'nombre de archivo'

y permite hacer referencia y uso de información almacenada con la declaración previa.

4. Declaración EJECT

Tiene la forma

EJECT

y produce que se avance a una nueva página para efectuar la impresión de datos, lo que puede ser deseable para separar los distintos problemas en

3.

una corrida.

### 5. Declaración FINISH

Se presenta en la forma

FINISH

e indica el término del último problema de una corrida. Cuando se continúa con otro problema del mismo subsistema se puede emplear la declaración 4 y en el caso de problemas de otro subsistema se sustituye la declaración FINISH por la declaración del nombre del nuevo subsistema.

Las declaraciones 6 y 7 se aplican a puntos y rectas observando que los puntos en los extremos de segmentos lineales se usan para definir el perfil de la pendiente y localizar los distintos suelos en un talud.

### 6. Declaración POINT

Se expresa con

$$\underline{\text{POINT}} \quad i \quad \left[ \begin{array}{c} X \\ \hline \end{array} \right] \quad v_x \quad \left[ \begin{array}{c} Y \\ \hline \end{array} \right] \quad v_y$$

en donde

$i$  indica el punto

$v_x$  es el valor de la coordenada X

$v_y$  es el valor de la coordenada Y

Las unidades son pies y todas las coordenadas X deben ser positivas.

4.

### 7. Declaración LINE

Se escribe en la forma

$$\underline{\text{LINE}} \quad i_1 \quad \left[ \begin{array}{c} P \\ \hline \end{array} \right] \quad i_2 \quad \left[ \begin{array}{c} P \\ \hline \end{array} \right] \quad i_3 \quad \left[ \begin{array}{c} S \\ \hline \end{array} \right] \quad i_4$$

en donde

$i_1$  especifica el número de la recta

$i_2, i_3$  indican los puntos extremos de esa recta

$i_4$  es el número de la capa de suelo inmediatamente abajo de la recta.

### 8. Declaración SOIL

Con esta declaración se almacenan las propiedades de un suelo y tiene la forma

$$\underline{\text{SOIL}} \quad i_1 \quad \left[ \begin{array}{c} D \\ \hline \end{array} \right] \quad i_2 \quad \left[ \begin{array}{c} C \\ \hline \end{array} \right] \quad i_3 \quad \left[ \begin{array}{c} PHI \\ \hline \end{array} \right] \quad i_4 \quad \left[ \begin{array}{c} RU \\ \hline \end{array} \right] \quad v_1 \quad \left[ \begin{array}{c} RC \\ \hline \end{array} \right] \quad v_2$$

en donde

$i_1$  es el número que identifica al suelo

$i_2$  especifica el valor de la densidad (libras/pte cúbico)

$i_3$  indica la cohesión (libras/pte cuadrado)

$i_4$  es el ángulo de fricción (grados)

$v_1$  es el cociente de la presión de poro en un punto dividido entre el pe

so de una columna de suelo de sección transversal unitaria, pudiendo proporcionarse también el número de la superficie freática de donde se calcula la presión.

$v_2$  indica el cociente de presión de poro en la zona capilar, o sea, el nivel piezométrico en la zona capilar expresado como fracción decimal de la distancia vertical sobre la superficie freática.

#### 9. Declaración PIEZ

Con esta declaración se define una superficie piezométrica que puede usarse con cualquier tipo de suelo. Las presiones de poro se pueden calcular utilizando superficies piezométricas. Una superficie piezométrica se define conectando por lo menos dos puntos piezométricos como entrada y tiene la forma general

$$\text{PIEZ } i_1 \left[ \frac{N}{\quad} \right] i_2 \left[ \frac{X}{\quad} \right] v_x \left[ \frac{Y}{\quad} \right] v_y$$

en donde

$i_1$  identifica a la superficie piezométrica

$i_2$  es el número del punto que se almacena en la superficie piezométrica.

$v_x$  indica la coordenada X del punto

$v_y$  es la coordenada Y del punto

5.

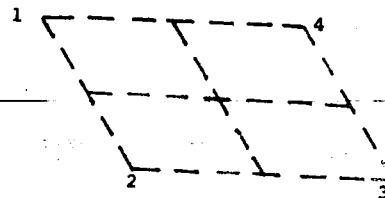
Una línea de filtración se indica con el número 1 para la superficie piezométrica y se usa para definir un punto sobre la pendiente o el pie del talud si están sumergidos en un cuerpo de agua.

#### 10. Declaración GRID

Esta declaración tiene la forma

$$\text{GRID } 1 \left[ \frac{X}{\quad} \right] v_x \left[ \frac{Y}{\quad} \right] v_y \quad 2 \left[ \frac{X}{\quad} \right] v_x \left[ \frac{Y}{\quad} \right] v_y \quad 3 \left[ \frac{X}{\quad} \right] v_x \left[ \frac{Y}{\quad} \right] v_y \quad i_1 \quad i_2$$

y define un paralelogramo con las coordenadas de los tres puntos, siendo el punto 2 la intersección de las rectas 12 y 23, como muestra la figura



Se forma una retícula con los números  $i_1, i_2$  que indican los números de segmentos en los que se dividen las rectas 12 y 23, respectivamente. En la figura se observan 9 intersecciones que serán los centros de círculos de prueba que se investigarán si  $i_1, i_2$  valen 2 cada uno. Para cada centro se tienen radios que se encuentran dentro de la sección transversal de finida.

#### 11. Declaración BEGIN

Con esta declaración se inicia un procedimiento de búsqueda de un mínimo

6.

relativo del factor de seguridad. Tiene la forma general

$$\underline{\text{BEGIN AT}} \left[ \underline{X} \right] v_x \left[ \underline{Y} \right] v_y \left[ \underline{\text{DELX}} \right] d_x \left[ \underline{\text{DELY}} \right] d_y$$

e inicia la búsqueda en el punto de coordenadas  $(v_x, v_y)$  con incrementos  $d_x, d_y$  en las direcciones X e Y, respectivamente. Cuando se encuentra un mínimo relativo, los incrementos se dividen entre 4 y se continúa la búsqueda hasta localizar un mínimo relativo con estos nuevos incrementos.

#### 12. Declaración DO

Con esta declaración se calcula el factor de seguridad para un centro y radio de círculo dados. Tiene la forma

$$\underline{\text{DO ONLY}} \left[ \underline{X} \right] v_x \left[ \underline{Y} \right] v_y \left[ \underline{R} \right] r$$

en donde  $(v_x, v_y)$  son las coordenadas del centro y  $r$  es el valor del radio del círculo de prueba.

Las declaraciones 13 a 16 permiten limitar el valor de los radios de círculos utilizados en el cálculo de factores de seguridad

#### 13. Declaración RMAX THRU POINT

Esta declaración tiene la forma

$$\underline{\text{RMAX THRU POINT}} \quad i_1 \quad i_2 \quad i_3 \quad i_4 \quad i_5$$

y limita el radio máximo para cualquier centro dado al máximo radio que pasa por los puntos  $i_1$  a  $i_5$ . El máximo número de puntos que pueden considerarse es de cinco.

#### 14. Declaración RMIN THRU POINT

Se expresa con

$$\underline{\text{RMIN THRU POINT}} \quad i_1 \quad i_2 \quad i_3 \quad i_4 \quad i_5$$

y limita el radio máximo para círculos con un centro dado al máximo de los radios que pasan por los puntos  $i_1$  a  $i_5$ .

#### 15. Declaración RMAX TANGENT LINE

Esta declaración se escribe

$$\underline{\text{RMAX TANGENT LINE}} \quad i_1 \quad i_2 \quad i_3 \quad i_4 \quad i_5$$

y limita al radio máximo para cualquier centro de círculo dado al menor de los radios tangente a las rectas  $i_1$  a  $i_5$ .

#### 16. Declaración RMIN TANGENT LINE

Tiene la forma

$$\underline{\text{RMIN TANGENT LINE}} \quad i_1 \quad i_2 \quad i_3 \quad i_4 \quad i_5$$

y limita al radio mínimo para cualquier círculo con centro dado al mayor de los radios tangente a las rectas  $i_1$  a  $i_5$ .

#### 17. Declaración NUMBER OF SLICES

Se expresa con

$$\underline{\text{NUMBER OF SLICES}}$$

en donde  $n$  es el número de franjas en las que se dividirá aproximadamente la sección transversal del suelo. Si no se proporciona este número se consideran automáticamente un número aproximado de 30 franjas.

#### 18. Declaración PRINT

Esta declaración permite la impresión de los elementos que se indican y tiene la forma

PRINT	}	<u>POINT</u> <u>LINE</u> <u>SOIL</u> <u>PIEZ</u> <u>ALL</u>	}
-------	---	---	---

Las declaraciones 10, 11 y 12 producen la impresión del arreglo de las -  
 fronteras del suelo. Para cada centro de prueba, las declaraciones 10 y 11  
 permiten el cálculo de los factores de seguridad de Bishop y Fellenius en  
 el rango de radios, proporcionándose el número de franjas para cada radio.  
 Se imprimen al final de los cálculos de las declaraciones anteriores, el fac-  
 tor de seguridad mínimo, las coordenadas del centro del círculo y el valor  
 del radio correspondientes a ese factor de seguridad. Los resultados que  
 se imprimen con la declaración 12 corresponden únicamente al círculo cal-  
 culado.

*notas sobre el sistema  
integrado de ingeniería civil*

**subsistema  
ICES-SEPOL**

*Dr. Rodolfo Luthe G.*



El subsistema SEPOL analiza las características de asentamiento de perfiles de suelos, pudiendo ser útil para el diseño de edificios, presas de tierra, terraplenes para carreteras y problemas similares. Se pueden investigar los asentamientos afectados por sistemas de carga simples o complejos, que pueden ser constantes o variables; variaciones de las propiedades de los suelos con la profundidad, variaciones en las condiciones del nivel freático y la incertidumbre en los aspectos antes mencionados.

El perfil de un suelo se puede describir en forma de capas o franjas (estratificado) o como un cuerpo semi-infinito (uniforme). En el caso de capas, el nivel freático se debe especificar en alguna frontera entre capas y cualquier propiedad debe ser constante en una capa dada, pudiendo tenerse prácticamente cualquier número de capas.

El caso de un perfil semi-infinito se considera como una sola capa con propiedades constantes, exceptuando aquellas que varían con las condiciones del nivel freático, en cuyo caso se especifica el nivel freático y las condiciones en la parte superior e inferior del mismo.

Se tiene la libertad de utilizar cualquier sistema de unidades, con la condición de ser consistente para un problema considerado.

Las declaraciones disponibles son más de 200 y se clasifican dentro de cuatro grupos de funciones:

\* El material que se presenta se ha tomado del manual 107-61, ICES SEPOL-I, A Settlement Problem Oriented Language, de J.C. Jordan y R.L. Schiffman, publicado por M.I.T. Press, Cambridge, Mass.

- I. Función INSITU
- II. Función STRESS DISTRIBUTION
- III. Función SETTLEMENT
- IV. Función RATE

La secuencia normal de las funciones es en el mismo orden que el antes presentado, pero cada función y las declaraciones que le corresponden pueden usarse independientemente.

Existen algunas declaraciones generales aplicables en cualquiera de las funciones anteriores y son las siguientes:

Declaración SEPOL

Tiene la forma

SEPOL

y es la primera declaración que debe usarse en todos los problemas y que identifica al subsistema.

Declaración TIME

Se presenta con

TIME

y se usa para los datos de carga variable y también para indicar en que instantes de la velocidad de asentamiento se desea la salida.

Declaración EJECT

Se usa

R

La declaración tiene la forma

EJECT

y ocasiona la impresión de la declaración que sigue en una nueva página.

Declaración FINISH

Se expresa como

FINISH

y es la última declaración que debe aparecer en todos los problemas de este subsistema.

Se tiene

$$\left\{ \begin{array}{c} Z \\ DEPTH \end{array} \right\}$$

para indicar la dirección y posición perpendicular a la superficie del suelo a la que se desea efectuar algún cálculo. En todos los casos Z puede sustituir a DEPTH pero la recíproca no es siempre válida.

Se utiliza

$$\left\{ \begin{array}{c} X \\ Y \end{array} \right\}$$

para indicar dos dimensiones al representar cargas rectangulares, adicionalmente a la dirección perpendicular de profundidad.

para representar el caso de carga circular.

#### 1. Función INSITU

El objeto de esta función es proporcionar las declaraciones necesarias para que el usuario calcule o proporcione como entrada varios valores de propiedades del suelo para perfiles estratificados o semi-infinitos. También se pueden calcular las presiones efectivas y los esfuerzos totales y efectivos en las direcciones horizontal y vertical a cualquier profundidad en el perfil del suelo, en su estado natural anterior al de aplicación de sistemas de carga.

No es necesario especificar la función INSITU cada vez que se proporcionan datos adicionales, pudiendo usarse las declaraciones correspondientes en cualquiera de las otras funciones, pero únicamente para proporcionar datos de entrada.

Los valores de propiedades del suelo proporcionados como entrada o calculados con la función INSITU, están disponibles automáticamente en las tres funciones restantes.

Los esfuerzos de tensión se indican anteponiendo al valor numérico un signo menos.

Las declaraciones de inicialización son:

SEPOL

INSITU (SOIL PROFILE)

$$\left\{ \begin{array}{l} \text{LAYERED SYSTEM } m \\ \text{SEMI (INFINITE PROFILE)} \end{array} \right\}$$

en donde  $m$  indica el número de capas o franjas de suelo.

Las declaraciones 1 a 23 se aplican a propiedades de suelos y son las siguientes:

## 1. Declaración WATER UNIT

Se presenta con

WATER UNIT (WEIGHT) v

y define el peso unitario del agua, en donde  $v$  indica el valor numérico de la propiedad mencionada.

## 2. Declaración WATER TABLE

Tiene la forma

WATER TABLE v

y define la profundidad del nivel freático.

Las declaraciones 1 y 2 sólo pueden utilizarse para proporcionar datos de entrada. Las declaraciones 3 a 6 se usan para datos de entrada o impresión, pero en el caso de impresión se debe anteponer a la declaración la palabra PRINT.

## 3. Declaración LAYER THICKNESS

Se escribe con

LAYER THICKNESS formato

en donde para entrada o INPUT se tiene

$$\text{formato} = \left\{ \begin{array}{l} (\text{LAYER}) \ n \ (\text{VALUE}) \ v \\ v \\ A \ v \\ B \ v \\ A \ v \ B \ v \end{array} \right\}$$

y para solución, impresión o ambas, es decir, SOLVE, PRINT, o SOLVE/PRINT, se usa

$$\text{formato} = \left\{ \begin{array}{l} n_1 \dots n_m \\ (\text{ningún formato}) \\ A \\ B \\ A \ B \end{array} \right\}$$

en donde

- $n$  indica el número de la capa
- $m$  indica el número total de capas
- $v$  es el valor numérico de la propiedad especificada
- $A$  representa la parte superior al nivel freático

B representa la parte inferior al nivel freático

La primera alternativa en ambos formatos sólo puede usarse para suelos estratificados.

#### 4. Declaración PORE PARAMETER

Se expresa como

A PORE PARAMETER formato

e indica el parámetro de presión efectiva de cada capa, en donde formato es como se indicó antes.

#### 5. Declaración COMPRESSION INDEX

Tiene la forma

COMPRESSION INDEX formato

y se define como la pendiente de la curva semilogarítmica presión-relación de vacíos.

El formato es como se indicó en la declaración 3. Los valores en esta declaración pueden variar con el nivel freático y pueden usarse las alternativas A y B en perfiles semi-infinitos. Esta observación se aplica también a la declaración 6.

#### 6. Declaración UNDRAINED MODULUS.

Se escribe

UNDRAINED (MODULUS) formato

y define la relación esfuerzo deformación (situación de impermeabilidad).

El formato se explica en la declaración 3.

Las declaraciones 7 a 23 pueden usarse para entrada de datos, impresión, solución o impresión y solución. En estos casos a la declaración se le anteponen PRINT, SOLVE o SOLVE/PRINT.

#### 7. Declaración SPECIFIC GRAVITY

Tiene la forma

SPECIFIC GRAVITY formato

y define la gravedad específica de cada capa, siendo formato como se indicó en la declaración 3.

En las declaraciones 8 a 10, los valores pueden variar con el nivel freático y pueden usarse las alternativas A y B en perfiles semi-infinitos.

#### 8. Declaración INITIAL WATER CONTENT

Con esta declaración se especifica el contenido inicial de agua de cada capa y se escribe

INITIAL WATER (CONTENT) formato

en donde formato se explicó antes.

#### 9. Declaración FINAL WATER CONTENT

Se especifica el contenido final de agua de cada capa y se expresa con

FINAL WATER (CONTENT) formato

en donde formato se definió antes.

10. Declaración SATURATION

Tiene la forma

SATURATION formato

e indica el grado de saturación de cada una de las capas.

11. Declaración INITIAL VOID RATIO

Se expresa con

INITIAL VOID (RATIO) formato

y define la relación de vacíos inicial para cada capa.

12. Declaración FINAL VOID RATIO

Tiene la forma

FINAL VOID (RATIO) formato

y define la relación de vacíos final para cada una de las capas.

13. Declaración VOID RATIO INCREMENT

Se escribe

VOID (RATIO INCREMENT) formato

e indica el incremento en los valores de la relación de vacíos.

14. Declaración POISSON RATIO

Define el cociente de esfuerzo horizontal entre esfuerzo vertical y se escribe

POISSON (RATIO) formato

En las declaraciones 15 a 17, los valores pueden variar con el nivel freático y pueden utilizarse las alternativas A y B para perfiles semi-infinitos.

15. Declaración SHEAR MODULUS

Se expresa con

SHEAR (MODULUS) formato

y especifica la resistencia al cortante de cada capa.

16. Declaración YOUNG MODULUS

Se escribe

YOUNG (MODULUS) formato

y representa la relación de esfuerzo entre deformación.

17. Declaración TOTAL UNIT WEIGHT

Se presenta como

TOTAL UNIT (WEIGHT) formato

e indica el peso unitario total de cada capa.

## 18. Declaración DRY UNIT WEIGHT

Tiene la forma

DRY UNIT WEIGHT formato

y especifica el peso unitario seco de cada capa.

## 19. Declaración BUOYANT UNIT WEIGHT

Se escribe

BUOYANT UNIT WEIGHT formato

y presenta el peso unitario flotante de cada capa.

## 20. Declaración PERMEABILITY

Tiene la forma

PERMEABILITY formato

y especifica la permeabilidad de cada capa.

## 21. Declaración COMPRESSIBILITY

Se presenta como

COMPRESSIBILITY formato

e indica la pendiente de la secante de la curva presión-relación de vacíos, para un incremento de presión dado.

## 22. Declaración VOLUME COMPRESSIBILITY

Se escribe

VOLUME COMPRESSIBILITY formato

y especifica el cambio de volumen debido a un incremento de presión.

## 23. Declaración COEFFICIENT OF CONSOLIDATION

Se expresa con

COEFFICIENT (OF CONSOLIDATION) formato

y es el coeficiente para indicar la razón de cambio de volumen en un suelo.

Las declaraciones 24 a 29 se aplican a esfuerzos y son:

## 24. Declaración INSITU EFFECTIVE VERTICAL STRESS

Tiene la forma

INSITU EFFECTIVE VERTICAL (STRESS) formato<sub>1</sub>

y calcula las presiones intergranulares verticales a la profundidad especificada. Además, se tiene

$$\text{formato}_1 = \underline{AT} \left\{ \begin{array}{l} z \left\{ \begin{array}{l} z_1, z_2, \dots \\ \text{TABLE 'm'} \end{array} \right\} \\ \text{DEPTH} \left\{ \begin{array}{l} z_1, z_2, \dots \\ \text{TABLE 'm'} \end{array} \right\} \end{array} \right\}$$

## 25. Declaración INSITU EFFECTIVE HORIZONTAL STRESS

Se escribe

INSITU EFFECTIVE HORIZONTAL (STRESS) formato<sub>1</sub>

semejante a la declaración anterior pero en la dirección horizontal.

## 26. Declaración PORE WATER PRESSURE

Se expresa con

PORE (WATER PRESSURE) formato<sub>1</sub>

y especifica la carga hidráulica bajo el nivel freático a la profundidad indicada.

## 27. Declaración INSITU TOTAL VERTICAL STRESS

Se presenta en la forma

INSITU TOTAL VERTICAL (STRESS) formato<sub>1</sub>

y representa la suma de la presión vertical efectiva y la de la carga hidráulica.

## 28. Declaración INSITU TOTAL HORIZONTAL STRESS

Se expresa con

INSITU TOTAL HORIZONTAL (STRESS) formato<sub>1</sub>

y representa la suma de la presión horizontal efectiva y la de carga hidráulica.

## 29. Declaración ALL

Se escribe simplemente

ALL formato<sub>1</sub>

y ejecuta las declaraciones 24 a 28.

## II. Función STRESS DISTRIBUTION

Esta función proporciona la posibilidad de calcular esfuerzos, deformaciones y desplazamientos en un punto, suponiendo sólidos homogéneos, isotrópicos, semi-infinitos y de elasticidad lineal. La superficie de carga se describe según su geometría, y puede ser rectangular, circular o de deformación plana. Antes de cargarse, se supone que el suelo no tiene peso ni está sujeto a esfuerzos.

Las declaraciones de inicialización son

STRESS DISTRIBUTIONSTRESS SOIL (PROFILE SEMI INFINITE)

## 30. Declaración SOLVE, PRINT

Se escriben

$$\left. \begin{array}{l} \text{SOLVE} \\ \text{PRINT} \\ \text{SOLVE/PRINT} \end{array} \right\} \text{componente}$$

en donde componente para el caso de constante elástica, es

componente =  $\left. \begin{array}{l} \text{YOUNG} \\ \text{SHEAR} \\ \text{POISSON} \end{array} \right\} \begin{array}{l} (\text{MODULUS}) \\ (\text{RATIO}) \end{array}$

## 31. Declaración LOAD TYPE

Tiene la forma

LOAD TYPE geometría carga

en donde

geometría carga =  $\left\{ \begin{array}{l} \text{PLANE (STRAIN LOADING)} \\ \text{AXIAL (SYMMETRIC LOADING)} \\ \text{RECTANGULAR (LOADING)} \end{array} \right\}$

Las declaraciones 32 a 34 describen los formatos que describen a la geometría de cargas anterior.

## 32. Declaración PLANE STRAIN LOADING

Para esta declaración se presentan las alternativas

STRIP [(EDGE)] v [(WIDTH)] v [(P)] v  
RAMP [(EDGE)] v [(WIDTH)] v [(P)] v  
INVERSE (RAMP) [(EDGE)] v [(WIDTH)] v [(P)] v  
SEMI (INFINITE) STRIP [(EDGE)] v [(P)] v  
ANY (P LOAD) [(EDGE)] v [(WIDTH)] v  
PX [(X)] v [(P)] v

elementos de proceso =

$\left. \begin{array}{l} \text{SIGMA } \underline{XX} \\ \text{SIGMA } \underline{YY} \\ \text{SIGMA } \underline{ZZ} \\ \text{SIGMA } \underline{XZ} \\ \text{SIGMA } \underline{1} \\ \text{SIGMA } \underline{2} \\ \text{SIGMA } \underline{3} \\ \text{TAUMAX} \\ \text{SIGMA AVERAGE} \\ \text{INITIAL (EXCESS PORE PRESSURE)} \\ \text{PRINCIPAL (DIRECTION) } \underline{ONE} \end{array} \right\}$

## 33. Declaración AXIAL SYMMETRIC LOADING

Para esta declaración la carga se resume con la expresión

CIRCULAR (LOAD) [(RADIUS)] v [(P)] v [(X)] v [(Y)] v

## 34. Declaración RECTANGULAR LOADING

Para la declaración de carga rectangular se tiene el formato

NORMAL (LOAD) [(LENGTH)] v [(WIDTH)] v [(P)] v [(X)] v [(Y)] v

Las declaraciones 35 a 37 se aplican a los elementos de proceso básicos de la geometría de cargas indicada en la declaración 31 y son

## 35. Declaración PLANE STRAIN LOADING

Para esta declaración, se tiene



elementos de proceso

PRINCIPAL (DIRECTION) TWO  
EPSILON XX  
EPSILON ZZ  
EPSILON XZ  
HORIZONTAL (DISPLACEMENT)  
STRESS  
STRAIN  
PRINCIPAL STRESS  
ALL PRINCIPAL (DIRECTION)  
STRESS / STRAIN  
ALL COMPONENTES

## 36. Declaración AXIAL SYMMETRIC LOADING

Para esta declaración, se tiene

elementos de proceso

SIGMA RR  
SIGMA TT  
SIGMA ZZ  
SIGMA RZ  
SIGMA 1  
SIGMA 2  
SIGMA 3  
TAUMAX  
SIGMA AVERAGE  
INITIAL (EXCESS PORE PRESSURE)

elementos de proceso

PRINCIPAL (DIRECTION) ONE  
PRINCIPAL (DIRECTION) TWO  
EPSILON RR  
EPSILON TT  
EPSILON ZZ  
EPSILON RZ  
RADIAL (DISPLACEMENT)  
VERTICAL (DISPLACEMENT)  
STRESS  
STRAIN  
PRINCIPAL STRESS  
STRESS/STRAIN  
ALL PRINCIPAL (DIRECTION)  
ALL COMPONENTS

## 37. Declaración RECTANGULAR LOADING

En este caso, se tiene

elementos de proceso

SIGMA XX  
SIGMA YY  
SIGMA ZZ  
SIGMA XY  
SIGMA YZ  
SIGMA 1  
SIGMA 2  
SIGMA 3

elementos de proceso :

- TAUMAX
- SIGMA AVERAGE
- INITIAL (EXCESS PORE PRESSURE)
- PRINCIPAL (DIRECTION) ONE
- PRINCIPAL (DIRECTION) TWO
- PRINCIPAL (DIRECTION) THREE
- EPSILON XX
- EPSILON YY
- EPSILON ZZ
- EPSILON XY
- EPSILON XZ
- EPSILON YZ

También se aceptarían los últimos 7 renglones de la declaración 36.

III. Función SETTLEMENT

Con esta función se calcula la magnitud de asentamiento de la superficie en uno o más puntos. Se tienen las declaración siguientes para la inicialización.

- SETTLEMENT
- INITIAL SETTLEMENT
- CONSOLIDATION SETTLEMENT

Para la declaración INITIAL SETTLEMENT se tienen las declaraciones

38 a 41.

38. Declaración INITIAL SETTLEMENT

Esta declaración se describe con alguna de las declaraciones

- LAYERED INITIAL SETTLEMENT (PROFILE) n
- SEMI (INFINITE) INITIAL SETTLEMENT (PROFILE)

39. Declaración SOLVE, PRINT INITIAL SETTLEMENT

Se tienen las alternativas

- SOLVE
  - SOLVE/PRINT
- INITIAL (AT) descripción

CALCULATE (AT) [LAYER] n [Z] v

END

en donde descripción proporciona las coordenadas en los lugares en los - - que se calculan los desplazamientos.

40. Declaración SEMI INFINITE INITIAL SETTLEMENT PROFILE

En solución o impresión para este caso, se tiene

- SOLVE
  - SOLVE/PRINT
- INITIAL
  - INFINITE
  - FINITE(AT) descripción

en donde si se usa FINITE se debe continuar con

CALCULATE (FROM) Z v TO v

41. Declaración PRINT INITIAL SETTLEMENT

Se tiene la declaración

PRINT INITIAL (SETTLEMENT) (AT) descripción

Para la declaración CONSOLIDATION SETTLEMENT se aplican las declaraciones 42 a 46.

42. Declaración CONSOLIDATION SETTLEMENT

La Declaración anterior se describe con alguna de las declaraciones

LAYERED CONSOLIDATION SETTLEMENT (PROFILE) n

SEMI (INFINITE) CONSOLIDATION (SETTLEMENT) (PROFILE)

43. Declaración SOLVE, PRINT CONSOLIDATION

Se tienen las alternativas

$\left\{ \begin{array}{l} \text{SOLVE} \\ \text{SOLVE/PRINT} \end{array} \right\}$  CONSOLIDATION método (AT) descripción

CALCULATE (AT) [LAYER] n [Z] v

END

en donde

método =  $\left\{ \begin{array}{l} \text{ONE (DIMENSIONAL)} \\ \text{ELASTIC} \\ \text{PRINCIPAL (STRESS)} \end{array} \right\}$   $\left\{ \begin{array}{l} \text{VOID (RATIO)} \\ \text{VOLUME (COMPRESSIBILITY)} \\ \text{COMPRESSIBILITY} \\ \text{COMPRESSION INDEX} \end{array} \right\}$

44. Declaración SEMI INFINITE CONSOLIDATION PROFILE

En solución e impresión para esta declaración, se tiene

$\left\{ \begin{array}{l} \text{SOLVE} \\ \text{SOLVE/PRINT} \end{array} \right\}$  CONSOLIDATION  $\left\{ \begin{array}{l} \text{INFINITE} \\ \text{FINITE} \end{array} \right\}$  (AT) descripción

en donde si se usa la alternativa FINITE se continúa con

CALCULATE (FROM) [Z] v [TO] v

45. Declaración PRINT CONSOLIDATION SETTLEMENT

Se tiene la expresión

PRINT CONSOLIDATION (SETTLEMENT) (AT) descripción

46. Declaración TOTAL SETTLEMENT

Esta declaración tiene la forma

TOTAL (SETTLEMENT) (AT) descripción

pudiendo anteponer a la declaración las alternativas de SOLVE, PRINT o SOLVE/PRINT.

IV. Función RATE

Con esta función se proporciona la posibilidad de calcular el progreso de asentamiento y el progreso de cambios en presiones para un suelo de varias capas.

La declaración de inicialización es

RATE (descripción superficie)

en donde en la descripción de la superficie se proporcionan las coordenadas X, Y o R.

Las declaraciones 47 a 54 se aplican a esta función.

47. Declaración LAYERED CONSOLIDATION PROFILE

Se expresa con

LAYERED CONSOLIDATION PROFILE n

que define el número de capas compresibles del perfil del suelo.

48. Declaración LAYER NUMBER

Tiene la forma

LAYER NUMBER n<sub>1</sub>, ..., n<sub>m</sub>

e indica las capas de suelo que se consideran en el análisis de la función.

49. Declaración LAYER VALUE

Esta declaración se utiliza para definir las propiedades del suelo y se escribe

propiedad = [LAYER] n [VALUE] v

en donde

propiedad = {  
COEFFICIENT (OF CONSOLIDATION)  
LAYER THICKNESS  
PERMEABILITY  
VOLUME (COMPRESSIBILITY)

50. Declaración DRAINAGE

Las alternativas para esta declaración son

{  
DOUBLE  
SINGLE } (DRAINAGE)

51. Declaración PORE PRESSURE

En la determinación de presiones se tienen las alternativas siguientes:

CONSTANT { (INITIAL EXCESS PORE PRESSURE)  
 (SO) }

INITIAL EXCESS (PORE PRESSURE) { [Z] n [SO] v  
LAYER n [SO] v }

LAYERED INITIAL EXCESS (PORE PRESSURE) [LAYER] n [SO] v

52. Declaración LOAD HISTORY

Tiene la forma

LOAD HISTORY

que será seguida por alguna de las declaraciones

$$\underline{\text{TIME}} \left\{ \begin{array}{l} \underline{\text{RAMP}} \left[ \underline{\text{START}} \right] \vee \left[ \underline{\text{END}} \right] \vee \left[ \underline{\text{P}} \right] \vee \\ \underline{\text{CONSTANT}} \left[ \underline{\text{START}} \right] \vee \left[ \underline{\text{P}} \right] \vee \end{array} \right\}$$

53. Declaraciones SOLVE, PRINT

Tienen la forma

SOLVE

PRINT

SOLVE/PRINT

y se aplican a los elementos

EXCESS (PORE PRESSURE)

PORE (PRESSURE)

DEGREE (OF CONSOLIDATION)

AVERAGE (EXCESS PORE PRESSURE)

TIME  $\left\{ \begin{array}{l} \underline{\text{CONSOLIDATION}} \\ \underline{\text{TOTAL}} \end{array} \right\}$  (SETTLEMENT)

AVERAGE (EXCESS PORE PRESSURE) LAYER  $n_1, \dots, n_m$

RATE (COMPONENTS)

ALL (CONSOLIDATION COMPONENTS)

54. Declaración INTERMEDIATE PRINTOUT

Se tienen las alternativas

SUPPRESS INTERMEDIATE PRINTOUT

INTERMEDIATE PRINTOUT

para el control de impresión de resultados intermedios.

*notas sobre el sistema  
integrado de ingeniería civil*

---

**subsistema  
ICES-TRANSET**

---

*Dr. Rodolfo Luthe G.*

---

El sistema TRANSET se utiliza en la solución de redes de transporte, representando carreteras o transportación masiva, aun cuando puede aplicarse a otras redes definidas por nodos, unidos por conexiones y con determinada separación entre ellos

Las declaraciones de este sistema se dividen en los grupos siguientes:

- I. Declaraciones para la generación de archivos.
- II. Declaraciones de trayectoria mínima.
- III. Declaraciones de asignación de tráfico.
- IV. Declaraciones de modificación y manejo de archivos.

1. Las declaraciones para la generación de archivos se pueden usar para crear archivos de información de redes temporales y permanentes, matriz de viajes, funciones de volumen-velocidad y velocidad de generación. Las declaraciones de este grupo son de la 1 a la 15;

Las redes de transporte pueden generarse como archivos temporales utilizando las declaraciones:

- Declaración SCALE FACTOR
- Declaración READ NETWORK
- Subdeclaraciones LINK
- Declaración EDIT NETWORK

que tienen la forma que se indica a continuación

1. Declaración SCALE FACTOR

Tiene la forma

USE SCALE (FACTOR) , x

en donde x es un factor de escala que traduce las distancias entre nodos y tiempos a un entero 0 y 4095. Cuando no se especifica se supone igual a 25, que corresponde a distancias entre 0 y 163.8 millas, y los mismos números en minutos para tiempos.

2. Declaración READ NETWORK

Esta declaración tiene la forma

READ NETWORK 'identif<sub>1</sub>', tipo de formato, (NODES) n<sub>1</sub>, (LINGS) n<sub>2</sub>, (ZONES) n<sub>3</sub>, (VOLUME DELAY SET) 'identif<sub>2</sub>', (ITEMS PER CARD) n<sub>4</sub>, 'formato'

en donde

'identif<sub>1</sub>' es el nombre alfanumérico de la red y consta de uno a ocho caracteres.

tipo de formato =  $\left. \begin{array}{c} \rightarrow \text{FREE} \\ \text{FIXED} \end{array} \right\} \text{FORMAT}$

n<sub>1</sub> es el número de nodos o intersecciones de la red. Si no se especifica, se obtiene de la información en las subdeclaraciones LINK.

$n_2$  es el número de conexiones o segmentos de un sentido entre intersecciones de la red. Se debe especificar si se usa - - FIXED FORMAT pero puede no especificarse como en el caso previo.

$n_3$  es el número de zonas o nodos de la red de donde se originan viajes.

'identif<sub>2</sub>' es el nombre del conjunto de volumen de vehículos-velocidad promedio, que se usa en la red.

$n_4$  es el número máximo de conexiones de cada subdeclaración LINK de formato fijo que siga. Se iguala a cuatro cuando no se especifica.

'formato' es la especificación FORTRAN del formato (máximo 78 caracteres) por parte del usuario, para controlar la lectura de las subdeclaraciones LINK de formato fijo.

### 3. Subdeclaración LINK

Tiene la forma

LINK  $n_1$ , descrip. conexión, descrip. conexión, ...

en donde  $n_1$  es un número secuencial de tarjeta proporcionado como ayuda de control y

descripción de la conexión = (FROM)  $n_1$  (TO)  $n_2$  (DISTANCE)  $x_3$   
(LANES)  $n_4$  (VOLUME/DELAY)  $n_5$

que conecta al nudo inicial  $n_1$  con el nudo de destino  $n_2$  (menores que el número 32,768) siendo  $n_4$  el número de carriles (máximo 7) y  $n_5$  el número de funciones volumen-velocidad (máximo 14).

En formato libre pueden tenerse para describir una red tantas subdeclaraciones LINK como sean necesarias, a continuación de una declaración de lectura de una red.

En formato fijo puede usarse el formato estándar de TRANSET o uno especificado por el usuario, requiriéndose en los dos casos para cada tarjeta el identificador de la tarjeta de conexión, su número secuencial y los elementos descriptivos. El formato estándar TRANSET para cuatro uniones por tarjeta es:

(A4, 13, 4 (2 13, F5.2, 212, 1X))

no debiendo de excederse este número de uniones por tarjeta.

### 4. Declaración EDIT NETWORK

Se expresa con

EDIT NETWORK

y se puede usar para indicar que se han leído las tarjetas de datos de la



5.

red y que puede iniciarse la verificación de datos, cuando la cantidad de datos de la red no se ha especificado exactamente en la declaración de lectura.

La declaración debe estar después de la última declaración 3 cuando se usa formato libre sin especificar el número de conexiones o cuando este número es mayor que el número real de conexiones de la red.

La matriz de los viajes reales o potenciales entre todos los pares de zona se puede generar como un archivo temporal utilizando las declaraciones:

Declaración READ TRIP MATRIX

Subdeclaraciones TRIP

Declaración EDIT TRIP MATRIX

que tienen la siguiente expresión.

5. Declaración READ TRIP MATRIX

Tiene la presentación

READ TRIP MATRIX 'identif<sub>1</sub>', tipo de formato, (ZONES) n<sub>1</sub>, (GENERATION RATE SET) 'identif<sub>2</sub>' (ITEMS PER CARD) n<sub>2</sub>, 'formato'

en donde

'identif<sub>1</sub>' es el nombre de la matriz de viaje

n<sub>1</sub> es el número de zonas en la matriz de viaje

6.

'identif<sub>2</sub>' es el nombre del conjunto de velocidad de generación.

n<sub>2</sub> es el número máximo de transferencias entre zonas en cada subdeclaración TRIP, necesaria sólo para formato fijo.

tipo de formato y 'formato' son como se definió en la declaración 2, pero aplicados a la subdeclaración TRIP.

6. Subdeclaración TRIP

Tiene la forma

TRIP n<sub>1</sub>, descrip. viaje, descrip. viaje, . . .

en donde

n<sub>1</sub> es el número secuencial de tarjeta proporcionado como ayuda de control y

descripción de viaje : (FROM) n<sub>1</sub> (TO) n<sub>2</sub> (VOLUME) n<sub>3</sub>, (GENERATION/RATE) n<sub>4</sub>

que describe el par de zona del nudo n<sub>1</sub> al nudo de destino n<sub>2</sub>. El volumen n<sub>3</sub> debe representar el mismo periodo de tiempo que la unidad de tiempo de volumen de conexión usado para definir las funciones de volumen-velocidad, siendo 32,766 el volumen máximo. El número n<sub>4</sub> de función de velocidad de generación se debe especificar sólo cuando se --

usan velocidades de generación en los viajes, y su valor máximo es 4.

En el formato libre pueden usarse tantas subdeclaraciones TRIP como sean necesarias para describir una matriz de viaje, después de una declaración de lectura de una matriz de viaje.

En el formato fijo puede utilizarse el formato estándar de TRANSET ó uno especificado por el usuario, necesitándose en ambos casos para todas las tarjetas de pares de zona. El formato estándar TRANSET cuando se usan las curvas de velocidad de generación, proporcionando cinco pares de zona por tarjeta es:

(A4, 1X, 13, 5 (213, 14, 12, 1X))

que cuando no se usan estas curvas, el formato para seis pares de zona por tarjeta, es

(A4, 1X, 13, 6 (213, 14, 1X))

no debiendo de excederse este número de pares de zona por tarjeta.

#### 7. Declaración EDIT TRIP MATRIX

Esta declaración tiene la forma

EDIT TRIP MATRIX

y se puede usar para indicar que todas las subdeclaraciones de viaje se

7.

han leído y que puede iniciarse la verificación de datos, cuando no se han especificado algunos intercambios de zona debido a que son nulos. Esta declaración debe ir a continuación de la última subdeclaración de viaje.

Se puede generar como archivo temporal un conjunto de funciones que relacione los tiempos de viaje de una conexión con su volumen, para varios tipos de carreteras, usando las declaraciones:

Declaración READ VOLUME DELAY

Subdeclaraciones VOLUME DELAY

Declaración EDIT VOLUME DELAY

y que se presentan a continuación

#### 8. Declaración READ VOLUME DELAY

Tiene la forma

READ VOL/DELAY SET 'identif', (FUNCTIONS)  $n_1$  (POINTS/FUNCTION)  $n_2$

en donde

'identif' es el nombre del conjunto volumen-velocidad

$n_1$  es el número de funciones volumen-velocidad y su máximo número es 14.

$n_2$  es el número de puntos usados para definir los segmentos lineales de cada función y puede utilizarse cualquier número

8.

ro de puntos.

Los datos volumen-velocidad se leen siempre en formato libre.

9. Subdeclaración VOLUME DELAY FUNCTION

Se expresa con

V/D FUNCTION  $n_1$ , TYPE tipo de conexión, CAPACITY POINT  $n_2$ ,  
especif. de punto, especif. de punto, . . .

Se define la función volumen-velocidad  $n_1$ , pudiendo seleccionarse el tipo de conexión entre:

EXPRESSWAY

ARTERIAL

LOCAL

El punto de capacidad,  $n_2$ , es un número de punto, que indica que el volumen del punto especificado es la capacidad práctica de las facilidades representadas por esta función. Cada especificación de punto incluye como valor X al volumen por carril por unidad de tiempo y como valor Y los minutos por milla (inverso de la velocidad) para el volumen especificado. Se supone que la unidad de tiempo para el valor X es de una hora, pero puede ser diferente.

El número de funciones debe ser igual al número especificado en la declaración de lectura de volumen-velocidad.

9.

10.

10. Declaración EDIT VOLUME DELAY

Tiene la forma

EDIT VOLUME DELAY SET

y señala que todas las funciones volumen-velocidad se han leído y que -- puede empezar la impresión de las funciones.

Se puede generar un conjunto de funciones que relacionan tiempo que viaja entre zonas y la fracción de viajes potenciales para varios tipos de viajes entre zonas, utilizando las declaraciones:

Declaración. READ GENERATION

Subdeclaración GENERATION RATE

Declaración EDIT GENERATION RATE

Las declaraciones anteriores son:

11. Declaración READ GENERATION

Se expresa con

READ GEN/RATE SET 'identif', (FUNCTIONS)  $n_1$ , (POINTS/FUNCTION)  $n_2$

en donde

'identif' es el nombre del conjunto de la velocidad de generación

$n_1$  es el número de las funciones de velocidad de generación del conjunto, y son cuatro como máximo.

$n_2$  es el número de puntos utilizados para definir los segmentos lineales de cada curva y puede usarse cualquier número de puntos.

Los datos se leen siempre con formato libre.

### 12. Declaración GENERATION RATE

Esta declaración tiene la forma

G/R (FUNCTION)  $n_1$ , especific. de punto, especific. de punto, . . .

y define la función  $n_1$  de velocidad de generación. Cada especificación de punto incluye como valor X los minutos por milla y como valor Y la fracción del volumen asignado al volumen potencial a la inversa de la velocidad especificada. Para el primer punto el valor de X es cero y el valor de Y es igual a uno.

Los elementos de datos se especifican en formato libre y el número de funciones debe ser igual al número especificado en la declaración de lectura de la velocidad de generación.

### 13. Declaración EDIT GENERATION RATE

Esta declaración se expresa con

EDIT GENERATION RATE SET

y se debe presentar después de la lectura de todas las funciones de velocidad de generación para indicar que la impresión puede iniciarse.

Para la generación y recuperación de archivos permanentes se usan las declaraciones 14 y 15.

### 4. Declaración STORE

Tiene la forma

STORE tipo de archivo 'identif'

y añade a los archivos permanentes de memoria secundaria un archivo de información temporal generado con declaraciones de lectura o cualquier archivo que exista en la memoria de la máquina.

tipo de archivo = {  
NET WORK  
TRIP MATRIX  
VOLUME DELAY SET  
GENERATION RATE SET

'identif' es el nuevo nombre del archivo de información temporal, el cual también cambia de nombre. Si no se especifica se conserva el nombre anterior.

### 15. Declaración LOAD

Se expresa con

LOAD tipo de archivo 'identif'

y permite recuperar cualquier archivo que forme parte de un archivo per

manente. El tipo de archivo es como se especificó en la declaración anterior.

'identif' es el nombre del archivo que se va a recuperar y que debe de almacenarse previamente como archivo permanente.

II. Las trayectorias mínimas se calculan como parte integral del procedimiento de asignación de tráfico y se deben especificar los datos por usarse, así como la cantidad y tipo de resultados que se desean. Esto se realiza con las declaraciones de generación y recuperación de archivos y las tres declaraciones de trayectorias mínimas. El orden normal de las declaraciones es

Declaraciones de generación o recuperación de archivos.

Declaración TIME VALUE

Declaración OUTPUT

Declaración SELECT MINIMUM

La selección de trayectorias mínimas implica que la información de la red y del volumen-velocidad se encuentre disponible en la memoria de la computadora. La información se puede generar con las declaraciones 1 a 4, o recuperar de un archivo permanente con la declaración 15. Si el archivo volumen-velocidad no existe como archivo permanente, se debe generar con las declaraciones 8 a 10.

16. Declaración TIME VALUE

Se presenta en la forma

USE TIME VALUE x

y para cada conexión determina los costos de tiempo, accidentes y operación del vehículo al seleccionar trayectorias de costo mínimo, que dependen del valor de tiempo x especificado. Cuando no se especifica, se usa un valor de 2 dólares por vehículo hora.

17. Declaración OUTPUT

Se expresa con

OUTPUT tipo de salida OF MINIMUM datos

y controla la salida de todas las declaraciones que le siguen hasta que aparezca una nueva declaración de salida, en donde

tipo de salida =	{	PRINTOUT (S)	}
		CARD DECK (S)	
		DISK FILE (NAMED) 'identif'	
		SEPARATION	
datos =	{	TRACES	}
		SEPARATION AND TRACES	

Las separaciones mínimas entre zonas son los tiempos, distancias o costos de viaje del punto de origen al destino. Las trazas son una lista ordenada de los nodos por los que se pasa del punto de origen al de destino.

18. Declaración SELECT MINIMUM

Tiene la forma

SELECT MINIMUM { TIME  
DISTANCE  
COST } { PATHS  
TREES } (BY PASSING

El orden normal de las declaraciones usadas para realizar una asignación completa, es

ZONE CENTROIDS), lista de nodos en donde se proporciona datos de trayectoria mínima para pares específicos de origen y destino con PATHS y datos de árbol mínimos para todas las trayectorias de los orígenes especificados a todas las destinaciones - si se usa TREES.

Declaraciones de generación o recuperación de archivos.

Declaraciones de especificación de control

Declaración de salida intermedia general

Declaraciones de salida intermedia

Declaración de ejecución de asignación

Declaraciones de salida final

La lista de nodos es la siguiente:

Cuando se solicitan datos con PATH, se escribe

Cuando necesita reiniciarse una asignación cuya ejecución se interrumpe, se usan las declaraciones

( ORIGIN { ZONE  
NODE } ) n<sub>1</sub> ( DESTINATION { ZONE  
NODE } ) n<sub>2</sub> . . .

Declaración de reinicio de ejecución

Fajo de tarjetas de reinicio

Declaraciones de salida final

cuando se solicitan datos con TREE, se escribe

{ ( ORIGIN { ZONES  
NODES } n<sub>1</sub> n<sub>2</sub> . . . )  
ALL ZONES }

en donde la segunda declaración no se requiere cuando se ha formado un archivo de reinicio.

III. Las declaraciones de asignación de tráfico implica que los archivos sobre la red, matriz de viaje y volumen-velocidad se encuentren en la memoria de la máquina cuando se ejecuten las declaraciones de asignación de tráfico. La generación de estos archivos se obtiene con las declaraciones 1 a 10 y si existen pueden recuperarse con la declaración 15. La creación de archivos de velocidades de generación se logra con las declaraciones 11 a 13.

Las declaraciones de especificación de control son opcionales y algunas de ellas sólo pueden usarse cuando se seleccionan modos de asignación particular.

19. Declaración RANDOM NUMBER

Se expresa con

USE RANDOM NUMBER n<sub>1</sub>, ( MULTIPLIER ) n<sub>2</sub>

y permite variar la sucesión aleatoria de zonas de origen o pares de zona origen-destino en los modos de asignación incremental: árboles, trayectorias, árboles y trayectorias, en donde

$n_1$  es el número aleatorio inicial, de seis dígitos y no divisible entre 2 ó 5.

$n_2$  es el multiplicador de valor 1003 ó 1011, suponiéndose 1011 si no se especifica.

#### 20. Declaración INCREMENT

Es de la forma

$$\text{USE INCREMENT} \left\{ \begin{array}{l} x \text{ PERCENT} \\ n \text{ TRIPS} \end{array} \right\}$$

y establece el incremento de viajes, en donde

$x$  es el porcentaje máximo de viajes entre zonas que debe asignarse cada vez que se selecciona un par de zona, variando entre 1 y 100.

$n$  es el número máximo de viajes que deben asignarse cada vez que se selecciona un par de zona y debe ser mayor que 1

#### Declaración ADJUSTMENT

Se presenta como

$$\text{USE ADJUSTMENT (FACTOR)} x_1, (\text{CUTOFF VALUE}) x_2$$

y controla el ajuste de tiempos de viaje de conexiones, en donde

$x_1$  es una fracción entre 0 y 1, que se multiplica primero por la diferencia entre el tiempo de viaje anterior y el tiempo con el nuevo volumen, añadiéndose el resultado al tiempo de viaje previo, y suponiendo un valor de 0.25 cuando no se especifica este factor.

$x_2$  es la diferencia de velocidades en valor absoluto y en millas por hora, que debe existir antes del ajuste del tiempo de conexión. El valor no debe ser negativo y se supone cero cuando no se especifica.

#### Declaración MAXIMUM TIME

Se escribe con

$$\text{MAXIMUM RUNNING TIME } n \left\{ \begin{array}{l} \rightarrow \text{MINUTES} \\ \text{SECONDS} \end{array} \right\}$$

y controla la duración de la ejecución de la asignación, en cualquier modo, excepto el modo libre. El tiempo máximo se indica con  $n$ , y la ejecución se para si la asignación no se ha completado en este tiempo, que es de 15 minutos cuando no se especifica su valor.

## 23. Declaración MAXIMUM ASSIGNMENTS

Se utiliza para especificar el número máximo de asignaciones completas, en el modo de asignación total o nulo usando árboles de trayectoria mínima con restricción al final de una asignación completa (BPR) y es de la forma

MAXIMUM NUMBER OF ASSIGNMENTS n

en donde cada asignación incluye la determinación de todas las trayectorias mínimas, la adición de todos los viajes a estas trayectorias y el ajuste posterior de todos los tiempos de viaje de las conexiones. Cuando no se especifica se supone un valor de n igual a 5.

## 42. Declaración MAXIMUM ITERATIONS

Se usa cuando se selecciona un modo de asignación incremental y se expresa con

MAXIMUM NUMBER OF ITERATIONS n

en donde n es el número máximo de iteraciones antes de la terminación de la asignación, definiéndose una iteración como la asignación de un incremento sencillo (posiblemente 100%) de los viajes de un origen a un destino.

## 25. Declaración SAVE DATA

Tiene la forma

SAVE RESTART DATA ON

{ -DISK  
CARD }

y permite guardar información si la ejecución de la asignación se detiene, porque se exceda el tiempo o cantidad de computación máximos. Esta información puede permitir la continuación de la ejecución posteriormente, mediante la declaración 33.

## 26. Declaración SAVE RESULTS

Se expresa con

SAVE ASSIGNMENT RESULTS IN (NETWORK FILE) 'identif'

y guarda los resultados de una asignación como un archivo permanente de nombre 'identif'. El nombre del archivo de la red puede ser el nombre de entrada de la red o un nombre nuevo.

Las declaraciones de salida intermedia pueden solicitarse con una declaración de salida intermedia general seguida de cuatro subdeclaraciones de salida intermedias, las que como son opcionales, no se necesitan si no se requieren salidas intermedias.

## 27. Declaración INTERMEDIATE ASSIGNMENT

Esta declaración se escribe como

INTERMEDIATE ASSIGNMENT tipo de salida, (EVERY) n ITERATIONS

en donde tipo de salida se indicón en la declaración 17 y n es el número de iteraciones de asignación incrementales (la asignación de todos o una porción de los viajes entre un origen y destino dados) entre salidas intermedias.



Cuando se procesa una asignación incremental, la información de la frecuencia es opcional y si no se especifican  $n$  se iguala a 1000.

Esta declaración sólo se escribe una vez para cualquiera de los cuatro tipos de salida intermedia que se deseen.

### 28. Declaración INTERMEDIATE LINK

Esta subdeclaración tiene la forma

REQUEST INTERMEDIATE LINK  $\left\{ \begin{array}{l} * \text{ VOLUMES} \\ \text{SPEEDS} \\ \text{TIMES} \end{array} \right\}$  FOR grupo de conexiones

en donde

grupo de conexiones =  $\left\{ \begin{array}{l} * \text{ EXPRESSWAYS} \\ \text{ARTERIALS} \\ \text{LOCAL STREETS} \\ \text{ALL LINKS} \\ \text{LINKS, lista de conexiones} \end{array} \right\}$

siendo la lista de conexiones de la forma

(FROM)  $n_1$  (TO)  $n_2$ , etc.

que conecta al nudo de origen  $n_1$  con el de destino  $n_2$ .

La subdeclaración proporciona tablas de cualquier combinación solicitada de volúmenes de conexiones, velocidades y tiempos de viaje tan fre-

cientemente como sea necesario en la declaración general. Esta información puede solicitarse para todas las conexiones selectas o todas las conexiones de vía rápida, arterial o local.

### 29. Declaración INTERMEDIATE TRIP

Esta subdeclaración tiene la expresión

REQUEST INTERMEDIATE TRIP STATUS FOR  $\left\{ \begin{array}{l} \text{ORIGIN ZONES, } n_1, n_2, \dots \\ \text{ALL ZONES} \end{array} \right\}$

Esta subdeclaración proporciona una matriz de datos del estado de viaje con la frecuencia que se requiera en la declaración general. La información puede solicitarse para todos los pares de zona o de zonas de origen seleccionadas hacia todas las zonas de destino.

### 30. Declaración INTERMEDIATE PATHS

Esta subdeclaración tiene la forma

REQUEST INTERMEDIATE MINIMUM PATHS FOR pares de nodos

en donde pares de nodos se indica con

(ORIGIN NODE)  $n_1$  (DESTINATION NODE)  $n_2$ , etc.

Esta subdeclaración proporciona datos de trayectorias mínimas tan frecuentemente como la declaración general lo solicite. Se obtienen listas de los nodos en las trayectorias mínimas usuales entre pares de nodos

seleccionados en la red, así como los tiempos de viaje en estas trayectorias.

### 31. Declaración INTERMEDIATE LINK USAGE

Esta subdeclaración se presenta en la forma

REQUEST INTERMEDIATE LINK USAGE FOR LINKS, lista de conexiones

en donde lista de conexiones se describió al final de la declaración 28.

Con esta subdeclaración se obtienen datos de usos de conexiones que se coleccionan continuamente durante la ejecución de una asignación y al final de cada asignación completa. Los datos de uso de conexiones incluyen el origen y destino de cada grupo de viajes utilizando conexiones especificadas.

Las declaraciones de ejecución son las declaraciones 32 y 33.

### 32. Declaración ASSIGN

Esta declaración tiene la forma

ASSIGN BY tipo MODE

en donde

tipo = {  
FREE  
BPR  
BPR USING BPR CAPACITY RESTRAINT  
TREE  
PATH  
TREE AND PATH

La declaración indica que la ejecución de la asignación puede empezar en el modo especificado y puede escribirse cuando se han especificado todos los datos y los parámetros de control estándar y cuando se han solicitado todas las salidas intermedias. En cada problema sólo se puede considerar una declaración de ejecución de asignación.

### 33. Declaración RESTART ASSIGNMENT

Tiene la expresión

$$\text{RESTART ASSIGNMENT ON } \left\{ \begin{array}{l} \text{CARDS} \\ \text{DISK (FILE) 'identif'} \end{array} \right\} \text{ MAXIMUM RUNNING} \\ \text{TIME } n_1 \left\{ \begin{array}{l} \text{MINUTES} \\ \text{SECONDS} \end{array} \right\} (\text{MAXIMUM (NUMBER OF } \left\{ \begin{array}{l} \text{ITERATIONS} \\ \text{ASSIGNMENTS} \end{array} \right\} n_2), \\ \text{(frecuencia)}$$

en donde

frecuencia = INTERMEDIATE OUTPUTS (EVERY)  $n_3$  ITERATIONS que se aplica sólo si se continúa una asignación incremental.

CARDS Se usa cuando se tiene un fajo de tarjetas de reinicio

'identif' es el nombre generado por el subsistema de un archivo permanente de reinicio

$n_1$  es el tiempo de corrida adicional permitido

$n_2$  es el número adicional de iteraciones o asignaciones

$n_3$  es el nuevo número de iteraciones entre salidas intermedias.

La declaración se puede usar cuando la asignación se detuvo por alcanzar alguno de sus límites (cálculo o tiempo máximo) antes de completarse y cuando se incluía la declaración 25 de reinicio de datos antes de la declaración de ejecución.

Las declaraciones de salida final deben usarse cuando se ha asignado completamente la red de referencia, proporcionando salidas algunas de las declaraciones únicamente en base de los datos de la red. Estas declaraciones se presentan de la declaración 34 a la 40.

#### 34. Declaración FINAL LINK TABLE

Se presenta en la forma

REQUEST FINAL tipo de salida OF LINK \*  $\left. \begin{array}{l} \text{VOLUMES} \\ \text{SPEEDS} \\ \text{TIMES} \end{array} \right\}$  red fuente FOR grupo de conexiones

en donde tipo de salida se indica en la declaración 17.

red fuente = FROM (NETWORK) 'identif'

especificándose con 'identif' el nombre de la red, incluyéndose un solo nombre.

Grupo de conexiones es como se definió en la declaración 28.

Con esta declaración se obtiene una tabla de cualquier información combinada que se solicite.

#### 35. Declaración FINAL MINIMUM PATH

Se expresa con

REQUEST FINAL tipo de salida OF MINIMUM PATHS, red fuente FOR (ORIGIN NODE)  $n_1$  (DESTINATION NODE)  $n_2$ , etc.

en donde tipo de salida se describió en la declaración 17 y red fuente se indicó en la declaración previa.

Con esta declaración se proporcionan datos de trayectoria mínima para la red asignada que se ha especificado. Se obtienen también listas de nodos y de tiempos de viaje en las trayectorias mínimas entre los pares de nodos seleccionados.

#### 36. Declaración FINAL TRAVEL TIME

Se escribe como

REQUEST FINAL tipo de salida OF TRAVEL TIMES, red fuente,

FOR  $\left. \begin{array}{l} \text{ORIGIN NODES } n_1, n_2, \dots \\ \text{ALL ZONES} \end{array} \right\}$

en donde tipo de salida y red fuente se describieron en las declaraciones 17 y 34, respectivamente.

Esta declaración proporciona una matriz de tiempos finales de viaje entre zonas para la red especificada. Estos tiempos se pueden obtener para todos los pares de zona o de nodos de origen seleccionados hacia todas las zonas de destino.

### 37. Declaración FINAL SYSTEM TRAVEL

Esta declaración tiene la expresión

REQUEST FINAL tipo de salida OF SYSTEM TRAVEL, red fuente en donde tipo de salida y red fuente se describieron en las declaraciones 17 y 34 respectivamente.

Esta declaración presenta un resumen de vehículo-horas y vehículo-millas de viaje en la red especificada. Además se proporcionan los totales de la red completa y los subtotales en vías rápidas, arteriales y calles locales.

### 38. Declaración FINAL TRIP DISTRIBUTION

Tiene la forma

REQUEST FINAL tipo de salida OF  $\left. \begin{array}{l} \text{TIME} \\ \text{DISTANCE} \end{array} \right\}$  DISTRIBUTION, red fuente,

FOR (GROUP SIZE) x  $\left. \begin{array}{l} \text{MINUTES} \\ \text{MILES} \end{array} \right\}$

en donde tipo de salida y red fuente se describieron en las declaraciones 17 y 34, respectivamente, y

x es el tamaño de los grupos, en minutos para distribuciones de tiempo y en millas para distribuciones de distancia, en los cuales se separan los viajes para la salida. Cuando no se especifica el tamaño del grupo, se usan valores de 5 minutos o 1 milla.

Esta declaración proporciona información en la distribución de tiempos de viaje o distancias para la red especificada. Se indican también el número de viajes y el porcentaje de viajes totales cuyo tiempo o distancia se encuentra dentro de cada número de grupos. El usuario puede seleccionar el tamaño de los grupos en minutos o millas. Los datos de distribución se resumen dando el promedio y la desviación estándar del tiempo de viaje o de la distancia.

### 39. Declaración FINAL INTERZONAL TRIP

Se escribe en la forma

REQUEST FINAL  $\left. \begin{array}{l} \text{PRINTOUT} \\ \text{CARD DECK} \end{array} \right\}$  OR INTERZONAL TRIPS

FOR {  
ORIGIN ZONES  $n_1, n_2, \dots$   
ALL ZONES }

y con esta declaración se proporcionan los viajes realmente asignados, que serán diferentes de la matriz de viaje de entrada sólo si durante la asignación se usaron funciones de velocidad de generación. Los viajes se pueden obtener para todos los pares de zona o para todas las zonas de destino a partir de zonas de origen seleccionadas. Para esta declaración no se hace referencia a una red y por tanto no necesita especificarse la red fuente.

#### 40. Declaración FINAL DISK INTERZONAL TRIP

Esta declaración tiene la forma

REQUEST FINAL DISK FILE (NAMED) 'identif' OF INTERZONAL  
TRIPS FOR ALL ZONES

en donde 'identif' es el nombre que se da al archivo de matriz de viaje que contiene los viajes realmente asignados.

Con esta declaración se guardan los viajes realmente asignados al usar se funciones de velocidad de generación, los que forman un nuevo archivo permanente, e igual que la declaración anterior, no implica el hacer referencia a una red y no es necesario especificar la red fuente.

IV. Las declaraciones de modificación y manejo de archivos se consideran en esta sección y las redes permanentes o temporales pueden modificar

se utilizando la sucesión de declaraciones

Declaración MODIFY NETWORK

Subdeclaraciones ADD LINK

Subdeclaraciones DELETE LINK

Subdeclaraciones CHANGE LINK

Subdeclaraciones ADD ZONE

Subdeclaraciones DELETE ZONE

Subdeclaraciones CHANGE NODE

Declaración EDIT NETWORK

El orden de las subdeclaraciones puede alterarse, presentándose las declaraciones anteriores a continuación.

#### 41 Declaración MODIFY NETWORK

Esta declaración tiene la forma

MODIFY NETWORK 'identif<sub>1</sub>' (FORMING 'identif<sub>2</sub>' )

en donde

'identif<sub>1</sub>' es el nombre de una red disponible en la memoria de la computadora o como archivo permanente

'identif<sub>2</sub>' es el nombre de la red modificada

La declaración especifica el nombre de una red existente que se va a cambiar por las subdeclaraciones que le siguen. Si la red fuente existe

en un archivo permanente, este archivo permanente no cambia aun si no se especifica un nombre nuevo.

## 42. Declaración ADD LINK

Esta subdeclaración se presenta como

ADD LINK, descrip. de conexión, descrip. de conexión, ...

en donde la descripción de una conexión es

(FROM)  $n_1$  (TO)  $n_2$  (DISTANCES x (LANES)  $n_4$  (VOLUME/DELAY)  $n_5$

siendo  $n_1$  el nudo de origen y  $n_2$  el de destino, con valor máximo de 32,768. La distancia x se expresa en millas y el máximo número de carriles  $n_4$  es 7, siendo 14 el máximo valor de las funciones  $n_5$  de volumen-velocidad.

La subdeclaración permite añadir a la red cualquier número de conexiones nuevas, existiendo en la red fuente los nodos de origen y destino, o siendo nodos nuevos.

## 43. Declaración DELETE LINK

Esta subdeclaración se expresa con

DELETE LINK, (FROM)  $n_1$  (TO)  $n_2$ , ...

en donde se especifica cualquier número de conexiones, cada una con su nudo de origen y de destino.

La subdeclaración permite borrar de la red cualquier número de conexiones existentes.

## 44 Declaración CHANGE LINK

La subdeclaración tiene la forma

CHANGE LINK, descripción de conexión, descripción de conexión, ...

en donde descripción de conexión es como se indica en la declaración 42.

Esta subdeclaración permite cambiar en una red fuente la distancia, la función volumen-velocidad y el número de carriles de un número cualquiera de conexiones.

## 45. Declaración ADD ZONE

Esta subdeclaración permite cambiar los nodos de una red fuente de centroides fuera de zona o centroides de zona y tiene la forma:

ADD ZONE,  $n_1$   $n_2$ , ...

en donde  $n_1$  y  $n_2$  son nodos existentes que no son centroides de zona.

Los nodos especificados se añaden en orden al final de la lista existente de los centroides de zona.

## 46. Declaración DELETE ZONE

Tiene la forma

DELETE ZONE,  $n_1$ ,  $n_2$ , ...

en donde  $n_1$  y  $n_2$  son centroides existentes en una zona.

Con esta subdeclaración se cambian los centroides de zona de una red fuente a nodos estándar, que no son centroides de zona.

#### 47. Declaración CHANGE NODE

Se presenta como

CHANGE NODE NUMBER, (FROM)  $n_1$  (TO)  $n_2$ , (FROM)  $n_3$  (TO)  $n_4$ , ...

en donde

$n_1$  y  $n_3$  son nodos existentes

$n_2$  y  $n_4$  son los números de los nodos nuevos.

Esta subdeclaración puede usarse para cambiar los números asignados a nodos existentes en la red original.

Las declaraciones para la modificación de matrices de viaje permanentes o temporales se realiza con la sucesión de declaraciones siguientes:

Declaración MODIFY TRIP MATRIX

Subdeclaraciones ADD ZONES

Subdeclaraciones DELETE ZONES

Subdeclaraciones CHANGE TRIPS

Subdeclaraciones CHANGE ZONE

Declaración EDIT TRIP MATRIX

que se presentan a continuación:

#### 48. Declaración MODIFY TRIP MATRIX

Tiene la forma

MODIFY TRIP MATRIX 'identif<sub>1</sub>' (FORMING 'identif<sub>2</sub>')

en donde

'identif<sub>1</sub>' es el nombre de una matriz de viaje disponible en la memoria de la computadora o como archivo permanente.

'identif<sub>2</sub>' es el nombre de la matriz de viaje modificada.

La declaración especifica el nombre de una matriz de viaje existente que se va a cambiar por las subdeclaraciones que la siguen. Si la matriz de viaje fuente existe en un archivo permanente, este archivo permanente no cambia aun si no se especifica un nombre nuevo.

#### 49. Declaración ADD ZONES

Se escribe como

ADD ZONES  $n_1$ ,  $n_2$ , ...

en donde  $n_1$  y  $n_2$  son los nuevos números de zonas.

Con esta subdeclaración se pueden añadir zonas nuevas a la matriz de viaje, las que se añaden al final de la lista existente de zonas.

## 50. Declaración DELETE ZONES

Se presenta como

DELETE ZONES  $n_1, n_2, \dots$

en donde  $n_1$  y  $n_2$  son los números de zona existentes.

Esta subdeclaración permite eliminar zonas existentes de la matriz de viaje y al borrar cada zona se borran también todos los viajes hacia o de cada zona.

## 51. Declaración CHANGE TRIPS

Tiene la forma

CHANGE TRIPS, (FROM)  $n_1$  (TO)  $n_2$  (NEW VOLUME)  $n_3$  (NEW GEN/RATE)  $n_4$ .

en donde se pueden cambiar los viajes  $n_3$  y opcionalmente el número  $n_4$  de velocidad de generación, entre la zona de origen  $n_1$  y la de destino  $n_2$ .

En esta subdeclaración se puede hacer un número cualquiera de cambios repitiendo todos los datos para cada cambio.

## 52. Declaración CHANGE ZONE

Se escribe en la forma

CHANGE ZONE NUMBER, (FROM)  $n_1$  (TO)  $n_2$  (FROM)  $n_3$  (TO)  $n_4, \dots$

en donde

$n_1$  y  $n_3$  son números de zona existentes.

$n_2$  y  $n_4$  son nuevos números de zona, menores de 32,768.

Con esta subdeclaración se pueden cambiar los números asignados a zonas existentes.

Las declaraciones para el manejo de archivos se han presentado antes pero ahora se agrupan por tipo de archivo, en lugar de por la capacidad o facilidad que proporcionan y son las siguientes:

## 53. Declaración PRINT DATA

Se expresa con

PRINT DATA FROM tipo de archivo DISK (FILE) 'identif<sub>1</sub>',  
(FILE) 'identif<sub>2</sub>', .

en donde

tipo de archivo =

NETWORK  
TRIP MATRIX  
VOLUME DELAY SET  
GENERATION RATE SET  
MINIMUM PATH OUTPUT  
INTERMEDIATE ASSIGNMENT OUTPUT  
FINAL ASSIGNMENT OUTPUT  
RESTART



Esta declaración no acepta RESTART en tipo de archivo y se imprimen los datos contenidos en los archivos permanentes 'identif<sub>1</sub>', 'identif<sub>2</sub>', - etc.

54. Declaración LIST NAMES

Tiene la forma

LIST NAMES FROM tipo de archivo DISK FILE

y produce el listado de los nombres de todos los archivos permanentes - de cualquier tipo de archivo.

55. Declaración ERASE FILE

Esta declaración se escribe

ERASE tipo de archivo DISK (FILE) 'identif<sub>1</sub>', (FILE) 'identif<sub>2</sub>', . . .  
ALL tipo de archivo DISK FILES

Con esta declaración se quitan de memoria secundaria los archivos permanentes especificados por sus nombres 'identif<sub>1</sub>', 'identif<sub>2</sub>', etc. o todos los archivos del tipo de archivo indicado, permitiendo dar otros usos a los espacios ocupados por dichos archivos.

*notas sobre el sistema  
integrado de ingeniería civil*

---

**subsistema**

---

**ICES-TABLE**

---

*Dr. Rodolfo Luthe G.*

---

ICES TABLE - I

Este subsistema permite formar y manipular datos tabulados, proporcionando una gran flexibilidad en la especificación de datos, de manera que el subsistema puede usarse en unión con otros subsistemas del ICES.

Un ejemplo puede ser el uso de las tablas de especificaciones de elementos en acero, ligadas al subsistema STRUDL. Las tablas son arreglos de dos dimensiones en donde se tiene un número arbitrario de renglones y de columnas, que se identifican, considerándose que los renglones representan elementos o entradas a la tabla y las columnas representan propiedades particulares. Los nombres no pueden exceder de 8 caracteres alfanuméricos. Los datos de entrada pueden especificarse en cualquier sistema de unidades y el subsistema TABLE convierte los datos a un solo sistema de unidades. Cuando no se especifican las unidades, se utilizan pulgadas, libras, radianes, segundos y grados Fahrenheit.

Lo anterior puede presentarse esquemáticamente como

	Prop. 1 (unidades)	Prop. 2 (unidades)	...	Prop. n (unidades)
Elem 1	v <sub>11</sub>	v <sub>12</sub>	...	v <sub>1n</sub>
Elem m	v <sub>m1</sub>	v <sub>m2</sub>	...	v <sub>mn</sub>

en donde un elemento puede ser acero, cobre, aluminio, etc. y una propiedad peso específico, módulo de elasticidad, etc.

Los encabezados de columnas o nombres de las propiedades y las unidades se proporcionan en una sola declaración, aunque pueden añadirse, modificarse o borrarse columnas en cualquier instante.

Se proporcionan dos niveles de conjuntos: el del subsistema y el del usuario. El nivel del subsistema protege a un conjunto de datos contra su uso no autorizado. Para formar una tabla o modificarla a este nivel se necesita la contraseña que permite tener acceso a esa tabla. Por el contrario, al nivel del usuario las tablas no necesitan contraseña y por lo mismo no tienen la misma protección.

Las declaraciones del subsistema TABLE son:

1. Inicialización del subsistema, con la declaración

TABLE (i) ('título')

en donde

i es una especificación opcional del tamaño de memoria que se desea tener disponible para los datos. Si no se indica, se supone el valor 15000, adecuado para una máquina de capacidad 128 K bytes. Se puede usar un valor mayor para computadoras de capacidad más grande, y la operación será más eficiente.

3.

'título' indica el título del problema, disponiéndose de un total de 64 caracteres alfanuméricos (incluyendo espacios en blanco). El título es opcional.

2. Contraseña para un conjunto de datos a nivel del subsistema. La contraseña es un identificador que puede tener un máximo de 8 caracteres alfanuméricos. Para cambiar una contraseña, primero se destruye y después se establece la nueva contraseña.

Para destruir una contraseña existente, se usa la declaración

DESTROY PASSWORD 'identif'

en donde 'identif' representa la contraseña existente.

Para establecer una contraseña nueva, se usa

CREATE PASSWORD 'identif'

en donde 'identif' representa la contraseña que se va a crear.

### 3. Declaración INITIATE TABLE

Esta declaración forma una nueva tabla y sólo puede utilizarse una vez para un nombre de la tabla dado. Tiene la forma

INITIATE TABLE 'nombre' ('contraseña')

en donde

'nombre' es el nombre de la tabla.

4.

'contraseña' es la contraseña al nivel protegido del subsistema. -  
Si no se indica, la tabla está al nivel del usuario.

### 4. Declaración MODIFY TABLE

Se modifica o actualiza una tabla existente, con

MODIFY TABLE 'nombre' ('contraseña')

en donde

'nombre' indica el nombre de la tabla existente, a nivel del subsistema o del usuario.

### 5. Declaración DELETE TABLE

Con esta declaración se borra una tabla existente. Tiene la forma

DELETE TABLE 'nombre' ('contraseña') (ALL)

en donde si aparece ALL, se borra toda la tabla identificada con 'nombre'. Si no aparece ALL, sólo se borra parte de la tabla y se esperan declaraciones más detalladas.

### 6. Declaración ORDER TABLE

Con esta declaración se ordenan los valores interiores de la tabla indicada. Tiene la forma

ORDER TABLE 'nombre' ('contraseña')

Después de esta declaración se esperan declaraciones de operación de la tabla.

## 7. Declaración TRANSFER TABLE

Se transmite información tabulada de un nivel a otro, con la declaración

$$\text{TRANSFER TABLE (FROM) 'nombre}_1 \left\{ \begin{array}{l} \text{SUBSYSTEM} \\ \text{USER} \end{array} \right\} \text{(TO) 'nombre}_2 \left\{ \begin{array}{l} \text{USER} \\ \text{SUBSYSTEM} \end{array} \right\} \text{'contraseña'} \text{(ALL)}$$

en donde

'contraseña' indica la contraseña del conjunto de datos del subsistema  
'nombre<sub>1</sub>' es el nombre de la tabla de donde se transfieren datos y debe existir en el conjunto especificado de datos.

'nombre<sub>2</sub>' es el nombre de la tabla a donde se transfieren datos y puede o no existir en el conjunto especificado de datos. Si no existe, 'nombre<sub>2</sub>' se inicia automáticamente y el número de propiedades en 'nombre<sub>2</sub>' es el mismo que en 'nombre<sub>1</sub>'. Si 'nombre<sub>2</sub>' existe, sólo las propiedades que existen en 'nombre<sub>2</sub>' se transfieren. Las propiedades que existen en 'nombre<sub>2</sub>' pero que no existen en 'nombre<sub>1</sub>' tendrán valor cero para los elementos transferidos y se necesita modificar 'nombre<sub>2</sub>' para completar la tabla. Para transferir una tabla al conjunto de datos del subsistema se requiere la contraseña correcta. Si no aparece ALL, sólo se transfiere parte de la tabla y se necesitan declaraciones adicionales de operaciones de las tablas.

## 8. Declaración OUTPUT TABLE

Con esta declaración se obtiene la impresión o perforación de tablas.

Tiene la forma

$$\text{OUTPUT} \left\{ \begin{array}{l} \text{PRINT} \\ \text{PUNCH} \end{array} \right\} \text{TABLE 'nombre' ('contraseña')} \left\{ \begin{array}{l} \text{FORMAT F} \\ \text{FORMAT E} \end{array} \right\} \text{(ALL)}$$

Cuando se especifica el formato F los datos tienen la forma decimal normal, con 6 dígitos a la derecha del punto decimal. Se añade la palabra ALL cuando se desea la tabla completa y si no se usa ALL se requieren declaraciones adicionales.

## 9. Declaración LIST FILE

Se lista el directorio de archivos de conjuntos de datos. Los archivos de un conjunto de datos pueden ser tablas que no necesariamente se crearon con TABLE. La declaración tiene la forma

$$\text{LIST (FILE) (DIRECTORY)} \left\{ \begin{array}{l} \text{SUBSYSTEM} \\ \text{USER} \end{array} \right\}$$

## 10. Declaración FILE TABLE

Es la última declaración de TABLE y tiene la forma

FILE TABLE

Sirve para terminar el bloque de operación de la última tabla y garantiza que se llevan a efecto todas las operaciones de archivo y actualización.

Las declaraciones 11 a 15 deben ir precedidas por alguna de las declaraciones INITIATE TABLE ó MODIFY TABLE.

11. Declaración ADD HEADINGS

Para añadir un nuevo encabezado de tabla a una tabla se usa la declaración

ADD HEADINGS

'nombre enc<sub>1</sub>'      dimensión (n) dimensión (n) . . .

'nombre enc<sub>m</sub>'      dimensión (n) dimensión (n) . . .

en donde

'nombre enc' es el nombre del nuevo encabezado y no debe exceder de 8 caracteres alfanuméricos, no permitiéndose en una sola tabla la duplicación de nombres de encabezados.

dimensión describe la dimensionalidad asociada con el encabezado añadido y puede ser

L ó LENGTH para longitud

W ó WEIGHT para peso o fuerza

A ó ANGLE para ángulo

TEMP para temperatura

TIME para tiempo

Si no se especifican dimensiones se supone que el en

cabecado añadido es adimensional.

n es un entero que indica la potencia de la dimensión precedente. Si no se especifica se considera igual a + 1. La dimensión se usa para convertir unidades de cualquier entrada y/o salida especificada a un sistema consistente de unidades.

12. Declaración ADD ITEMS

Se utiliza para añadir elementos nuevos a una tabla. Tiene la forma

ADD ITEMS ('nombre enc<sub>1</sub>' 'nombre enc<sub>2</sub>' . . .)

'nombre <sub>1</sub> '	v <sub>11</sub>	v <sub>12</sub>	v <sub>13</sub>	. . .
⋮				
'nombre <sub>m</sub> '	v <sub>m1</sub>	v <sub>m2</sub>	v <sub>m3</sub>	. . .

en donde

'nombre enc' es el nombre del encabezado de una columna de la tabla, como se explicó en la declaración anterior.

'nombre<sub>1</sub>' es el nombre del elemento o renglón nuevo que se añade, no excediendo 8 caracteres en longitud y no permitiéndose la duplicidad de nombres de elementos en una tabla.

v<sub>ij</sub> es el valor decimal de cada elemento que se añade.

13. Declaración **MODIFY ITEMS**

Esta declaración modifica los valores de un elemento existente en la tabla. Tiene la forma

```

MODIFY ITEMS ('nombre enc1' 'nombre enc2' . . . .)
      'nombre1'   v11   v12   v13   . . .
      ⋮
      'nombrem'   vm1   vm2   vm3   . . .
    
```

en donde

'nombre<sub>1</sub>' es el nombre del elemento o renglón que se va a modificar.

v<sub>ij</sub> es el nuevo valor decimal del elemento.

14. Declaración **ADD PROPERTIES**

Se utiliza para añadir valores numéricos faltantes en la tabla y tiene la forma

```

ADD PROPERTIES 'nombre enc1' 'nombre enc2' . . . .
('nombre1')   v11   v12   v13   . . .
⋮
('nombrem')   vm1   vm2   vm3   . . .
    
```

en donde los distintos términos se han definido en las declaraciones previas.

15. Declaración **UNITS**

Con esta declaración se especifican las unidades de entrada y/o salida. Tiene la forma

```

*
UNITS {
  unidad de longitud
  unidad de fuerza
  unidad angular
  unidad de temperatura
  unidad de tiempo
}
    
```

en donde

unidad de longitud =	<u>INCHES</u> <u>FEET</u> <u>FT</u> <u>CENTIMETERS</u> <u>CMS</u> <u>METERS</u> <u>MTS</u>	unidad de fuerza =	<u>POUNDS</u> <u>LBS</u> <u>KIPS</u> <u>TONS</u> <u>MTONS</u> <u>KILOGRAMS</u> <u>KGS</u>
----------------------	--	--------------------	---

unidad angular =	<u>RADIANS</u> <u>DEGREES</u>	unidad de tiempo =	<u>SECONDS</u> <u>MINUTES</u> <u>HOURS</u>
unidad de temperatura =	<u>FAHRENHEIT</u> <u>CENTIGRADE</u>		

La declaración UNITS es opcional y si no se proporciona se suponen las unidades: INCHES, POUNDS, RADIANS, FAHRENHEIT y SECONDS. Las unidades de los datos durante la entrada y salida pueden ser diferentes.

Las declaraciones 16 y 17 deben ir precedidas por la declaración DELETE TABLE.

#### 16. Declaración DELETE PROPERTIES

Se usa para borrar ciertos valores de todos los renglones. Tiene la forma

DELETE PROPERTIES 'nombre enc<sub>1</sub>' 'nombre enc<sub>2</sub>' . . .

en donde

'nombre enc' es el nombre de la propiedad o encabezado de la columna que se borra, al igual que todos los valores numéricos de esa columna.

#### 17. Declaración DELETE ITEMS

Con esta declaración se borran renglones o elementos de la tabla y tiene la forma

DELETE ITEMS 'nombre<sub>1</sub>' 'nombre<sub>2</sub>' . . .

en donde

'nombre<sub>1</sub>' indica el nombre del elemento o renglón que se borra, al igual que todos los valores numéricos de ese renglón.

Las declaraciones 18 y 19 deben ir precedidas por la declaración ORDER TABLE.

#### 18. Declaración ORDER BY

Se utiliza para el ordenamiento de columnas y tiene la forma

ORDER BY { DECREASING  
INCREASING } 'nombre'

en donde si se usa DECREASING el ordenamiento es en orden descendente teniendo como primer valor al valor máximo y utilizando INCREASING el orden es ascendente teniendo como primer valor al valor mínimo. Además, 'nombre' indica el nombre de la propiedad o renglón en donde se efectúa el ordenamiento.

#### 19. Declaración ARRANGE

Esta declaración permite que el ordenamiento de una columna o propiedad sea el mismo que el de otra ya ordenada. Tiene la forma

ARRANGE 'nombre<sub>2</sub>' (ON) (ORDERING) 'nombre<sub>1</sub>'  
(WITH) (ID) 'ident'



en donde

'nombre<sub>1</sub>' indica un ordenamiento existente (nombre de una propiedad o columna)

'nombre<sub>2</sub>' indica el nombre de otra columna cuyos valores se arreglarán según el ordenamiento de 'nombre<sub>1</sub>'

'ident' es el identificador del nuevo arreglo.

Las declaraciones 18 y 19 proporcionan información para efectuar diseños tentativos en forma secuencial. Las dos declaraciones proporcionan información secuencial y el valor numérico de una propiedad (columna) para cada elemento (renglón) en la sucesión. Con la declaración ARRANGE se proporcionan valores de otra propiedad con esa secuencia.

## 20. Declaración TRANSFER ITEMS

Esta declaración permite la transferencia de datos de una tabla a otra y tiene la forma

$$\text{TRANSFER ITEMS} \left\{ \begin{array}{l} \text{'elem}_1' \text{ 'elem}_2' \dots \\ \text{FROM } n_1 \text{ TO } n_2 \\ \text{FROM 'elem}_1' \text{ TO 'elem}_2' \end{array} \right\}$$

en donde

'elem<sub>1</sub>' indica el nombre del elemento (renglón), que va a transferirse, en la primera tabla.

n<sub>1</sub> es la posición, en la primera tabla, del primer elemento del bloque que va a transferirse.

n<sub>2</sub> es la posición, en la primera tabla, del último elemento del bloque que va a transferirse. Tanto n<sub>1</sub> como n<sub>2</sub> deben ser enteros y n<sub>2</sub> debe ser mayor que n<sub>1</sub>.

Las declaraciones 21 a 23 deben ir precedidas por la declaración OUTPUT.

## 21. Declaración PRINT ó PUNCH ORDERING

Para la impresión o perforación de ordenamiento se usa la declaración

$$\left( \begin{array}{l} \text{PRINT} \\ \text{ORDERING 'ordenam}_1' \text{ 'ordenam}_2' \\ \text{PUNCH} \end{array} \right)$$

en donde

'ordenam<sub>1</sub>' es el identificador del ordenamiento que se desea imprimir o perforar. Si no hay identificador todos los ordenamientos existentes se procesan. En es-

te proceso un arreglo se considera como un ordena\_  
miento.

## 22. Declaración PRINT ó PUNCH PROPERTIES

Con esta declaración se especifica la salida de una propiedad y tie-  
ne la forma

$$\left. \begin{array}{l} \text{PRINT} \\ \text{PUNCH} \end{array} \right\} \text{PROPERTIES 'prop}_1 \text{' 'prop}_2 \text{' } \dots$$

en donde

'prop<sub>1</sub>' indica el nombre de una propiedad (columna) que va a  
imprimirse o perforarse para los elementos (renglo-  
nes) que se especifican en la declaración 23, de mane-  
ra que es una declaración de información.

## 23. Declaración PRINT ó PUNCH ITEMS

Se usa para la salida de propiedades de un bloque de elementos. Tie-  
ne la forma

$$\left. \begin{array}{l} \text{PRINT} \\ \text{PUNCH} \end{array} \right\} \text{ITEMS } \left\{ \begin{array}{l} \text{'elem}_1 \text{' 'elem}_2 \text{' } \dots \\ \text{FROM 'elem}_1 \text{' TO 'elem}_2 \text{' } \\ \text{FROM } n_1 \text{ TO } n_2 \end{array} \right\}$$

en donde

'elem<sub>1</sub>' es el nombre del elemento (renglón) que se va a im-  
primir o perforar.

n<sub>1</sub> indica la posición del primer elemento del bloque que  
va a imprimirse o perforarse.

n<sub>2</sub> indica la posición del último elemento del bloque que  
va a imprimirse o perforarse. Tanto n<sub>1</sub> como n<sub>2</sub>  
deben ser enteros y n<sub>2</sub> debe ser mayor que n<sub>1</sub>.

Finalmente, las unidades para los datos de salida pueden especificar-  
se con la declaración UNITS utilizada también para los datos de en-  
trada.