

01161
3
24

ORFEO-90

PROCESADOR NUMERICO DIDACTICO
PARA ANALISIS ESTRUCTURAL
VER 1.50

POR

CARLOS ENRIQUE MONTOYA DULCHE

MANUAL DEL USUARIO

MEXICO 1990

TESIS CON
FALLA DE ORIGEN



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

INDICE

AGRADECIMIENTOS	1
INDICE	11
CAPITULO 1.- EL ENTORNO DE ORFEO-90	
1.1 INTRODUCCION	1-1
1.2 REQUERIMIENTOS	1-2
1.3 DISPOSICION DEL PROGRAMA	1-2
1.4 MENUS DE OPERACION	1-3
1.4.1 ARCHIVOS	1-4
1.4.2 EDITOR	1-5
1.4.3 EJECUTAR	1-5
1.4.4 MODO	1-6
1.5 LAS TECLAS CALIENTES	1-7
1.6 EDICION DE TEXTOS	1-9
1.7 SINTAXIS GENERAL	1-9
CAPITULO 2.- FUNCIONES BASICAS	
2.1 ASIGNACION Y OPERACION DE VARIABLES	2-1
2.2 FUNCIONES MATEMATICAS	2-2
2.3 FUNCIONES ESPECIALES	2-2
2.3.1 FUNCION @MAXV	2-2
2.3.2 FUNCION @SUM	2-3
2.3.3 FUNCION @E	2-3
2.3.4 FUNCION @INT	2-3
CAPITULO 3.- COMANDOS MATRICIALES BASICOS	
3.1 TIPOS DE MATRICES	3-1
3.2 FUNCION LEMAT	3-2
3.3 CONTROL E IMPRESION DE MATRICES	3-6
3.4 FUNCIONES PARA DUPLICACION Y MANIPULACION DE MATRICES	3-8

3.5 GENERACION DE DATOS EN MATRICES	3-10
3.6 OPERACIONES MATRICIALES BASICAS	3-11
3.6.1 ARITMETICA DE MATRICES	3-11
3.6.2 FUNCIONES AUXILIARES	3-13
3.7 GRABACION Y RECUPERACION DE MATRICES	3-14
CAPITULO 4.- ENSAMBLE Y SOLUCION DE SISTEMAS DE ECUACIONES LINEALES	
4.1 FUNCION @ENSB	4-1
4.2 SOLUCION DE SISTEMAS DE ECUACIONES LINEALES	4-3
4.2.1 COMANDO GAUSS	4-3
4.2.2 COMANDO CHOLESKI	4-4
4.2.3 COMANDO INVERTIR	4-5
4.3 FUNCIONES AUXILIARES	4-6
4.3.1 VALUA	4-6
4.3.2 @FRZ	4-6
CAPITULO 5.- COMANDOS PARA DINAMICA ESTRUCTURAL	
5.1 VECTORES Y VALORES CARACTERISTICOS	5-1
5.2 ANALISIS DINAMICO MODAL	5-1
5.3 ANALISIS PASO A PASO	5-2
5.4 FUNCIONES AUXILIARES	5-4
CAPITULO 6.- GRAFICACION	
6.1 REPRESENTACIONES GRAFICAS EN ORFEO-90	6-1
6.2 CONFIGURACION DE GRAFICAS	6-2
6.3 GRAFICACION DE DATOS MATRICIALES	6-2
6.4 GRAFICACION DE EXPRESIONES MATEMATICAS	6-4
6.4.1 COMANDO GRAF2D	6-4
6.4.2 COMANDO GRAF3D	6-6
CAPITULO 7.- MISCELANEA DE FUNCIONES	
7.1 INICIACION DE VARIABLES	7-1

7.2 LISTADO DE UN PROGRAMA	7-1
7.3 COMANDO PAUSA	7-1
7.4 MEMORIA DISPONIBLE	7-1
7.5 LISTADO DEL DIRECTORIO ACTIVO	7-2
CAPITULO 8.- PROGRAMAS DE APOYO	
8.1 INSTALACION	8-1
8.1.1 DISCOS FLEXIBLES	8-1
8.1.2 DISCO RIGIDO	8-2
8.2 CONFIGURADOR CNFORFEO.COM	8-3
8.3 PROTECCION Y DESPROTECCION DE ARCHIVOS	8-3
8.4 PROCESAMIENTO DE GRAFICAS GRABADAS EN DISCO	8-4
8.5 ANIMACION	8-4
8.5.1 ANIMA.EXE	8-5
8.5.2 SISMO.EXE	8-8
8.5.3 BINARIO.EXE	8-9
APENDICE A PROGRAMAS DE EJEMPLO PARA EL MANEJO DE ORFEO-90	
APENDICE B PROGRAMAS DE APLICACION DE ORFEO-90	
APENDICE C MATRICES DE RIGIDECES Y TRANSFORMACION	
C.1 BARRA SOMETIDA A FUERZA AXIAL EN EL PLANO	C-1
C.2 BARRA SOMETIDA A FUERZA AXIAL EN EL ESPACIO	C-2
C.3 BARRA SOMETIDA A FUERZA AXIAL Y MOMENTO FLEXIONANTE EN EL PLANO	C-3
C-4 BARRA SOMETIDA A FUERZA AXIAL Y MOMENTOS FLEXIONANTE Y TORSIONANTE EN EL ESPACIO	C-5
APENDICE D TABLA DE ERRORES	
APENDICE E TECNICAS DE PROGRAMACION UTILIZADAS EN EL DESARROLLO DE ORFEO-90	
E.1 LENGUAJE DE PROGRAMACION	E-1
E.2 ENTORNO DE LAS COMPUTADORAS PERSONALES	E-2

E.3 ANALIZADORES DE LENGUAJE	E-3
E.4 ANALIZADOR DESCENDENTE RECURSIVO	E-4
E.5 TECNICAS DE PROGRAMACION PARA ANALIZADORES DESCENDENTES RECURSIVOS	E-5
E.6 MANEJO DE LA MEMORIA	E-10
E.7 ESTRUCTURAS DE DATOS	E-15
E.8 ALMACENAMIENTO Y REFERENCIA DE DATOS MATRICIALES EN VECTORES	E-15

EL ENTORNO DE ORFEO-90

1

1.1 INTRODUCCION

ORFEO-90 es un paquete de cómputo orientado al proceso numérico de datos, matrices y algoritmos empleados en el análisis estructural con un enfoque didáctico, por lo que resulta de gran utilidad para toda persona que se encuentre realizando estudios en este ámbito.

El marco general del programa obedece a dos modos de operación; el interactivo y el programado. En el primer caso se realizan las funciones solicitadas por el usuario inmediatamente después de ser tecleadas en el editor que para tal efecto se ha desarrollado. En el modo programado es posible realizar una lista de operaciones mediante un procesador de textos incorporado, para posteriormente ejecutarlas en secuencia sin requerir, salvo en el caso de gráficos, de una acción directa del usuario.

El programa posee un analizador descendente recursivo de expresiones matemáticas, lo que permite, además de utilizar las funciones propias de una calculadora electrónica, efectuar operaciones más complejas como la integración numérica de expresiones. En todos los casos es posible asignar el resultado de las funciones a cualquiera de las veintiseis variables globales (A-Z) que se mantienen en memoria.

En el manejo de matrices se ha incorporado un riguroso control de memoria para un aprovechamiento óptimo de los recursos de la computadora. De ello se desprende la distinción de varios tipos de matrices entre los que destacan enteras, reales, simétricas, diagonales y en banda.

La posibilidad de hacer prácticamente cualquier operación matricial proporciona gran versatilidad al manejo de datos. Se cuenta con operaciones que van desde funciones básicas como suma, resta y multiplicación, hasta operaciones de alto nivel como la inversión de matrices y la solución de sistemas de ecuaciones lineales, almacenados de manera tradicional o en banda.

Una de las principales innovaciones del programa es la posibilidad de manejar matrices de tipo algebraico, las cuales tienen la particularidad de ser simétricas y estar compuestas por elementos que representan expresiones matemáticas como $E^*A/L^*COS(X)^2$, introduciendo con ello la posibilidad de definir prácticamente cualquier tipo de matriz de rigideces.

Se cuenta con un ensamblador de matrices sumamente versátil que permite la adición de una matriz de cualquier tipo a una matriz global de tipo real o en banda atendiendo a los indicadores de ecuación almacenados por renglones o columnas dentro de una matriz de números enteros, en un vector con las mismas características o incluso mediante la definición directa de cada indicador de ecuación.

Para realizar estudios en dinámica estructural, se han incorporado al programa funciones que permiten la obtención de vectores y valores característicos, la solución de sistemas de ecuaciones diferenciales desacopladas de segundo orden (Integral de Duhamel) y el análisis paso a paso. En todos los casos se proporcionan las funciones necesarias para el pre y postprocesamiento de datos, incluyendo un grupo de funciones gráficas para la representación de la respuesta obtenida e incluso la animación de esta última en el caso de tratarse de sistemas de un grado de libertad.

Además de la factibilidad de graficar expresiones en dos dimensiones se cuenta con la posibilidad de representar funciones tridimensionales, utilizando para ello las proyecciones isométrica y diamétrica, con la opción de rotarlas alrededor de cualquiera de los ejes coordenados.

En todo caso existe la posibilidad de salvar y/o cargar matrices,

gráficas y programas, en disco.

El paquete ha sido desarrollado de tal forma que presente la mayor facilidad y comodidad al usuario. Todas las opciones que ofrece son implementadas de la manera más sencilla posible mediante menús de operación. Si durante el manejo del programa se presenta algún problema, se ha incorporado un detector de errores que, aunado a las ayudas incluidas, proporcionan una buena orientación al usuario.

1.2 REQUERIMIENTOS

ORFEO-90 puede ser ejecutado en toda la gama de computadoras IBM PC incluyendo los modelos XT, AT y PS/2, o en máquinas compatibles. Requiere aproximadamente de 200 Kb de memoria RAM libre. Es compatible con las tarjetas gráficas para video CGA, MCGA, EGA, VGA y MDA (Hércules).

Aunque la configuración mínima, en cuanto a lectoras de disco se refiere, es de una, de 360 Kb, se recomienda tener dos unidades, o en el mejor de los casos disco rígido.

El programa está compilado con un emulador para coprocesador matemático 8087 lo que permite utilizarlo en caso de encontrarse instalado. No es necesario, pero sí recomendable.

En cuanto al equipo de impresión, ORFEO-90 soporta cualquier tipo de impresora, inclusive laser, en cuanto a datos y programas se refiere. Sin embargo, para gráficas es necesario equipo EPSON serie MX o compatible.

1.3 DISPOSICION DEL PROGRAMA

El paquete se proporciona en dos discos de 5 1/4", de 360 Kb, con el siguiente contenido:

DISCO 1 SISTEMA

ORFEO-90.EXE	Programa principal no ejecutable hasta después de la instalación.
ORFEO.PNT	Archivo parcial de pantallas. Para uso exclusivo de ORFEO-90.EXE.
ORFEO_A.COM	Ayudas. Programa para uso exclusivo de ORFEO-90.EXE.
CNFORFEO.COM	Configurador de ORFEO-90.EXE
*.BGI	Manejadores de video.
LITI.CHR	Manejador del tipo de letra en modo gráfico.

DISCO 2 UTILERIAS

ANIMA.EXE	Animador de respuesta sísmica para sistemas de un grado de libertad.
-----------	--

BINARIO.EXE	Convertidor de archivos formato ASCII a binario.
GRAFICAR.EXE	Impresor y visualizador de gráficas.
SISMO.EXE	Animador de sistemas de un grado de libertad ante un acelerograma.
PLIB.COM	Protector de librerías y programas.
DLIB.COM	Desprotector de librerías y programas.
E-nn.PRG	Ejemplos para el manejo de ORFEO-90.
A-nn.PRG	Ejemplos de aplicación para análisis estructural.
*.BIN	Sismos en formato binario.
*.MAT	Matrices de rigideces.
INSTALAR.EXE	Programa de instalación.

1.4 MENUS DE OPERACION

Para el manejo general del programa se han implementado una serie de menús de fácil operación.

La ventana principal de ORFEO-90 se presenta en la figura 1.1.

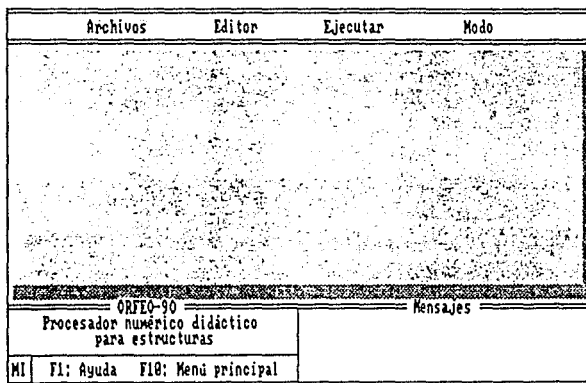


Figura 1.1 Ventana principal de ORFEO-90

En la parte superior de la pantalla aparecen las cuatro funciones principales; Archivos, Editor, Ejecutar y Modo. En la porción inferior derecha se distingue la ventana de mensajes y, a la izquierda, dos pequeñas regiones; el indicador de modo que por defecto se encuentra en interactivo MI, y las

"teclas calientes" , F1 destinada a las ayudas y F10 al retorno del menú principal.

1.4.1 ARCHIVOS

Al elegir la opción Archivos, se da paso a una nueva pantalla, definida en la figura 1.2, con las siguientes operaciones:

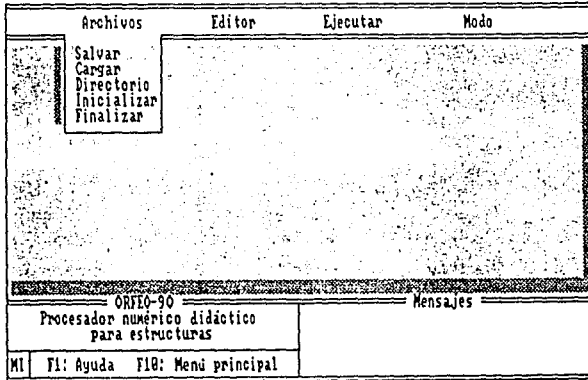


Figura 1.2 Ventana de la opción Archivos.

Salvar

Función orientada a grabar en disco los datos, o el programa, en edición, dependiendo del modo en que se esté utilizando. En el caso de encontrarse el sistema trabajando en modo interactivo MI, se grabarán todas las matrices definidas hasta el momento, a diferencia del modo programado MP, en cuyo caso será grabado el programa en edición.

Al activarse la opción salvar, aparece una nueva pantalla, figura 1.3. En esta parte el programa solicita el nombre del archivo destinado a albergar la grabación. Junto a la palabra salvar aparece centelleando un indicador de modo que recuerda al usuario el tipo de información que se almacenará. El nombre no debe de tener extensión ya que el programa asigna internamente INT o PRG atendiendo al modo interactivo o programado, respectivamente.

Cargar

Esta opción recupera datos o programas, funcionando con las mismas consideraciones hechas para Salvar.

Directorio

Define el directorio activo para grabar o recuperar información. Por defecto se asume que el directorio inicial es desde donde se invocó ORFEO-90.

Iniciar

Borra los datos definidos o el programa en edición, de acuerdo con el modo en que se esté trabajando. En modo interactivo se borran las matrices de memoria principal; en programado, el programa en edición.

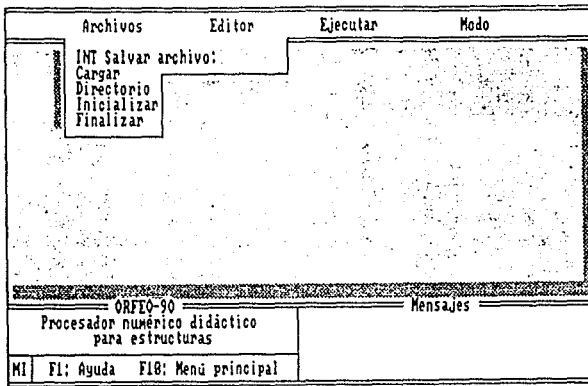


Figura 1.3 Submenú de grabación. Opción Salvar.

Finalizar

Abandona la ejecución de ORFEO-90 perdiendo todos los datos y el programa definido en ese momento. Antes de salir del programa, la región de mensajes realiza una confirmación. Si realmente se desea abandonar el programa se debe presionar la tecla ESC de lo contrario, cualquier otra.

1.4.2 EDITOR

El editor es el punto de comunicación entre el usuario y el programa. Puede operar de dos formas distintas, nuevamente atendiendo al tipo de modo que se esté utilizando. En la figura 1.4 se muestra el editor interactivo. Se sitúa en la porción inferior izquierda de la pantalla principal. La comunicación con el programa se lleva a cabo por medio de instrucciones de no más de 37 caracteres. Cuando se sobrepasa este límite es necesario recurrir al editor programado, figura 1.5. Este se compone de 150 renglones de 77 caracteres cada uno, en donde se pueden definir varios comandos y ejecutarlos en forma secuencial. Al trabajar en este editor la porción inferior izquierda de la ventana principal proporciona la posición dentro de la pantalla.

1.4.3 EJECUTAR

Esta opción ejecuta las funciones definidas en el editor programado en forma secuencial, de derecha a izquierda y de arriba a abajo. No es aplicable en modo interactivo, y en este caso se reportará el mensaje de error

correspondiente.

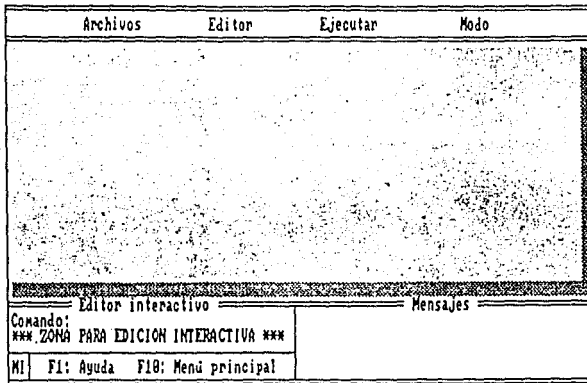


Figura 1.4 Editor interactivo.

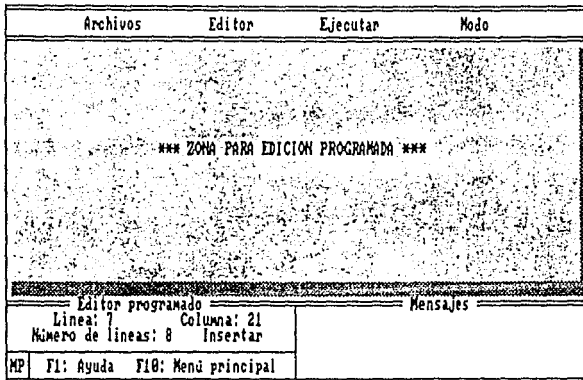


Figura 1.5 Editor programado.

1.4.4 MODO

Proporciona las opciones mostradas en la figura 1.6. Para definir el modo deseado, sólo es necesario posicionar la barra de selección y presionar <return>. En ese momento el indicador de modo, situado en el extremo inferior izquierdo de la ventana principal, asumirá las claves MI o MP si el modo

activo es interactivo o programado, respectivamente.

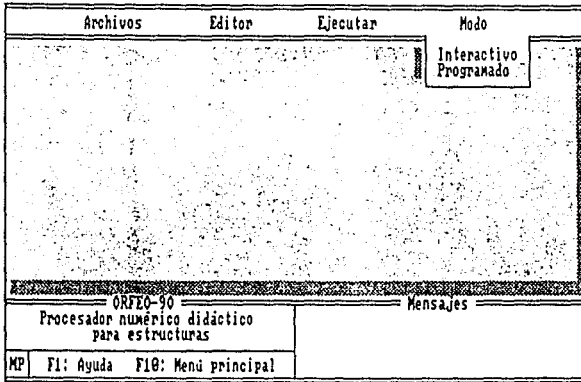


Figura 1.6 Submenú para selección de modo activo.

1.5 LAS TECLAS CALIENTES

Se les llama teclas calientes "hot keys" a aquellas que realizan una función completa al presionarlas solas o en combinación con CTRL o ALT.

ORFEO-90 posee varias funciones definidas de esta manera. A continuación se enumeran en su totalidad; sin embargo, cuando se traten los tópicos para las cuales fueron diseñadas, se entrará en detalle.

- F1:** Proporciona el menú de ayudas mostrado en la figura 1.7. La información concerniente a la selección hecha mediante las flechas del teclado es reportada tal y como se muestra en la figura 1.8.
- F2:** Salva el programa en edición sin necesidad de salir del editor mediante F10. Sólo es aplicable en modo programado. Si no se ha definido algún archivo de salida, el programa preguntará por él, de lo contrario grabará directamente en el archivo definido por última vez.
- F10:** Retorna el control al menú principal. Se utiliza para salir de cualquiera de los editores.

En modo gráfico; funciones GRAF...

- Alt-F1:**
Alt-F2: Salvan una gráfica de pantalla a disco, siempre y cuando el equipo de video lo permita. Para mayor información consulte el capítulo 6.
Alt-F3:
Alt-F4:
Alt-F5:

Archivos	Editor	Ejecutar	Modo		
AYUDA					
ALMDG	EJEV	INVERTIR	MAX	SUBTITULO	@FUNC
ALMSA	ESCALA	JACOBI	MEMC	SUM	@FUNR
BORPAT	GAUSS	LEMAT	MEML	TITULO	@GNC
CARGA	GRAF3D	LIMPAU	MODIF	TMULT	@GENR
CHOLESKI	GRAF2D	LISTAD	MULT	TRANS	@ITP
DUP	GRAFXY	LISTAM	PAUSA	TMULT	@PASO
DUPDCM	GRAFXY	LISTAMI	RAIZ	VALDA	@RSC
DUPDCU	INPMAT	LISTAP	REMOH	EDIM	EDITOR
DUPSM	INPMAT	LISTAV	RES	@MSB	ERRORES
EJEX	INVERTIR	LISTAVI	SALVA	@FRZ	FUNCIONES
ORFEO-90					
Procesador numérico didáctico para estructuras				Mensajes	
MP	F1: Ayuda	F10: Menú principal			

Figura 1.7 Menú de ayudas.

Archivos	Editor	Ejecutar	Modo		
AYUDA					
ALMDG	@PASO				@FUNC
ALMSA	Realiza el análisis paso a paso de una estructura.				@FUNR
BORPAT	$M a(t) + C v(t) + K u(t) = F P(t)$, almacenando los				@GNC
CARGA	desplazamientos $u(t)$ en la matriz U .				@GENR
CHOLESKI	@PASO X M C UVA U FAC Fo INT NPNTS dt, s, m, 0				@ITP
DUP	continúa...				@PASO
DUPDCM	INVERTIR	LISTAVI	SALVA	@FRZ	@RSC
DUPDCU					EDITOR
DUPSM					ERRORES
EJEX					FUNCIONES
ORFEO-90					
Procesador numérico didáctico para estructuras				Mensajes	
MI	F1: Ayuda	F10: Menú principal			

Figura 1.8 Formato para reportar ayudas.

Alt-F10: Imprime la gráfica a papel, siempre y cuando se cuente con impresora compatible y en servicio.

En la función GRAF3D

F1: Accede al menú para manejo de gráficas tridimensionales. Para mayor información consulte el capítulo 6.

- F2:** Tipo de proyección Axonométrica: Isométrica o Diamétrica.
- F3:** Parámetros de la gráfica.
- F4:** Incremento del ángulo de rotación para edición.
- F5:** Edición del ángulo θ_x .
- F6:** Edición del ángulo θ_y .
- F7:** Edición del ángulo θ_z .
- F8:** Tipo de escala: proporcional o no proporcional.

1.6 EDICION DE TEXTOS

En todo lugar donde se requiera proporcionar algún dato de entrada, las teclas comunes de edición de texto estarán activas.

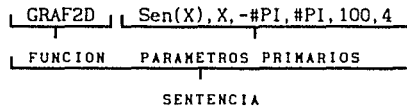
- | | |
|------------------|--|
| Del | Suprime el carácter del cursor. |
| Ins | Activa y desactiva el modo de inserción. |
| Home | Posiciona el cursor en el inicio de la línea. |
| End | Posiciona el cursor al final de la línea. |
| PgUp | Avanza a la página inmediata superior. |
| PgDn | Avanza a la página inmediata inferior. |
| Bksp | Regresa el cursor borrando el carácter. |
| Ctrl Home | Posiciona el cursor en la primera línea de la página en edición. |
| Ctrl End | Posiciona el cursor en la última línea de la página en edición. |
| Ctrl PgUp | Avanza a la primera página de la edición. |
| Ctrl PgDn | Avanza a la última página de la edición. |
| Ctrl Y | Borra la línea en edición. |

1.7 SINTAXIS GENERAL

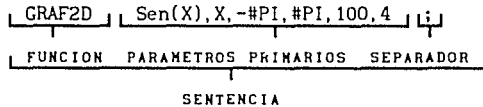
En general la sintaxis de las funciones de ORFEO-90 es la misma para los dos modos de operación. Sin embargo, dadas las características de los editores, se debe hacer la siguiente distinción.

En el modo interactivo las *sentencias* son escritas sin separador (;) al

término de ésta. Se llama *sentencia* a una función con sus parámetros primarios. Por ejemplo:



Cuando se trabaja en modo programado, es necesario utilizar el separador (;) para delimitar las diferentes *sentencias*. Utilizando el ejemplo anterior:



Esto permite escribir los *programas* (sentencias en orden secuencial dentro del editor programado) con una mayor claridad. Por ejemplo:

```

    GRAF2D Sen(X),X,
           -#PI , #PI,
           100, 4;
  
```

En los *programas* escritos en ORFEO-90 es posible poner comentarios siempre y cuando se encuentren entre comillas (son válidas las anidaciones) y estén fuera de cualquier *sentencia*.

Ejemplo:

"Se grafica la función seno(x) de -PI a PI en cien intervalos con el formato gráfico número cuatro..."

```

    GRAF2D Sen(X),X,-#PI,#PI,100,4;
  
```

"Comentario anidado " "... Comentario ..." "

```

    GRAF2D Sen(X),"... Comentario inválido ..."X,-#PI,#PI,100,4;
  
```

ORFEO-90 no detecta diferencia entre mayúsculas y minúsculas en funciones y variables globales; sin embargo, sí realiza esta distinción en los nombres que definen a las matrices utilizadas. De esta manera escribir la *sentencia*:

```

    GRAF2D COS(X),X,-#PI,#PI,100,4;
  
```

es completamente equivalente a:

```

    Graf2d Cos(x),x,-#PI,#PI,100,4;
  
```

Sin embargo, las matrices:

```

    KG1 Kg1 kg1 kG1
  
```

son enteramente distintas para el programa.

FUNCIONES BASICAS

2

2.1 ASIGNACION Y OPERACION DE VARIABLES

ORFEO-90 cuenta con veintiseis variables globales que permanecen en memoria durante el tiempo de ejecución. Para definir las basta realizar una asignación sintáctica tradicional como:

$$A = 56.78 \quad B = A$$

o incluso:

$$C = D = E = F = 1.0$$

en cuyo caso todas las variables adquirirán el valor 1.0.

Las operaciones que con estas variables se pueden efectuar, obedecen a la aritmética común, pudiéndose asignar valores dados por:

$$K = 5 + (B - 4) * H \quad \text{ó} \quad Z = X / 7 / (4 + A)$$

La jerarquía de operadores es la siguiente:

- | | | |
|---|------------------------------|-----|
| 1 | Operaciones entre paréntesis | () |
| 2 | Exponenciación | ^ |
| 3 | Multiplicación y división | * / |
| 4 | Suma y resta | + - |

Cuando se ejecuta el programa en modo interactivo, toda operación aritmética es reportada en la ventana de mensajes, aun y cuando se trate de una simple asignación. En el caso del modo programado, la asignación es realizada internamente. Para observar el valor de las variables, puede utilizarse el comando LISTAV, lo que genera una lista de todos los valores actuales en memoria (figura 2.1). Si se cuenta con impresora en servicio, es posible obtener dicha lista impresa mediante el comando LISTAVI.

Archivos	Editor	Ejecutar	Modo
A=	5.00000E+00	H=	0.00000E+00
B=	5.00000E+00	O=	0.00000E+00
C=	5.00000E+00	P=	0.00000E+00
D=	0.00000E+00	Q=	0.00000E+00
E=	0.00000E+00	R=	0.00000E+00
F=	0.00000E+00	S=	0.00000E+00
G=	0.00000E+00	T=	0.00000E+00
H=	0.00000E+00	U=	0.00000E+00
I=	0.00000E+00	V=	0.00000E+00
J=	0.00000E+00	W=	0.00000E+00
K=	0.00000E+00	X=	0.00000E+00
L=	0.00000E+00	Y=	0.00000E+00
M=	0.00000E+00	Z=	0.00000E+00

Editor interactivo		Mensajes
Comando: LISTAV		
MI Trabajando en función: LISTAV	Para continuar presione cualquier tecla	

Figura 2.1 Comando LISTAV.

2.2 FUNCIONES MATEMATICAS

Las funciones matemáticas disponibles en el programa se enumeran a continuación:

ABS(n)	Proporciona el valor absoluto de n.
ACOS(n)	Calcula el ángulo cuyo coseno es n.
ALN(n)	Calcula el antilogaritmo natural de n.
ALOG(n)	Calcula el antilogaritmo base 10 de n.
ASEN(n)	Calcula el ángulo cuyo seno es n.
ATAN(n)	Calcula el ángulo cuya tangente es n.
FAC(n)	Calcula el factorial de n.
LN(n)	Calcula el logaritmo natural de n.
LOG(n)	Calcula el logaritmo base 10 de n.
SEN(n)	Calcula el seno de n.
TAN(n)	Calcula la tangente de n.
#PI	Proporciona el número $\Pi = 3.141592653589\dots$
#E	Proporciona el número $e = 2.718281828459\dots$

ORFEO-90 trabaja siempre a los números reales como variables de doble precisión.

En todos los casos es posible realizar operaciones con números y variables. Los siguientes ejemplos muestran este aspecto:

```
A = SEN( #PI/2 )
B = LOG( A + 5 )
C = ( B * A / ABS( COS(X) ) ) ^ 2
```

Si el procedimiento matemático presenta algún error, el programa lo reportará inmediatamente en la ventana de mensajes.

2.3 FUNCIONES ESPECIALES

ORFEO-90 cuenta con funciones *asignables* de mayor nivel. El término *asignable* se aplica a funciones que retornan un valor almacenable en una variable global.

De esta manera se tienen las siguientes funciones especiales:

```
@MAXV
@SUM
@E
@INT
```

2.3.1 FUNCION @MAXV

La función @MAXV proporciona el máximo valor absoluto de un vector columna. El formato de uso es:

```
@MAXV ( nombre del vector )
```

El comando es aplicable también a arreglos de varias columnas (matrices)

con la salvedad de proporcionar el máximo valor absoluto de la primera columna. Algunos ejemplos de aplicación son:

$$A = @MAXV(Vec1)$$
$$B = @MAXV(V1) + 2 * @MAXV(V2)$$

2.3.2 FUNCION @SUM

Esta función realiza la suma de los elementos de un vector columna. El formato de uso es:

$$@SUM (nombre del vector)$$

Al igual que la función @MAXV, es posible aplicar el comando a matrices, calculándose la suma de elementos sobre la primera columna. Ejemplos:

$$A = @SUM(Vec1)$$
$$B = 2 * @SUM(V1) + @SUM(V2)$$

2.3.3 FUNCION @E

La función @E representa una poderosa herramienta para manejar datos dentro de una matriz. Su objetivo es proporcionar el valor numérico de un elemento posicionado dentro de una matriz de cualquier tipo salvo matrices con números enteros. El formato de uso es:

$$@E(M, i, j)$$

donde

M nombre de la matriz
i número de renglón
j número de columna

La importancia de esta función estriba en la posibilidad de obtener el resultado numérico de una expresión matemática almacenada en una matriz algebraica. En el próximo capítulo se tratará con detalle este tipo de matrices. Algunos ejemplos de aplicación son:

$$A = @E(K1, 1, 2)$$
$$B = @E(K2, 1, 3) / @E(K3, 2, 2)$$

2.3.4 FUNCION @INT

@INT permite realizar la integración numérica de una expresión matemática usando distintos métodos. El formato de uso es:

$$@INT(EXPRESION, VARIABLE, INFERIOR, SUPERIOR, N, TIPO)$$

La sintaxis del formato se interpreta como "Integral de EXPRESION con respecto a VARIABLE de INFERIOR a SUPERIOR en N intervalos con el método TIPO".

Los diferentes métodos de integración numérica disponibles en ORFEO-90 se enumeran a continuación:

Clave para el parámetro *TIPO*

S2	Simpson 1/2.
S3	Simpson 1/3.
S8	Simpson 3/8.
G1	Cuadratura de Gauss-Legendre grado 1.
G2	Cuadratura de Gauss-Legendre grado 2.
G3	Cuadratura de Gauss-Legendre grado 3.
G4	Cuadratura de Gauss-Legendre grado 4.
G5	Cuadratura de Gauss-Legendre grado 5.

Como es sabido, el número de intervalos de integración *N* depende del método empleado:

S2 *N* cualquiera.
 S3 *N* número par.
 S8 *N* múltiplo de tres.

Aun y cuando no es necesario para fines de cálculo un número de intervalos de integración para la cuadratura de Gauss-Legendre, por sintaxis se requiere proporcionar una *N* mayor que tres.

En todo caso se revisan los posibles errores en el proceso de integración y evaluación de la expresión, reportándose cualquier problema en la ventana de mensajes de la pantalla principal.

Ejemplos:

A = @INT(SEN(X),X,-#PI,#PI,50,S2)

Realiza la integración numérica de la función seno de x con respecto a x de $-\pi$ a π en 50 intervalos utilizando el criterio de Simpson 1/2, asignando el resultado a la variable global A.

B = @INT(COS(Y) + 3 * C,Y,2,8.5,10,G5)

Realiza la integración numérica de la función $\text{COS } Y + 3C$ con respecto a Y de 2 a 8.5 mediante el criterio de Gauss-Legendre de grado 5. Nótese que por sintaxis se ha proporcionado un número de intervalos mayor a tres (10), aun y cuando no se requiere. En el momento de evaluar la expresión para integración, la variable C toma el valor que tenga en ese momento en memoria.

La función @INT soporta anidaciones, con lo que es posible calcular integrales múltiples, con la ventaja de utilizar para cada una de ellas diferentes métodos de integración. Por ejemplo:

$$\int_{-\pi}^{\pi} \int_{-\pi/2}^{\pi/4} y \text{ sen } x \text{ dx dy}$$

se puede calcular como:

$\text{@INT}(\text{@INT}(Y*\text{SEN}(X), X, -\#PI, \#PI, 24, S8), Y, -\#PI/2, \#PI/4, 10, S2)$

COMANDOS MATRICIALES BASICOS

3

3.1 TIPOS DE MATRICES

ORFEO-90 maneja una gran variedad de tipos de matrices, con dos finalidades; la primera obedece a la función que desempeñan, y la segunda a la optimización de memoria. De esta forma se cuenta con:

Matrices reales

Compuestas por números reales. Almacenan $n \times m$ elementos, siendo n el número de renglones y m el número de columnas.

Matrices enteras

Compuestas por números enteros. Almacenan $n \times m$ elementos.

Matrices simétricas

Contienen números reales. Internamente se almacenan $n(n+1)/2$ elementos.

Matrices diagonales

Poseen números de tipo real. Sólo se almacenan en memoria n elementos (diagonal principal).

Matricesseudodiagonales

Compuestas por números reales. Almacenan la diagonal principal y un elemento adicional que representa todos los valores fuera de ésta. La cantidad de elementos en memoria es de $n+1$.

Matrices en banda

Se componen también de números reales. Se almacenan $n \times mb$ elementos, en donde mb es el ancho de banda. Para su definición, los valores son referidos como en una matriz real; sin embargo, al utilizarse en la solución de sistemas de ecuaciones, la disposición de elementos es en banda.

Matrices identidad

Se componen por un elemento real unitario.

Matrices algebraicas

Se componen por expresiones matemáticas de un máximo de 55 caracteres cada una, son simétricas y almacenan $n(n+1)/2$ expresiones. Están orientadas a definir matrices de rigideces.

En todos los casos, el acceso a los elementos de una matriz obedece a la notación índice tradicional; es decir, se puede gestionar cualquier valor, dentro del orden válido, sin importar el número de elementos que se almacene en el arreglo. Por ejemplo, supóngase que se tiene una matriz identidad de orden 5. Como se ha mencionado, almacena sólo un elemento. Si se requiere para algún procedimiento el elemento 3,3, el programa regresará un uno. Si se gestiona el elemento 4,3, el programa proporcionará un cero. La decisión de los valores que se deben tomar es internamente resuelta por el programa y transparente para el usuario.

3.2 FUNCION LEMAT

Para definir una matriz en la memoria principal, se cuenta con dos opciones, dependiendo del modo en el que se esté trabajando. Para cada una de ellas la sintaxis varía ligeramente. A continuación se muestran los dos formatos posibles en esta tarea.

Para el modo interactivo:

```
LEMAT <nombre> <tipo>
```

Trabajando en modo programado:

```
LEMAT <nombre> <tipo> R= <renglones> C= <columnas>
```

o bien, si se trata de una matriz cuadrada:

```
LEMAT <nombre> <tipo> O=<orden>
```

Como se puede observar, los parámetros primarios comunes en todas las sentencias son <nombre> y <tipo>.

<nombre> Secuencia de no más de cuatro caracteres que identifica a una matriz.

<tipo> Clave que determina el tipo de matriz.

TIPO DE MATRIZ

CLAVE

Real	R
Entera	E
Simétrica	S
Diagonal	D
Seudodiagonal	SD
En banda	B
Algebraica	A
Real autoinicialable a cero	RC
En banda autoinicialable a cero	BC
Identidad autoinicialable	DI

Cuando se trabaja en modo interactivo los parámetros de la matriz, tales como número de renglones y columnas, así como datos, son pedidos secuencialmente por el programa. La figura 3.1 muestra un ejemplo de este aspecto.

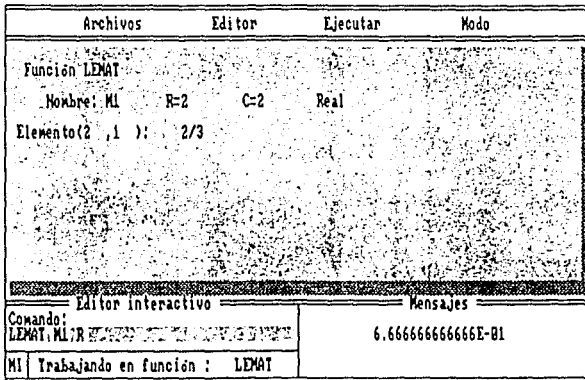


Figura 3.1 Lectura de la matriz M1 de tipo real (R) en modo interactivo.

En modo programado, la sentencias para lectura de matrices tienen algunos parámetros primarios adicionales.

- R= <renglones> número de renglones.
- C= <columnas> número de columnas.

o en el caso de matrices cuadradas:

- O= <orden> orden de la matriz.

Es importante aclarar que los datos de la matriz no son redundantes. Por ejemplo, si se requiere leer los elementos de una matriz simétrica, sólo se proporcionarán aquellos que estén sobre la diagonal principal, incluyendo, claro está, ésta última.

A continuación se proporciona un ejemplo de lectura para cada tipo de matriz en modo programado, con los comentarios pertinentes en cada caso.

Matriz real (R)

Código	Matriz definida
LEMAT M1 R R=3 C=2; 1.0; 17; 1/2; 1/4; A=SEN(0); COS(0);	$\begin{bmatrix} 1.0 & 17.0 \\ 0.5 & 0.25 \\ 0.0 & 1.0 \end{bmatrix}$

Como se puede observar las sentencias son evaluadas al generarse la matriz. El elemento 3,1 (A=SEN(0)) realiza una asignación y, al mismo tiempo, retorna el valor de la misma, el cual es introducido dentro de la matriz definida.

Matriz entera (E)

Código	Matriz definida
LEMAT M2 E R=1 C=3; 1; 2; 3;	$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$

Matriz simétrica (S)

Código	Matriz definida
LEMAT M3 S O=3; 1.0; 2.0; 3.0; 4.0; 5.0; 6.0;	$\begin{bmatrix} 1.0 & 2.0 & 3.0 \\ 2.0 & 4.0 & 5.0 \\ 3.0 & 5.0 & 6.0 \end{bmatrix}$

Nótese que por tratarse de una matriz simétrica sólo se proporcionan los elementos requeridos. Así mismo, la definición del orden se hace directamente como O=3; sin embargo, se pudo emplear R=3 y C=3 sin ningún problema.

Matriz diagonal (D)

Código	Matriz definida
LEMAT M4 D O=3; 1.0; 2.0; 3.0;	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 2.0 & 0.0 \\ 0.0 & 0.0 & 3.0 \end{bmatrix}$

Nuevamente se observa la no redundancia en la definición de los elementos.

Matriz pseudodiagonal (SD)

Código	Matriz definida
LEMAT M5 SD O=3; 1.0; 2.0; 3.0; -1.0;	$\begin{bmatrix} 1.0 & -1.0 & -1.0 \\ -1.0 & 2.0 & -1.0 \\ -1.0 & -1.0 & 3.0 \end{bmatrix}$

Matriz en banda (B)

Código
LEMAT M6 B R=3 C=2; 1.0; 2.0; 3.0; 4.0; 5.0; 0.0;

Este tipo de matrices sólo funciona adecuadamente en algunos comandos como CHOLESKI, @PASO y @ENSB. En estos casos la distribución de elementos dentro del arreglo se considera como:

$$\begin{bmatrix} 1.0 & 2.0 & 0.0 \\ 2.0 & 3.0 & 4.0 \\ 0.0 & 4.0 & 5.0 \end{bmatrix}$$

En las funciones restantes la definición de la matriz es exactamente igual que en las matrices reales. De esta forma, si se quiere acceder a los elementos del arreglo, la disposición de los mismos obedecerá a:

$$\begin{bmatrix} 1.0 & 2.0 \\ 3.0 & 4.0 \\ 5.0 & 0.0 \end{bmatrix}$$

Matriz algebraica (A)

Código	Matriz definida
LEMAT K A O=2; 4*E*I/L; 2*E*I/L; 4*E*I/L;	$\begin{bmatrix} \frac{4EI}{L} & \frac{2EI}{L} \\ \frac{2EI}{L} & \frac{4EI}{L} \end{bmatrix}$

Obsérvese que las expresiones son almacenadas considerando la no redundancia, por tratarse de una matriz simétrica.

Al operar con estos arreglos el programa substituye los valores de las variables que se encuentren en memoria, obteniéndose datos numéricos. Con éstos se efectúan las operaciones en que la matriz se encuentre involucrada.

Matrices autoiniciables (RC) (BC) (DI)

Las matrices definidas con las claves RC y BC, se comportan igual que las matrices de tipo real; y en banda, respectivamente. La diferencia estriba en que no es necesario proporcionar las sentencias que definen a los datos; de ahí su carácter *autoinicialable*.

Código	Matriz definida
LEMAT KG1 RC O=3;	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix}$

Código	Matriz definida
LEMAT KG2 BC R=3 C=3;	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix}$

En el caso del tipo DI la generación automática de la matriz obedece a la identidad.

Código	Matriz definida
LEMAT IDN DI O=3;	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$

Como función auxiliar a LEMAT se cuenta con MODIF. Este comando permite modificar un elemento en particular, dentro de una matriz de cualquier tipo. El formato de uso es:

`MODIF <matriz> <renglón> <columna> <expresión>`

Ejemplo:

`MODIF M1 2 2 COS(#PI);`

Con esta sintaxis se modificará el elemento 2,2 de la matriz M1. Si este arreglo es de tipo algebraico se almacenará la expresión COS(#PI), en caso contrario el resultado de evaluarla. Cuando se trata con arreglos de tipo entero, no deben utilizarse expresiones, ya que estas reportan número reales.

3.3 CONTROL E IMPRESION DE MATRICES

Para determinar el nombre, tamaño, tipo y cantidad de matrices definidas en memoria, así como los datos que éstas contienen, ORFEO-90 cuenta con una serie de comandos que permiten control minucioso en cuanto a este aspecto se refiere.

IMPMAT

Esta función se encarga de proporcionar una lista de los elementos que constituyen una matriz. El formato de uso es:

IMPMAT <nombre de la matriz>

En la figura 3.2 se muestra el formato de salida de este comando.

Ejemplo:

IMPMAT M3

IMPMATI

Opera exactamente igual a la función anterior, con la salvedad de realizar la salida por la impresora.

	Archivos	Editor	Ejecutar	Modo	
Nombre: M3	R=5	C=5	Real		
	Col 1	Col 2	Col 3	Col 4	Col 5
Ren 1	2.00000E+00	2.00000E+00	2.00000E+00	2.00000E+00	2.00000E+00
Ren 2	3.00000E+00	3.00000E+00	3.00000E+00	3.00000E+00	3.00000E+00
Ren 3	4.00000E+00	4.00000E+00	4.00000E+00	4.00000E+00	4.00000E+00
Ren 4	5.00000E+00	5.00000E+00	5.00000E+00	5.00000E+00	5.00000E+00
Ren 5	6.00000E+00	6.00000E+00	6.00000E+00	6.00000E+00	6.00000E+00

ORIG-90 Mensajes

Procesador numérico didáctico para estructuras

MP Trabajando en función : IMPMAT Para continuar presione cualquier tecla

Figura 3.2 Formato de salida del comando IMPMAT.

LISTAM

Proporciona una lista de las matrices definidas hasta el momento. El formato de uso es:

LISTAM

La figura 3.3 muestra el resultado de emplear este comando.

El número máximo de matrices que pueden estar definidas al mismo tiempo es de 50, siempre y cuando la cantidad de memoria requerida no supere 41 Kb, aproximadamente.

LISTAMI

Funciona de manera análoga al comando anterior, pero la salida se efectúa por la impresora. El formato de uso es:

LISTAMI

Archivos	Editor	Ejecutar	Modo
Nombre	#Renglones	#Columnas	Tipo
Mpsc	3	1	Real
Mabs	3	1	Real
UAUX	20	3	Real
M	3	1	Real
C	3	1	Real
Cp	3	1	Real
So	6	2	Real
U(1)	3	20	Real
RES	21	5	Real
9 Arreglos definidos		41 Arreglos disponibles	
Editor interactivo		Mensajes	
Comando: LISTAM			
MI Trabajando en función : LISTAM		Para continuar presione cualquier tecla	

Figura 3.3 Formato de salida del comando LISTAM.

BORMAT

Borra una matriz de la memoria principal. Una vez aplicado el comando no es posible recuperar los datos. El formato de uso es:

BORMAT <nombre de la matriz>

Ejemplo:

BORMAT M7

RENOM

Cambia el nombre de una matriz. El formato de uso es:

RENOM <nombre original> <nuevo nombre>

Ejemplo:

RENOM K1 K2

En el caso de que el nuevo nombre coincida con el de alguna matriz ya definida, el comando no procede y se reporta el error correspondiente.

3.4 FUNCIONES DE DUPLICACION Y MANIPULACION DE MATRICES

ORFEO-90 permite obtener copias totales y parciales de matrices, así como el almacenaje, por substitución, de submatrices en arreglos mayores.

DUP

Duplica una matriz. El formato de uso es:

DUP <origen> <destino>

Ejemplo:

DUP M1 M2

la matriz M2 es creada a partir de los elementos del arreglo M1.

DUPSM

Forma una matriz a partir de una porción de un arreglo determinado. El formato de uso es:

DUPSM <origen> <submatriz> <#ren> <#col> <i> <j>

Ejemplo:

DUPSM K Kg 4 4 1 1

Se crea la submatriz Kg de 4 renglones y 4 columnas con los datos de la matriz K a partir de la posición 1,1.

DUPDGM

Crema una matriz diagonal con los elementos de la diagonal principal de un arreglo determinado. El formato de uso es:

DUPDGM <matriz de origen> <matriz diagonal>

Ejemplo:

DUPDGM M Dm

Se forma la matriz Dm de tipo diagonal (D) con los elementos de la diagonal principal de la matriz M.

DUPDGV

Forma una matriz columna (vector) con los elementos de la diagonal principal de un arreglo determinado. El formato de uso es:

DUPDGV <matriz> <vector>

Ejemplo:

DUPDGV M1 Vec

ALMDG

Almacena la diagonal principal de una matriz en otra. El formato de uso es:

ALMDG <origen> <destino>

Ejemplo:

ALMDG M1 M2

Se almacena la diagonal de M1 en M2. Para que el comando proceda, es necesario que las matrices sean cuadradas y que el orden de la matriz de origen tenga orden menor o igual a la de destino.

ALMSM

Almacena una submatriz dentro de un arreglo determinado. El formato de uso es:

ALMSM <submatriz> <destino> <i> <j>

Ejemplo:

ALMSM M1 M2 2 2

Se almacena la submatriz M1 en M2 a partir de la posición 2,2. Si por alguna razón el tamaño de la submatriz supera al orden del arreglo <destino>, el mensaje de error correspondiente será reportado, sin que el comando proceda.

3.5 GENERACION DE DATOS EN MATRICES

Existen cuatro funciones que permiten generar secuencias de números o funciones dentro de una matriz real. Estas son:

@GENC

Genera una secuencia de números reales sobre la columna de un arreglo a partir de una posición determinada. El formato de uso es:

@GENC <matriz> <i> <j> <inicio>, <incremento>

Ejemplo:

@GENC M 3 4 1.5, 1/10

Se genera sobre la columna 4, a partir del renglón 3, una secuencia de reales comenzando en 1.5 e incrementándose 0.1 unidades. Nótese que el incremento puede estar dado por una expresión matemática.

@GENR

Trabaja de manera análoga a la función anterior, pero define la secuencia sobre un renglón. Formato:

@GENR <matriz> <i> <j> <inicio>, <incremento>

Ejemplo:

@GENR N 2 3 0, 1.0

Sobre el renglón 2 y a partir de la columna 3 de la matriz N, se genera una secuencia de reales iniciando en cero e incrementándose en uno hasta el final del arreglo (número de columnas de la matriz N).

@FUNC

Valúa una expresión matemática considerando como variable independiente a los elementos de una columna del arreglo. El resultado de la operación es colocado en cualquier otra columna de la matriz. El formato de uso es:

`@FUNC <matriz> <i> <j> <variable>, <expresión>, <columna de términos ind.>`

Ejemplo:

`@FUNC R 3 2 X, SEN(X), 1`

Se evalúa la expresión seno de X, tomando como variable independiente X a los elementos de la columna 1 comenzando por el renglón 3. Los resultados son colocados sobre la columna 2 a partir del renglón 3.

@FUNR

Funciona de igual manera que el comando @FUNC, con la diferencia de operar sobre los renglones de la matriz. El formato de uso es:

`@FUNR <matriz> <i> <j> <variable>, <expresión>, <renglón de términos ind.>`

Ejemplo:

`@FUNR H 4 1 Y, COS(Y)+C, 1`

Asigna sobre el renglón 4 de la matriz H, el resultado de evaluar la expresión $\text{COS}(Y)+C$, tomando como variable independiente a los elementos del renglón 1.

3.6 OPERACIONES MATRICIALES BASICAS

3.6.1 ARITMETICA DE MATRICES

SUM

Realiza la adición de dos matrices. el formato de uso es:

`SUM <sumando 1> <sumando 2> <resultado (opcional)>`

Ejemplo:

`SUM M1 M2 M3`

Se realiza la operación $M3 = M1 + M2$. En el caso de no ser proporcionado el último parámetro primario (M3), el resultado de la adición es colocado en la primera matriz (M1), es decir; $M1 = M1 + M2$.

RES

Efectúa la substracción de dos matrices. Formato:

RES <minuendo> <sustraendo> <diferencia (opcional)>

Ejemplo:

RES M1 M2 M3

Se ejecuta la operación $M3 = M1 - M2$. Si no se proporciona el último parámetro primario <diferencia> (M3), el resultado es almacenado en el <minuendo> (M1), esto es; $M1 = M1 - M2$.

En ambos casos los comandos deciden internamente el tipo de matriz resultante. Por ejemplo, si M1 y M2 son simétricas el resultado es de tipo simétrico. Si M1 es diagonal y M2 es real, el resultado tiene el tipo real. En todo caso se revisa la conformabilidad de matrices, reportándose oportunamente cualquier error.

MULT

Efectúa el producto entre dos matrices. Formato:

MULT <factor 1> <factor 2> <producto>

Ejemplo:

MULT M1 M2 M3

Se realiza la operación $M3 = M1 \times M2$.

TMULT

Efectúa el producto matricial entre la transpuesta del factor 1 y el factor 2. El formato de uso es:

TMULT <factor 1> <factor 2> <producto>

Ejemplo:

TMULT B D AUX

Se realiza la operación $AUX = B^t D$

TTMULT

Realiza el triple producto matricial dado por la transpuesta del factor 1 por factor 2 por factor 1. El formato de uso es:

TTMULT <factor 1> <factor 2> <resultado>

Ejemplo:

TTMULT a k K

Se realiza la operación $K = a^t k a$

En todas las funciones de multiplicación de matrices se revisa la conformabilidad, reportándose cualquier error.

Si alguna de las matrices es de tipo algebraico, la operación gestionada tendrá efecto, utilizando los valores que se obtengan de substituir las variables en memoria dentro de las expresiones matemáticas que conforman a la matriz algebraica.

3.6.2 FUNCIONES AUXILIARES

TRANS

Transpone una matriz. El formato de uso es:

TRANS <matriz> <matriz transpuesta (opcional)>

Ejemplo:

TRANS A AT

La transpuesta de la matriz A es almacenada en el arreglo AT. En caso de no proporcionarse el último parámetro primario (AT), la transpuesta es almacenada sobre la misma matriz.

ESCALA

Escala una matriz. El formato de uso es:

ESCALA <matriz> <factor de escala>

Ejemplo:

ESCALA K1 E*A/L

Se multiplica cada uno de los elementos de la matriz K1 por el resultado de la expresión E*A/L.

RAIZ

Substituye los elementos de una matriz por su raíz cuadrada. El modo de empleo es:

RAIZ <matriz> <* (opcional)>

Ejemplo:

RAIZ M1 *

Se substituyen los elementos de la matriz M1 por su raíz cuadrada. El carácter * indica que se tome el valor absoluto de ellos. En caso de no ser proporcionado este asterisco, se realiza una revisión del signo en los datos

del arreglo M1. Si existe algún elemento negativo, el comando no surte efecto y se reporta el mensaje de error correspondiente.

INVERTER

Substituye los elementos de una matriz por sus inversos. Formato:

INVERTER <matriz> <* (opcional)>

Ejemplo:

INVERTER M1 *

Se substituyen los elementos de la matriz M1 por sus inversos $1/M1(i,j)$. El caracter * indica que los elementos nulos deben ser ignorados. En caso de no ser proporcionado el asterisco, se revisan los datos de la matriz. Si alguno de ellos resulta ser nulo, el comando no opera y se reporta el error correspondiente a división entre cero.

3.7 GRABACION Y RECUPERACION DE MATRICES

Cualquier tipo de matriz es susceptible de ser almacenada en disco para posteriormente recuperarla. Para tal efecto se utilizan los comandos:

SALVAR Graba una matriz en el directorio activo.
CARGAR Recupera una matriz del directorio activo.

Formatos:

SALVAR <nombre de la matriz>
CARGAR <nombre de la matriz>

Quando no es posible grabar o cargar una matriz, el programa despliega el mensaje de error explicativo.

Si es proporcionado en la función SALVAR el nombre de algún arreglo que se encuentre grabado con anterioridad, el programa substituye al archivo por su nueva versión. Para evitar el borrado accidental de un arreglo valioso (una matriz de rigideces algebraica por ejemplo), ORFEO-90 cuenta entre sus utilerías con dos programas; PLIB.COM y DLIB.COM. Para mayor información consulte el capítulo 8.

ENSAMBLE Y SOLUCION DE SISTEMAS DE ECUACIONES LINEALES

4

4.1 FUNCION @ENSB

Para conformar sistemas de ecuaciones a partir de la contribución de submatrices, ORFEO-90 cuenta con un versátil ensamblador, el cual es descrito a continuación.

@ENSB

Este comando ensambla submatrices en una matriz global atendiendo a los indicadores de ecuación correspondientes. El formato de uso de la función presenta varias posibilidades:

- A).- @ENSB <submatriz> <matriz global> <matriz indicadora> R=<ren>
- B).- @ENSB <submatriz> <matriz global> <matriz indicadora> C=<col>
- C).- @ENSB <submatriz> <matriz global> <vector indicadora> V=<pos>
- D).- @ENSB <submatriz> <matriz global> <arg 1> <arg 2> ... <arg n>

La <submatriz> puede ser de cualquier tipo, exceptuando por supuesto a las matrices enteras y en banda. En caso de ser algebraica, los valores ensamblados obedecerán a la substitución de las variables activas en memoria, dentro de las expresiones que conforman al arreglo.

La <matriz global> puede ser de dos tipos. Real o en banda. De acuerdo con el tipo de matriz seleccionado, el programa decide internamente el modo de ensamble, proporcionando sistemas de ecuaciones almacenados de forma tradicional o en banda.

La <matriz indicadora>, así como el <vector indicadora> deben ser de tipo entero. Atendiendo a la modalidad de ensamble ,A) ó B), la <matriz indicadora> contiene los indicadores de ecuación dispuestos por renglones o columnas respectivamente. En el caso de emplear el formato C), los indicadores de ecuación se encuentran almacenados secuencialmente en el <vector indicadora>.

En estos tres formatos, el último parámetro primario de la sentencia representa la posición en donde se almacenan los indicadores de ecuación para ensamble. De esta manera:

- Caso A) <ren> Número de renglón donde se sitúan los indicadores de ecuación.
- Caso B) <col> Número de columna donde se sitúan los indicadores de ecuación.
- Caso C) <pos> Señala la posición de inicio de los indicadores de ecuación dentro del vector secuencial.

El número de grados de libertad para ensamble se toma a partir del número de columnas de la <submatriz>. Esto permite no sólo el ensamble de sistemas de ecuaciones, sino también de vectores o matrices de términos independientes.

El último formato, D), es el llamado directo, ya que permite asignar explícitamente cada indicador de ecuación.

Para ejemplificar lo anteriormente expuesto, considere la existencia de las submatrices K1 y K2 que a continuación se presentan:

$$K1 = \begin{bmatrix} 4.0 & 2.0 \\ 2.0 & 4.0 \end{bmatrix} \quad K2 = \begin{bmatrix} 2.0 & 1.0 \\ 1.0 & 2.0 \end{bmatrix}$$

Así mismo, supóngase que se tienen las matrices globales KG y KGB, del tipo real (RC) y en banda (BC), respectivamente, iniciadas a cero, y las matrices indicadoras RI CI y VI (vector columna).

De esta manera, como punto de partida se tiene:

$$KG = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} \quad KGB = \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix}$$

y los indicadores de ecuación dipuestos por renglones (RI), por columnas (CI) y sobre un vector secuencial (VI).

$$RI = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} \quad CI = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} \quad VI = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

De la submatrices se deduce que el número de grados de libertad a considerar es de 2. Nótese que es posible tener submatrices de diferentes órdenes, en cuyo caso, el número de grados de libertad por submatriz es tomado individualmente para cada ensamble.

De esta manera, para ensamblar la primera submatriz K1 se tiene:

Formato A) Matriz indicadora por renglones

①ENSB K1 KG RI R=1 matriz completa
 ①ENSB K1 KGB RI R=1 matriz en banda

Formato B) Matriz indicadora por columnas

①ENSB K1 KG CI C=1 matriz completa
 ①ENSB K1 KGB CI C=1 matriz en banda

Formato C) Vector secuencial indicador

①ENSB K1 KG VI V=1 matriz completa
 ①ENSB K1 KGB VI V=1 matriz en banda

Formato D) Asignación directa

①ENSB K1 KG 1 2 matriz completa
 ①ENSB K1 KGB 1 2 matriz en banda

Bajo cualquiera de estos formatos, el resultado obtenido será:

$$KG = \begin{bmatrix} 0.0 + 4.0 & 0.0 + 2.0 & 0.0 \\ 0.0 + 2.0 & 0.0 + 4.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} = \begin{bmatrix} 4.0 & 2.0 & 0.0 \\ 2.0 & 4.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix}$$

$$KGB = \begin{bmatrix} 0.0 + 4.0 & 0.0 + 2.0 \\ 0.0 + 4.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix} = \begin{bmatrix} 4.0 & 2.0 \\ 4.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix}$$

Considerando ahora que los indicadores de ecuación para ensamble de la submatriz K2 son 2 y 3:

Formato A) Matriz indicadora por renglones

```
@ENSB K2 KG RI R=2
@ENSB K2 KGB RI R=2
```

Formato B) Matriz indicadora por columnas

```
@ENSB K2 KG CI C=2
@ENSB K2 KGB CI C=2
```

Formato C) Vector secuencial indicador

```
@ENSB K2 KG VI V=3
@ENSB K2 KGB VI V=3
```

Formato D) Asignación directa

```
@ENSB K2 KG 2 3
@ENSB K1 KGB 2 3
```

De esta manera, con la aplicación del comando @ENSB en cualquiera de sus formatos, las matrices resultantes serán:

$$KG = \begin{bmatrix} 4.0 & 2.0 & 0.0 \\ 2.0 & 4.0 + 2.0 & 0.0 + 1.0 \\ 0.0 & 0.0 + 1.0 & 0.0 + 2.0 \end{bmatrix} = \begin{bmatrix} 4.0 & 2.0 & 0.0 \\ 2.0 & 6.0 & 1.0 \\ 0.0 & 1.0 & 2.0 \end{bmatrix}$$

$$KGB = \begin{bmatrix} 4.0 & 2.0 \\ 4.0 + 2.0 & 0.0 + 1.0 \\ 0.0 + 1.0 & 0.0 \end{bmatrix} = \begin{bmatrix} 4.0 & 2.0 \\ 6.0 & 1.0 \\ 2.0 & 0.0 \end{bmatrix}$$

4.2 SOLUCION DE SISTEMAS DE ECUACIONES LINEALES

Para resolver sistemas de ecuaciones lineales, ORFEO-90 está equipado con tres comandos:

4.2.1 COMANDO GAUSS

Esta función resuelve sistemas de ecuaciones lineales almacenados en matrices de cualquier tipo exceptuando a las enteras y en banda. El comando soporta múltiples vectores de términos independientes dispuestos por columnas dentro de un arreglo de tipo real. El formato de uso es:

GAUSS <система> <términos independientes>

GAUSS almacena la solución del sistema en la matriz de <términos independientes>. Cabe mencionar que el arreglo que alberga el sistema de ecuaciones no sufre modificación alguna.

Ejemplo:

Tomando la matriz real KG ensamblada en el apartado anterior, y considerando que se requiere resolver el sistema para las condiciones:

$$\begin{bmatrix} 1.25 \\ 2.48 \\ 3.26 \end{bmatrix} \quad \begin{bmatrix} 2.51 \\ 8.24 \\ -4.28 \end{bmatrix}$$

almacenada en la matriz real X de la forma:

$$\begin{bmatrix} 1.25 & 2.51 \\ 2.48 & 8.24 \\ 3.26 & -4.28 \end{bmatrix}$$

El comando GAUSS adquiere la forma:

GAUSS KG X

Prácticamente en forma instantánea, el sistema de ecuaciones es resuelto, quedando la solución almacenada en la matriz X tal y como se muestra a continuación:

$$\begin{bmatrix} 2.87500E-01 & -3.86388E-01 \\ 5.00000E-02 & 2.02777E+00 \\ 1.60500E+00 & 3.15388E+00 \end{bmatrix}$$

4.2.2 COMANDO CHOLESKI

En el caso de haberse ensamblado un sistema de ecuaciones lineales sobre una matriz en banda, el comando CHOLESKI permite resolverlo con tres opciones. El formato general de uso es:

CHOLESKI <sis^{te}ma> <términos independientes>

Primera opción:

CHOLESKI <sis^{te}ma>

Bajo esta circunstancia el comando opera sobre la matriz <sis^{te}ma> triangulándola, y poniendo en ella un indicador interno de tipo que la acredita con estas características.

Segunda opción:

Si la matriz del sistema se encuentra triangulada el comando realiza las sustituciones hacia adelante e inversa con la matriz de <términos independientes>.

CHOLESKI <sis^{te}ma triangulado> <términos independientes>

Tercera opción:

En el caso de no encontrarse triangulada la matriz <sis^{te}ma> y ser proporcionado el arreglo de <términos independientes>, el comando realiza la triangulación y las sustituciones hacia adelante e inversa.

CHOLESKI <sis^{te}ma no triangulado> <términos independientes>

En todo caso, la matriz que almacena al sistema de ecuaciones debe ser de tipo en banda (B) o (BC), y el vector o matriz de términos independientes de tipo real (R).

Ejemplo:

Tomado la matriz ensamblada en banda del apartado 4.1, se tiene:

CHOLESKI KGB

Al aplicar el comando de esta manera, la matriz KGB es triangulada, modificándose sus elementos a:

$$\begin{bmatrix} 4.00 & 0.50 \\ 5.00 & 0.20 \\ 1.80 & 0.00 \end{bmatrix}$$

Supongamos que se requiere resolver el sistema bajo las condiciones del comando GAUSS:

$$\begin{bmatrix} 1.25 & 2.51 \\ 2.48 & 8.24 \\ 3.26 & -4.28 \end{bmatrix}$$

Entonces la función toma la sintaxis:

CHOLESKI KGB X

almacenándose en X la solución ya conocida del sistema de ecuaciones.

Si la matriz KGB se hubiera proporcionado sin triangular junto con el arreglo de términos independientes X, se habría obtenido el mismo resultado.

En el caso de cometerse algún error debido a que la matriz < sistema > es singular o el vector de términos independientes es inconformable, el programa reportará cualquier problema. Sin embargo, es posible que las matrices hayan sido alteradas por el programa. Los comandos DUP... o SALVAR pueden ayudar en estas circunstancias.

4.2.3 COMANDO INVERTIR

Aunque sólo es necesario aplicar este comando en situaciones particulares para la solución de sistemas de ecuaciones, INVERTIR permite obtener la inversa de una matriz. El formato de uso es:

INVERTIR <matriz> <inversa (opcional)>

Ejemplo:

INVERTIR KG KGI

La matriz inversa de KG se almacena en KGI. En el caso de no ser proporcionado el segundo parámetro primario (KGI), la matriz (KG) es substituida por su inversa.

4.3 FUNCIONES AUXILIARES

Aunque no pertenecen directamente al proceso de solución de sistemas de ecuaciones lineales, las funciones VALUA y @FRZ se relacionan en este aspecto en el análisis estructural.

4.3.1 VALUA

Genera una matriz real simétrica con los datos obtenidos de substituir las variables pertinentes en las expresiones que conforman una matriz algebraica. El formato de uso es:

VALUA <algebraica> <real simétrica>

Ejemplo:

Supóngase que se tiene la matriz KB algebraica definida por:

$$KB = \begin{bmatrix} \frac{EA}{L} & \frac{-EA}{L} \\ \frac{-EA}{L} & \frac{EA}{L} \end{bmatrix}$$

y que $E = A = L = 2.0$.

Al aplicar el comando tal y como se muestra a continuación:

VALUA KB K1

la matriz K1 será definida como un arreglo simétrico:

$$K1 = \begin{bmatrix} 2.0 & -2.0 \\ -2.0 & 2.0 \end{bmatrix}$$

4.3.2 @FRZ

Este comando calcula los elementos mecánicos de cualquier miembro estructural, siempre y cuando se cuente con su matriz de rigideces, de transformación, y los desplazamientos del sistema.

Como es sabido, la obtención de estos elementos se realiza mediante la operación:

$$\{ f \}_e = [k]_e \{ U \}_e$$

donde

$\{ f \}_e$ es el vector de elementos mecánicos del elemento e.

$[k]_e$ es la matriz de rigideces del elemento e en coordenadas locales.

$\{ U \}_e$ es el vector de desplazamientos en coordenadas locales.

Por otra parte, cuando se ensambla la matriz global de rigideces, las contribuciones de cada elemento se refieren al sistema coordenado global. Esto se logra transformando la matriz de rigideces local de la siguiente manera:

$$[K]_e = [A]_e^t [k]_e [A]_e$$

donde

- [K]_e es la matriz de rigideces del elemento asociada al sistema global de referencia.
- [A]_e es la matriz de transformación o matriz de continuidad.

De esta manera los desplazamientos obtenidos de la solución del sistema de ecuaciones { d }, están referidos al sistema global. Por ello, para calcular los elementos mecánicos en coordenadas locales, se necesita que los desplazamientos sean también transformados a este marco de referencia. Matricialmente ésto es posible mediante el producto:

$$\{ U \}_e = [A]_e \{ d \}_e$$

en donde { d }_e son los desplazamientos del elemento en coordenadas globales.

Lo anteriormente expuesto es realizable en ORFEO-90 mediante la función @FRZ, la cual se describe a continuación:

Formatos:

- A) @FRZ <F> <T> <K> <d> <IND> R= <ren>
- B) @FRZ <F> <T> <K> <d> <IND> C= <col>
- C) @FRZ <F> <T> <K> <d> <IND> V= <pos>
- D) @FRZ <F> <T> <K> <d> <arg 1> <arg 2> ... <arg n>

donde

- <F> vector o matriz de elementos mecánicos
- <T> matriz de transformación
- <K> matriz de rigideces
- <d> vector o matriz de desplazamientos
- <IND> matriz indicadora de ecuación
- <ren> renglón de la matriz <IND> que almacena los indicadores de ecuación
- <col> columna de la matriz <IND> que almacena los indicadores de ecuación
- <Pos> posición del vector secuencial <IND> que almacena los indicadores de ecuación
- <arg> argumentos enteros que definen directamente los indicadores de ecuación

La función opera creando un vector, o una matriz, con los elementos mecánicos de un miembro estructural, obtenidos a partir de los desplazamientos almacenados en $\{d\}$. La diferencia entre vector o matriz se debe a que la función puede atender a diversas condiciones de carga (número de columnas de la matriz $\{d\}$).

Para ilustrar completamente la forma de operación de la función, considere el siguiente ejemplo:

Supóngase que el vector de desplazamientos de una estructura referido al sistema global $\{d\}$, la matriz de rigideces local del elemento $[k]_e$ y la matriz de transformación $[A]_e$ del mismo, están dadas por:

$$\{d\} = \begin{bmatrix} 1.0 \\ 2.0 \\ 3.0 \\ 4.0 \\ 5.0 \end{bmatrix} \quad [k]_e = \begin{bmatrix} 4.0 & -4.0 \\ -4.0 & 4.0 \end{bmatrix} \quad [A]_e = \begin{bmatrix} 0.7 & 0.7 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.7 & 0.7 \end{bmatrix}$$

Así mismo, considérese que los grados de libertad del elemento estructural en cuestión son 1 3 2 5 (indicadores de ecuación).

Ya se ha mencionado que para obtener los elementos mecánicos se requiere efectuar el producto matricial:

$$\{F\}_e = [k]_e \{U\}_e = [k]_e [A]_e \{d\}_e$$

Se construye el vector $\{d\}_e$ a partir de los indicadores de ecuación:

$$\{d\} = \begin{bmatrix} 1.0 \\ 2.0 \\ 3.0 \\ 4.0 \\ 5.0 \end{bmatrix} \begin{matrix} \text{grado de libertad 1} \\ \text{grado de libertad 2} \\ \text{grado de libertad 3} \\ \text{grado de libertad 4} \\ \text{grado de libertad 5} \end{matrix} \quad \{d\}_e = \begin{bmatrix} 1.0 \\ 3.0 \\ 2.0 \\ 5.0 \end{bmatrix} \begin{matrix} \text{grado de libertad 1} \\ \text{grado de libertad 3} \\ \text{grado de libertad 2} \\ \text{grado de libertad 5} \end{matrix}$$

Realiza así la transformación de coordenadas con la matriz de continuidad $[A]_e$

$$\{U\}_e = [A]_e \{d\}_e = \begin{bmatrix} 0.7 \times 1.0 + 0.7 \times 3.0 + 0.0 \times 2.0 + 0.0 \times 5.0 \\ 0.0 \times 1.0 + 0.0 \times 3.0 + 0.7 \times 2.0 + 0.7 \times 5.0 \end{bmatrix}$$

$$\{U\}_e = \begin{bmatrix} 2.8 \\ 4.9 \end{bmatrix}$$

Y los elementos mecánicos serán:

$$\{ f \}_e = [k]_e \{ U \}_e = \begin{bmatrix} 4.0 \times 2.8 - 4.0 \times 4.9 \\ -4.0 \times 2.8 + 4.0 \times 4.9 \end{bmatrix} = \begin{bmatrix} -8.4 \\ 8.4 \end{bmatrix}$$

Para realizar estas operaciones con la sintaxis de ORFEO-90, sólo es necesario escribir, en el caso del formato D):

```
@FRZ F A k d 1 3 2 5
```

considerando, claro está, que $A = [A]_e$, $k = [k]_e$ y $d = \{ d \}$.

En el caso de los formatos A), B) y C), la función @FRZ trabaja exactamente igual que el comando @ENSB.

Finalmente se aclara que la conformabilidad de matrices es revisada internamente por el programa, reportándose cualquier error en la ventana de mensajes de la pantalla principal.

COMANDOS PARA DINAMICA ESTRUCTURAL

5

5.1 VECTORES Y VALORES CARACTERISTICOS

Para resolver el problema de vectores y valores característicos orientados al análisis estructural, se cuenta con el comando JACOBI el cual se describe a continuación.

Formato:

JACOBI <K> <M> <V> <F>

donde

- <K> matriz de rigideces de tipo real (R).
- <M> matriz de masas de tipo real (R) o diagonal (D).
- <V> matriz en cuyas columnas se almacenan los vectores característicos (configuraciones modales).
- <F> vector de valores característicos (cuadrado de las frecuencias circulares).

La matriz de rigideces debe estar conformada con elementos tales que produzcan una matriz simétrica y positiva definida.

Ejemplo:

JACOBI K M V F

Se obtienen las configuraciones modales por columnas dentro de la matriz V ordenadas de menor a mayor frecuencia y normalizadas con respecto a la matriz de masas M. Así mismo se almacenan las frecuencias circulares, al cuadrado en forma ascendente en el vector F.

IMPORTANTE: Después de aplicarse el comando JACOBI las matrices de rigideces y masas se modifican.

5.2 ANALISIS DINAMICO MODAL

Para obtener la solución de sistemas de ecuaciones diferenciales desacopladas de segundo orden de manera exacta (integral de Duhamel), de la forma:

$$\frac{d^2x}{dt^2} + 2c\omega \frac{dx}{dt} + \omega^2 x = fp g(t)$$

ORFEO-90 cuenta con el comando @DIN descrito a continuación:

Formato:

@DIN <W> <C> <Fp> <g(t)> <u(t)> <N> <INT>

donde

- n número de ecuaciones a resolver.
- <W> Vector de frecuencias. (Orden nx1).
- <C> Vector de amortiguamiento expresado como una fracción del amortiguamiento crítico. (Orden nx1).
- <Fp> Factores de participación. (Orden nx1).
- <u(t)> Matriz de respuesta generada automáticamente. (Orden nxN). Cada columna almacena la respuesta del sistema en un tiempo determinado.
- <g(t)> Matriz de excitación . (Orden qx2) donde q es cualquier número.
- <N> Número de intervalos <INT> a almacenar.
- <INT> Intervalo que indica cada cuanto tiempo se desea realizar el almacenaje de la respuesta.

Observaciones:

La primera columna de la matriz <g(t)> es el tiempo y la segunda, la excitación, sin ser necesario que los intervalos de tiempo sean iguales.

No es necesario que la matriz de respuesta <u(t)> se encuentre definida.

IMPORTANTE: Siempre se debe cumplir que $N * INT < g(t) (q, 1)$.

Ejemplo:

@DIN W C Cp So U 20 0.15

Se obtiene la respuesta del sistema cuyas matrices de frecuencias, amortiguamiento y factores de participación se encuentran definidas en W, C, y Cp, respectivamente, ante la acción de la excitación almacenada en So. La respuesta es calculada para 20 puntos con un intervalo de tiempo de 0.15 y almacenada en la matriz U.

5.3 ANALISIS PASO A PASO

Para la resolución de la ecuación incremental de movimiento de un sistema de múltiples grados de libertad expresado como:

$$[M] a(t) + [C] v(t) + [K] u(t) = F_m P(t)$$

se ha incorporado al programa la función @PASO con el siguiente formato:

@PASO <K> <M> <C> <UVA> <U> <fac> <Fo> <INT> <NPTS> <dt> , δ , α , θ

donde

- <K> matriz de rigideces en banda tipo (B)
- <M> matriz de masas, real o diagonal, tipos (R) o (D)
- <C> matriz de amortiguamiento de tipo real o diagonal
- <UVA> matriz de condiciones iniciales almacenadas por columnas; desplazamientos, velocidades y aceleraciones, respectivamente
- <fac> vector que contiene los factores F_m que alteran la excitación.
- <Fo> Arreglo que almacena la excitación (fuerza) en una o dos columnas. Si se encuentra una sola columna, se considera $F_o(i,1) = F(t_i)$.
En el caso de dos o más columnas, la primera representará al tiempo y la siguiente a la excitación.
 $F_o(i,1) = t_i$ y $F_o(i,2) = F(t_i)$.
Es importante recordar que la historia de fuerzas debe estar a intervalos de tiempo constantes.
- <INT> Indica cada cuantos intervalos de tiempo <dt> debe ser almacenada la respuesta.
- <NPTS> Indica el número de puntos que se desean almacenar.
- <dt> Intervalo de tiempo constante.

δ, α, θ Parámetros de integración implícita

Método de integración	δ	α	θ
Aceleración constante	1/2	1/4	1.00
Aceleración lineal	1/2	1/6	1.00
θ amortiguamiento alto	1/2	1/6	1.42
θ amortiguamiento bajo	1/2	1/6	2.00

Observaciones:

Los parámetros de integración mostrados proporcionan una solución incondicionalmente estable. Sin embargo, como estos valores se pueden cambiar arbitrariamente, es posible incurrir en inestabilidad numérica.

Si son suprimidos de la sentencia todos los parámetros de integración numérica el programa supone una variación lineal de la aceleración (Método de Newmark con $\beta = 1/6$).

IMPORTANTE: Siempre se debe cumplir que el producto $NPTS * INT * dt$ sea menor al número de fuerzas $F(t_i) * dt$.

Ejemplo:

@PASO K M C Cl U FAC Ft 2 50 0.05,1/2,1/6,1.0

Aquí se realiza el análisis paso a paso del sistema definido por las matrices de rigideces K, de masas M y de amortiguamiento C. Las condiciones

iniciales se almacenan en C1. La excitación se encuentra en el arreglo Ft y, para cada grado de libertad, es afectada por los elementos de la matriz FAC. La respuesta es calculada para 50 puntos espaciados cada 2 intervalos dt=0.05, es decir; a cada 0.10. El método de integración empleado es el de Newmark 1/6.

Este mismo sistema puede ser integrado de la misma forma (Newmark 1/6) si son suprimidos los parámetros de integración de la sentencia, tal y como se muestra a continuación:

```
@PASO K M C C1 U FAC Ft 2 50 0.05
```

5.4 FUNCIONES AUXILIARES

ORFEO-90 cuenta con las siguientes funciones de apoyo a los comandos @DIN y @PASO:

@IIP

Genera un vector columna con la interpolación a intervalos iguales de la función definida en las dos primeras columnas de una matriz. El formato de uso es:

```
@IIP <matriz> <vector> <Ninter> <dt>
```

donde

<matriz>	Es la matriz que contiene la función a interpolar. En la primera columna tiene a la variable independiente, y en la segunda a la dependiente.
<Vector>	Es el vector generado por la interpolación.
<Ninter>	Representa el número de intervalos que se desea calcular.
<dt>	Intervalo de interpolación.

Ejemplo:

Suponga que se tiene la función definida en la matriz F:

$$F = \begin{matrix} & \mathbf{x} & \mathbf{f(x)} \\ \begin{bmatrix} 0.0 & 0.0 \\ 5.0 & 5.0 \end{bmatrix} \end{matrix}$$

y se desea interpolarla en cinco intervalos iguales de 2.0 unidades, almacenando el resultado en el vector columna Vec.

La sintaxis para tal efecto es:

```
@IIP F V 5 2.0
```

y el resultado:

$$V = \begin{bmatrix} 0.0 \\ 2.0 \\ 4.0 \\ 0.0 \\ 0.0 \end{bmatrix}$$

Nótese que en caso de superarse los valores de la función, el comando rellena con ceros las localidades correspondientes en el vector.

MAX

Genera un vector renglón o columna con el máximo valor absoluto de los elementos de una matriz. El formato de uso es:

MAX <matriz> <vector> <clave>

donde

<matriz> Contiene los elementos de donde se obtendrán los máximos.

<vector> Vector renglón o columna donde se almacenan los máximos valores absolutos.

<clave> Tiene los valores R y C. Indica el sentido de operación del comando. En el caso de ser R, la búsqueda de máximos se lleva a cabo sobre los renglones de la <matriz>, generándose un <vector> columna. Si por el contrario, es C, la búsqueda se realiza sobre las columnas, produciéndose un <vector> renglón.

Ejemplo:

Considere nuevamente la matriz F definida anteriormente:

$$F = \begin{bmatrix} 0.0 & 0.0 \\ 5.0 & 5.0 \end{bmatrix}$$

Si el comando MAX es aplicado como:

MAX F V R

se generará un vector V igual a:

$$V = \begin{bmatrix} 0.0 \\ 5.0 \end{bmatrix}$$

Si la función es escrita como:

MAX F V C

el vector V adquirirá la forma:

V = [5.0 5.0]

ØRCSC

El comando opera de manera análoga al anterior, pero en lugar de obtener el máximo valor absoluto por renglones o columnas, calcula la raíz cuadrada de la suma de los cuadrados de los elementos de la matriz indicada. El formato de uso es:

ØRCSC <matriz> <vector> <clave>

Ejemplo:

Para la matriz F, utilizando la clave R (por renglones):

ØRCSC F V R

Se genera:

$$V = \begin{bmatrix} 0.0 \\ 7.07106 \end{bmatrix}$$

Tomando la clave C:

ØRCSC F V C

Se obtiene:

V = [5.0 5.0]

GRAFICACION

6.1 REPRESENTACIONES GRAFICAS DE ORFEO-90

Como un apoyo a la representación de datos y resultados, ORFEO-90 se encuentra equipado con un versátil grupo de funciones gráficas.

De esta forma es posible graficar:

Los datos almacenados, por columnas, en una matriz.

Funciones de una variable independiente (curvas planas).

Funciones de dos variables independientes (superficies).

En todas las funciones es posible imprimir a papel y grabar las gráficas a disco, siempre y cuando se cumplan con las siguientes condiciones:

Para grabación

Contar con tarjeta gráfica adaptadora CGA, MCGA o MDA (Hércules). En caso de tenerse tarjetas EGA o VGA, la representación gráfica es posible, pero no así la grabación a disco.

Tener espacio disponible en disco. Para tarjeta CGA 16Kb y para MCGA o Hércules 40Kb, aproximadamente.

Para impresión

Poseer impresora EPSON serie MX, o equivalente, conectada y en línea.

La operación de estos procesos se lleva a cabo mediante teclas calientes.

Para grabar gráficas se utilizan las combinaciones Alt-F1 a Alt-F5. Cada una de ellas genera un archivo en disco de acuerdo con la tecla Fn seleccionada. Por ejemplo, si se presiona la combinación Alt-F1 se salva la imagen gráfica en el archivo ORFEO1.GRF, con Alt-F2 el archivo ORFEO2.GRF etc. De esto se desprende la restricción de poder tener en disco sólo 5 gráficas al mismo tiempo.

Para recuperar e imprimir las gráficas salvadas es necesario recurrir al programa de apoyo GRAFICAR.EXE, el cual es descrito en el capítulo 8.

En cuanto a la impresión de imágenes se refiere, sólo existe la restricción del tipo de impresora utilizada; sin embargo, el resultado obtenido puede presentar alguna distorsión por el tipo de monitor empleado. La función de impresión fue diseñada tomando en cuenta la tarjeta CGA, y aun cuando es aplicable al resto de los monitores, no se recomienda. Para salvar este aspecto puede utilizarse cualquier programa residente para mapeo de pantallas a impresora, sin que ORFEO-90 entre en conflicto con él.

Para imprimir una gráfica en pantalla se utilizan la combinación de teclas calientes Alt-F10.

6.2 CONFIGURACION DE GRAFICAS

Cualquiera de las gráficas realizadas por el programa es susceptible de ser configurada mediante cinco formatos definidos a partir de las siguientes funciones:

TITULO[<n>]	Define el título del formato <n>.
SUBTITULO[<n>]	Define el subtítulo del formato <n>.
EJEX[<n>]	Define la leyenda de las abscisas del formato <n>.
EJEY[<n>]	Define la leyenda de las ordenadas del formato <n>.

<n> es el número de formato no menor a uno ni mayor a cinco.

Ejemplo:

Supóngase que se desea configurar el formato 5 para utilizarlo posteriormente en una gráfica.

Considérese que el título debe ser COMANDO GRAFMX, el subtítulo FUNCIONES SEN(X) Y COS(X), y las leyendas para las abscisas y ordenadas R A D I A N E S y F U N C I O N E S, respectivamente.

Bajo estas condiciones la sintaxis para la definición del formato 5 queda expresada como:

```
TITULO[5]= COMANDO GRAFMX
SUBTITULO[5]= FUNCIONES SEN(X) Y COS(X)
EJEX[5]= R A D I A N E S
EJEY[5]= F U N C I O N E S
```

Observaciones:

La extensión máxima para el título y subtítulo es de 50 caracteres.

La extensión máxima para las leyendas de los ejes coordenados es de 20 caracteres.

La asignación nula es válida; ejemplo:

```
EJEY[3]=
```

Por defecto los cinco formatos gráficos se encuentran sin caracteres definidos.

6.3 GRAFICACION DE DATOS MATRICIALES

Para representar los datos almacenados matricialmente en una gráfica, se han implementado dos comandos:

GRAFMX

Grafica todos los elementos de una matriz considerando a la primera columna como los términos independientes (abscisas) y como términos

dependientes (ordenadas) al resto de las columnas. El comando considera que cada columna de la matriz representa funciones distintas. El modo de empleo es:

GRAFMX <matriz> <formato (opcional)>

Ejemplo:

Supóngase que existe una matriz $A^{6,1}$ de orden 31×3 , cuyas columnas están constituidas por:

primera columna).- Secuencia de números equiespaciados dentro del intervalo cerrado $[0, \#PI]$.

segunda columna).- Secuencia de números obtenidos de evaluar la función $SEN(X)$ tomando como variable independiente X a los elementos de la columna 1.

tercera columna).- Secuencia de números obtenidos de evaluar la función $COS(X)$ tomando como variable independiente X a los elementos de la columna 1.

Considere además, para el ejemplo, que el formato de la gráfica será el 5, configurado en el apartado anterior.

Bajo tales circunstancias la sintaxis de la función **GRAFMX** se escribe como:

GRAFMX A 5

obteniéndose el resultado de la figura 1.6.

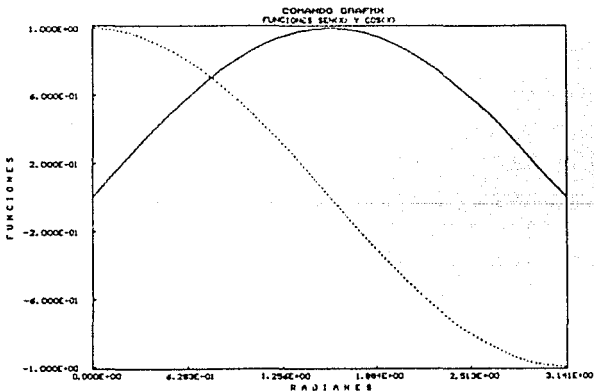


Figura 1.6 Función **GRAFMX**.

6.1 Para generar la matriz **A** basta utilizar secuencialmente los comandos:

```
LENAT A RC R=31 C=3;
@GENC A 1 1 0, #PI/30;
@FUNC A 1 2 X, SEN(X), 1;
@FUNC A 1 3 X, COS(X), 1;
```

GRAFMY

Trabaja de manera análoga al comando anterior, pero considerando a la primera columna de la matriz como los términos dependientes (ordenadas) y a las restantes como abscisas (término independientes). Formato:

`GRAFMY <matriz> <formato (opcional)>`

Ejemplo:

Considerando a la matriz A empleada en `GRAFMX` y suprimiendo el formato (opcional), la sintaxis queda definida como:

`GRAFMY A`

La figura 6.2 muestra la gráfica obtenida.

Nótese que al no ser proporcionado algún número de formato, es colocado como título de la gráfica el nombre de la matriz.

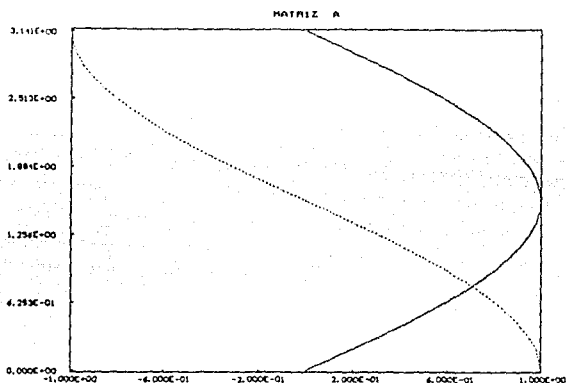


Figura 6.2 Graficación de la matriz A sin formato gráfico.

6.4 GRAFICACION DE EXPRESIONES MATEMATICAS

Se ha mencionado ya la factibilidad de graficar funciones bi o tridimensionalmente. A continuación se explican los comandos orientados a esta tarea:

6.4.1 COMANDO GRAF2D

Este comando permite evaluar y graficar una expresión matemática de una variable independiente. El formato de uso es:

`GRAF2D <F(X)>, <X>, <lim. inf.>, <lim. sup.>, <#Inter.>, <formato (opcional)>`

donde

<F(X)>	expresión matemática a graficar
<X>	variable independiente
<lim. inf.>	límite inferior de la gráfica
<lim. sup.>	límite superior de la gráfica
<#inter.>	número de intervalos en los que se desea graficar
<formato>	número entero que representa el formato destinado a la gráfica

Observaciones:

El número de intervalos sólo está restringido por la cantidad de memoria, pudiéndose con ésto alcanzar gran definición.

El formato gráfico es opcional. Si no es proporcionado, el programa asume como título la expresión matemática y como subtítulo el intervalo de graficación.

Ejemplo:

GRAF2D SEN(X),X,-#PI,#PI,300

Se grafica en 300 puntos la función SEN(X) variando X de -#PI a #PI, sin emplear formato alguno. El resultado del ejemplo se muestra en la figura 6.3.

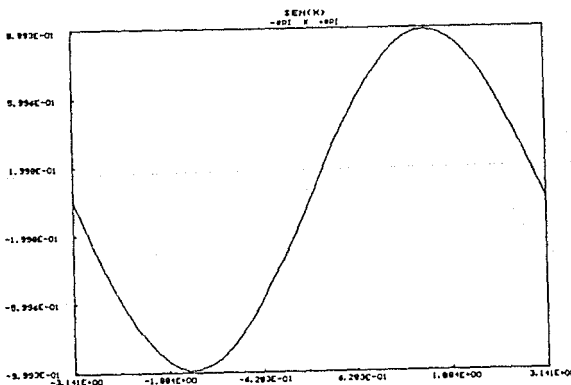


Figura 6.3 Graficación de la función SEN(X) mediante el comando GRAF2D.

$$\text{SEN}(X^2+Y^2)/(X^2+Y^2)$$

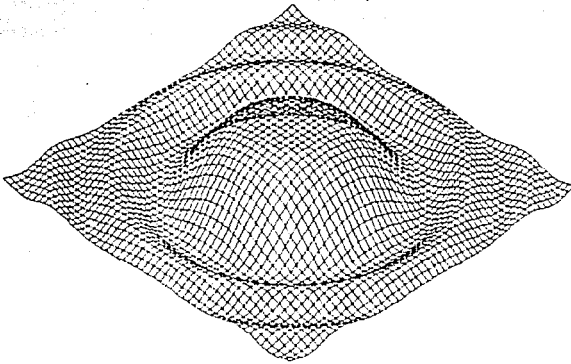


Figura 6.4 Gráfica de la función $\text{SEN}(X^2+Y^2)/(X^2+Y^2)$

Archivos	Editor	Ejecutar	Modo
GRAF3D			
F2: Proyección Isométrica			
F3: Parámetros de la gráfica			
F4: Incremento $\delta\theta$ $\delta\theta= 1^\circ$			
F5: Angulo de rotación $\theta_x= 0^\circ$			
F6: Angulo de rotación $\theta_y= 0^\circ$			
F7: Angulo de rotación $\theta_z= 0^\circ$			
F8: Escala no proporcional			
GRAFLO-90		Mensajes	
Procesador numérico didáctico para estructuras			
MP	Trabajando en función : GRAF3D		

Figura 6.5 Menú para el manejo de gráficas tridimensionales.

F2: Proyección

Cambia el tipo de proyección axonométrica utilizada para visualizar la gráfica. Por defecto se usa la proyección isométrica; sin embargo, al presionar la tecla F2 se cambia a proyección diamétrica. En la figura 6.6 se muestra la diferencia de proyección aplicada al ejemplo.

F3: Parámetros de la gráfica

Proporciona los valores máximos y mínimos de las variables independiente y la función. En la figura 6.7 se muestra el resultado para el ejemplo.

$$\text{SEN}(X^2+Y^2)/(X^2+Y^2)$$

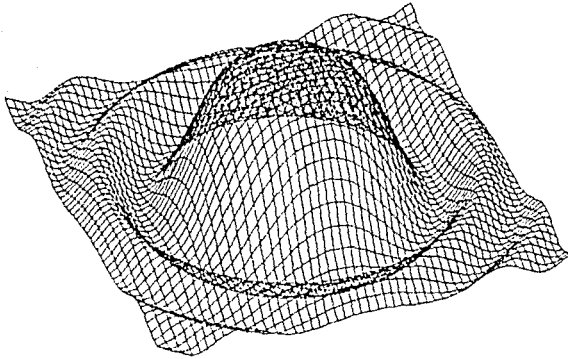


Figura 6.6 Proyección diamétrica de la función $\text{SEN}(X^2+Y^2)/(X^2+Y^2)$.

Archivos	Editor	Ejecutar	Modo
GRAF3D			
F2: Proyección Isométrica			
F3: Parámetros de la gráfica			
Valores típicos de la gráfica			
-3.1415E+00 ≤ X ≤ 3.1415E+00			
-3.1415E+00 ≤ Y ≤ 3.1415E+00			
-2.1691E-01 ≤ f(X,Y) ≤ 1.0000E+00			
Número de intervalos 50			
Número de puntos calculados 2601			
ORFEO-90		Mensajes	
Procesador numérico didáctico para estructuras			
MF	Trabajando en función : GRAF3D		

Figura 6.7 Parámetros de la función $\text{SEN}(X^2+Y^2)/(X^2+Y^2)$.

Rotación de gráficas.

Para poder observar la gráfica bajo distintos ángulos, se han incorporado las funciones:

- F4: Incremento $\delta\theta$
- F5: Angulo de rotación θ_x
- F6: Angulo de rotación θ_y
- F7: Angulo de rotación θ_z

La función F4: aumenta, o disminuye, el ángulo de edición para modificar los valores de θ_x, θ_y y θ_z . Mediante F5:, F6: y F7: se modifican estos valores pulsando las flechas del teclado.

El signo de los ángulos de rotación se rige por el sentido antihorario de un sistema coordenado izquierdo (figura 6.8).

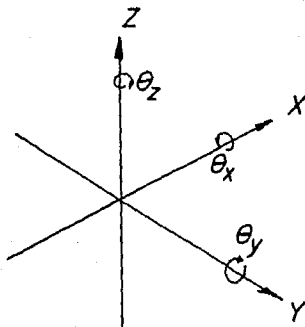


Figura 6.8 Sistema coordenado utilizado por ORFEO-90

Por defecto el valor de $\delta\theta$ es uno y de θ_x , θ_y , y θ_z es cero. En la figura 6.9 se muestra el aspecto del ejemplo considerando un giro sobre el eje Z de 45 grados.

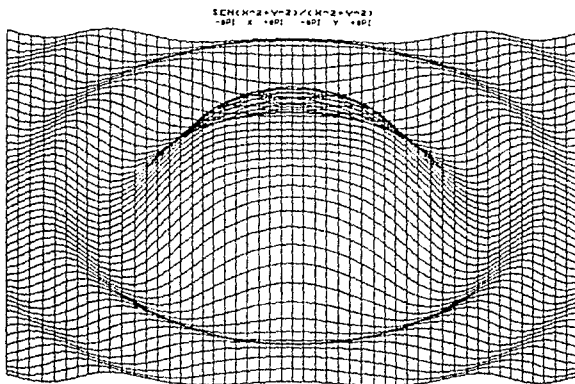


Figura 6.9 Proyección isométrica de la función $\text{SEN}(X^2+Y^2)/(X^2+Y^2)$ rotada 45 grados respecto al eje Z.

F8: Escala

Esta opción permite elegir el tipo de escala utilizada para graficar la función. La escala no proporcional ajusta la figura utilizando el máximo espacio del puerto de visión, pero distorsionando la gráfica. La escala

proporcional elimina este problema, pero en algunas ocasiones dificulta la comprensión de la superficie. Por defecto se utiliza la escala no proporcional. La figura 6.10 muestra el efecto de aplicar escala proporcional a la proyección diamétrica del ejemplo estudiado.

$$\text{SEN}(X^2+Y^2)/(X^2+Y^2)$$

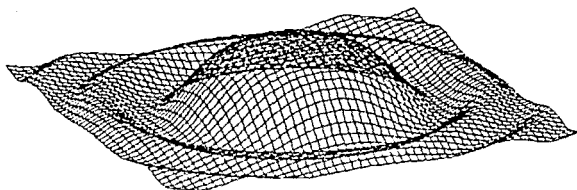


Figura 6.10 Proyección diamétrica de la función $\text{SEN}(X^2+Y^2)/(X^2+Y^2)$ empleando escala proporcional.

MISCELANEA DE FUNCIONES

7

7.1 INICIACION DE VARIABLES

Las veintiseis variables globales de ORFEO-90 se encuentran iniciadas a cero cuando el programa es puesto en operación. Sin embargo, es posible que durante la ejecución de algún procedimiento se requiera iniciar nuevamente las variables a cero. Para tal efecto se cuenta con el comando LIMPIAV. El formato de uso es:

LIMPIAV

7.2 LISTADO DE UN PROGRAMA

Cuando se ha realizado en modo programado una secuencia de sentencias, frecuentemente se desea imprimirlas, ya sea para corregir su estructura, o como respaldo de un programa. La función LISTAP produce un listado por impresora del programa en edición. El formato de uso es:

LISTAP

7.3 COMANDO PAUSA

PAUSA es una función que permite interrumpir la secuencia de operaciones definidas, en el editor programado, que se encuentren ejecutando en ese momento para que el usuario decida si continúa con el proceso o no. El formato de uso es:

PAUSA

Al aplicarse el comando se despliega en la ventana de mensajes la leyenda:

*Para continuar presione cualquier tecla
de lo contrario ESC*

Si la opción es ESC, el programa retorna inmediatamente al menú principal, sin importar en donde se encontraba, ni qué proceso se estaba realizando.

7.4 MEMORIA DISPONIBLE

Para conocer la cantidad de memoria disponible en el sistema, ORFEO-90 cuenta con dos funciones:

MEMC

Proporciona la cantidad de memoria disponible para definir matrices por encima de la última gestión. El formato de uso es:

MEMC

La cantidad de memoria disponible es retornada a la ventana de mensajes. Al comenzar la operación de ORFEO-90, se proporcionan aproximadamente 41 Kb de memoria para asignación de matrices.

MEML

Proporciona la cantidad de memoria disponible en el sistema para funciones internas. Aunque no es posible establecer con exactitud la cantidad de memoria libre necesaria para las operaciones internas del programa, es importante revisar este parámetro cuando se utiliza ORFEO-90 junto con programas residentes en memoria, como los impresores de pantallas gráficas. Se estima que al menos debe haber unos 40 Kb libres para satisfacer la demanda interna de memoria. El formato de uso es:

MEML

Al igual que en la función anterior, la cantidad de bytes libres es reportada en la ventana de mensajes.

7.5 LISTADO DEL DIRECTORIO ACTIVO

La función LISTAD crea una ventana temporal dentro del editor programado (figura 7.1) con los archivos del directorio activo. El formato de uso es:

LISTAD <extensión (opcional)>

Archivos	Editor	Ejecutar	Modo	
Directorio activo:				
A:\				
E-01.PRG	E-02.PRG	E-03.PRG	E-04.PRG	E-05.PRG
E-06.PRG	E-07.PRG	E-08.PRG	E-09.PRG	

Editor interactivo		Mensajes
Comando:	LISTAD PRG	
MI	Trabajando en función : LISTAD	

Figura 7.1 Ventana temporal del comando LISTAD

Ejemplos:

LISTAD

Al suprimirse la extensión se produce una lista de todos los archivos del directorio activo.

LISTAD PRG

Proporciona todos los archivos con extensión PRG del directorio activo.

NOTA: Es válido el uso de comodines como PR* ó M*.

PROGRAMAS DE APOYO

8.1 INSTALACION

En el primer capítulo se mencionó que ORFEO-90 puede ser usado en discos flexibles o en disco rígido, siempre y cuando se instale en forma adecuada.

Debido a la alta compatibilidad perseguida en la elaboración del programa, la instalación del mismo adquiere cierto grado de complejidad. Para evitar al usuario del sistema una larga lista de condiciones a cumplir en el copiado de archivos, el paquete ORFEO-90 cuenta con un instalador de fácil manejo. El programa que cumple este objetivo se llama `INSTALAR.EXE` y se encuentra en el disco de `UTILERIAS`.

IMPORTANTE:

Antes de realizar cualquier operación de instalado asegúrese de hacer una copia de trabajo de los discos del paquete ORFEO-90, sobre todo si desea instalar el programa para trabajar en discos flexibles, ya que el instalador `MODIFICA` los discos.

8.1.1 DISCOS FLEXIBLES

Para instalar el programa para discos flexibles se deben seguir los pasos indicados a continuación:

- 1.- Realizar una copia de los dos discos del paquete ORFEO-90, SISTEMA y UTILERIAS, debidamente etiquetada.
- 2.- Insertar el disco COPIA de UTILERIAS en cualquier unidad de lectura de disco flexible (drive) activa.
- 3.- Teclar:

`INSTALAR FLEXIBLE`

Al presionar <return> el programa determinará el tipo de configuración de su computadora tomando las medidas pertinentes para una correcta operación de ORFEO-90. Inmediatamente después se desplegará el mensaje:

Inserte LA COPIA del disco SISTEMA de ORFEO-90 en la unidad que está utilizando para instalación y presione cualquier tecla...

Saque el disco de UTILERIAS, replácelo por la COPIA del disco SISTEMA, y presione <return>.

El programa configurará entonces al sistema para el tipo de computadora y monitor que esté empleando. Si piensa utilizarlo en alguna otra máquina, es posible que la configuración de las pantallas, en particular, no sea de su completo agrado. Si este es el caso, utilice el programa `CNFORFEO.COM` incluido en el disco SISTEMA. Esta utilería reemplaza la configuración de video por la que usted escoja. Posteriormente, dentro de este capítulo, se tratará en detalle este programa.

8.1.2 DISCO RIGIDO

Aun y cuando la instalación sobre disco rígido no modifica los discos del paquete, se recomienda efectuar una copia de respaldo. Los pasos de instalación son:

- 1.- Insertar el disco COPIA de UTILERIAS en cualquier unidad de disco flexible (drive) activa.
- 2.- Teclar:

INSTALAR RIGIDO

Al presionar <return>, el programa de instalación solicitará la unidad de disco rígido en la cual se desea almacenar, y configurar, ORFEO-90. Las unidades válidas van desde C a Z. Para seleccionar el disco rígido deseado, presione la letra correspondiente a él.

Inmediatamente el programa procederá a generar el directorio ORFEO-90. Por esta razón es importante que NO exista un directorio con este nombre, ya que de ser así el programa reportará el mensaje:

Error: No es posible crear el directorio ORFEO-90.
Consulte el manual del usuario.

Este mismo mensaje es desplegado si la unidad de disco rígido es inexistente o inaccesible.

Una vez creado el directorio, el programa pide que se remplace el disco UTILERIAS por el disco SISTEMA.

Al insertar el nuevo disco el programa procede a copiar ORFEO-90.EXE junto con todo el sistema de entorno, configurándolo de acuerdo con el tipo de computadora que se halla detectado.

Es posible que algunos de los programas de apoyo no sean copiados. Esto se debe a que no en todos los sistemas pueden correr estas utilerías. La incompatibilidad reside principalmente en el tipo de tarjeta gráfica adaptadora que se tenga. La tabla 8.1 muestra cuáles son los archivos suprimidos del sistema, atendiendo al tipo de monitor detectado.

TARJETA	UTILERIAS SUPRIMIDAS
CGA	NINGUNA
HCGA	NUNUNA
HERCULES	ANIMA.EXE, SISMO.EXE y *.BIN
EGA	GRAFICAR.EXE
VGA	GRAFICAR.EXE

Tabla 8.1

Además de las condiciones mencionadas, se aclara que sólo es cargado el manejador de video necesario, ahorrándose con ésto memoria.

El procedimiento de instalación para disco rígido va solicitando el intercambio de discos, detectándose cualquier error.

8.2 CONFIGURADOR CNFORFEO.COM

Esta utilería permite cambiar los atributos de video (colores) utilizados en las pantallas del programa. Para su operación, inserte el disco COPIA del SISTEMA en la unidad de disco flexible activa, si instaló el programa para disco flexible, o bien posicione en el directorio ORFEO-90 para disco rígido. Escriba:

CNFORFEO

Al presionar <return> aparecerá el mensaje:

Seleccione la configuración deseada con las teclas Fn o Ctrl-Fn

Las teclas que pueden utilizarse van de F1 a F10 o de Ctrl-F1 a Ctrl-F10. Esto permite tener 20 distintas configuraciones.

Observaciones:

En el caso de contar con monitor monocromático el programa CNFORFEO.COM sólo permitirá una configuración.

Las combinaciones Ctrl-Fn suprimen el efecto tridimensional de las ventanas de ORFEO-90.

Por defecto la configuración adoptada para monitores a color es la F6.

8.3 PROTECCION Y DESPROTECCION DE ARCHIVOS

ORFEO-90 cuenta con dos programas orientados a la creación de bibliotecas de matrices protegidas, PLIB.COM y DLIB.COM. El término protegido se aplica porque bajo ningún comando de borrado o sobrescritura del sistema operativo convencional, es posible eliminar los archivos. Los protectores se encuentran en el disco UTILERIAS. Para ejemplificar el manejo de estos programas considere que se tiene una matriz de rigideces algebraica (o de cualquier tipo) de nombre KASG grabada en disco, y que por algún motivo desea protegerla. El procedimiento es simple. Escriba:

PLIB KASG

y presione <return>. Inmediatamente aparecerá el mensaje:

Archivo
KASG.MAT
protegido

Nótese que no es necesario proporcionar la extensión .MAT del archivo que almacena la matriz.

IMPORTANTE: Tanto PLIB.COM como la matriz a proteger deben encontrarse en el mismo directorio.

Si ahora desea borrar el archivo KASG.MAT mediante el comando DEL del sistema operativo, se desplegará el mensaje "acceso denegado".

Para invertir el proceso de protección se cuenta con el programa DLIB.COM. Para aplicarlo basta escribir, de acuerdo con el ejemplo:

DLIB KASG

Con lo cual recibirá el mensaje:

Archivo
KASG.MAT
desprotegido

Ambos programas de apoyo pueden ser utilizados para proteger o desproteger cualquier tipo de archivo. Para ello sólo es necesario proporcionar la extensión. Ejemplo:

PLIB ORFEO.CNF

8.4 PROCESAMIENTO DE GRAFICAS GRABADAS EN DISCO

Cuando se han grabado gráficas en disco mediante las teclas calientes Alt-Fn (ver sección 6.1), el programa GRAFICAR.EXE puede recuperarlas para impresión, o simplemente para ser observadas. Para tal efecto, sólo es necesario escribir:

GRAFICAR ORFEO n

en donde n indica el número de gráfica a procesar.

Ejemplo:

Supóngase que durante la operación de ORFEO-90, se utilizó en una representación gráfica las teclas calientes Alt-F4. Bajo estas condiciones se creó el archivo ORFEO4.GRF. Para recuperar la grafica con la utilería en cuestión, basta escribir:

GRAFICAR ORFEO4

Observaciones:

El nombre del archivo que contiene a la gráfica debe ser proporcionado SIN extensión.

Tanto GRAFICAR.EXE como ORFEO n .GRF deben estar en el mismo directorio.

GRAFICAR.EXE sólo es compatible con monitores CGA, MCGA y MDA (Hércules) y con impresoras EPSON serie MX o equivalentes.

GRAFICAR.EXE se encuentra en el disco de UTILERIAS.

8.5 ANIMACION

Como apoyo a la dinámica estructural ORFEO-90 cuenta con dos animadores de respuesta para sistemas de un grado de libertad.

8.5.1 ANIMA.EXE

Este programa, situado en el disco UTILERIAS, permite animar la respuesta de sistemas de un grado de libertad calculadas en ORFEO-90. Para operarlo se utiliza la siguiente sintaxis:

ANIMA <matriz> <salto> <color>

donde

- <matriz> Es el arreglo que contiene la respuesta que se desea animar previamente grabado en disco mediante el comando SALVAR de ORFEO-90.
- <salto> Indica cada cuantos puntos de la respuesta se quiere un cuadro de animación. Si se proporciona 1, la animación es muy continua, pero puede resultar lenta si no se cuenta con coprocesador matemático. Por otra parte, si se toma un valor alto de <salto>, se incrementa la velocidad de animación pero pueden presentarse brinco discontinuos por perderse "picos" importantes en la respuesta. Es tarea del usuario calibrar la animación de acuerdo con el intervalo de tiempo de la respuesta. Un buen margen de inicio es tomar un valor de <salto> de 3 - 8.
- <color> Número entero que representa la gama de colores que se desea tener. Los valores válidos son 1 ó 0. El primero produce los colores negro, blanco, azul y magenta; y el segundo negro, rojo, amarillo y verde.

Para imprimir mayor claridad en la explicación, considere el siguiente ejemplo:

Suponga que se tienen dos matrices, R1 y R2, que contienen las respuestas de uno y dos sistemas de un grado de libertad respectivamente.

Las respuesta consideradas son:

Matriz	Orden	Sistema 1	Sistema 2
R1	R x 2	Sen(t)	****
R2	R x 3	Sen(t)	Cos(t)

en donde t varía de 0 a 9.90 radianes. La figura 8.1 muestra la codificación de estas respuestas, y las figuras 8.2 y 8.3 la representación gráfica de las mismas.

Una vez grabadas las matrices R1 y R2 en disco es posible animar su contenido:

Matriz R1

ANIMA R1 2 1

Al presionar <return> comenzará el proceso de carga de la matriz R1 y la configuración del sistema gráfico. Cuando aparezca el mensaje:

Para iniciar la animación presione cualquier tecla...

el programa se encuentra listo para comenzar.

Archivos	Editor	Ejecutar	Modo
*** CODIFICACION PARA OBTENER LAS MATRICES DE RESPUESTA R1 Y R2 ***			
<pre> LEMAT R1 RC R=100 C=2; EGENC R1 1 1 0,0,1; EFUNC R1 1 2 X,SEN(X),1; LEMAT R2 RC R=100 C=3; EGENC R2 1 1 0,0,1; EFUNC R2 1 2 X,SEN(X),1; EFUNC R2 1 3 X,COS(X),1; SALVA R1; SALVA R2; </pre>			
ORFEO-90 Procesador numerico didactico para estructuras		Mensajes	
MP	F1: Ayuda	F10: Menu principal	

Figura 8.2

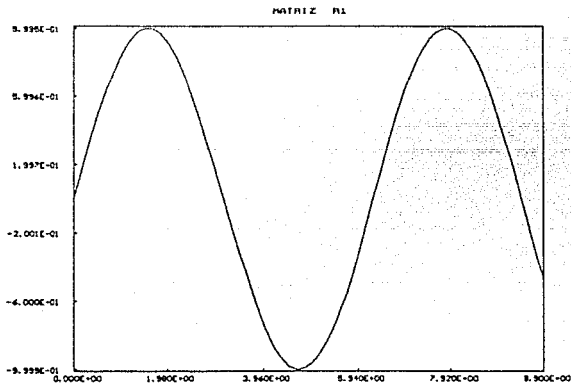


Figura 8.3

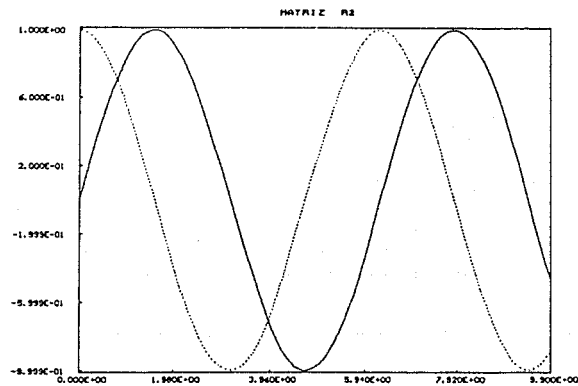


Figura 8.3

De acuerdo con la sintaxis empleada, la animación se llevará a cabo cada dos puntos de la respuesta, es decir; para los tiempos 0.0, 0.02, 0.04...etc, utilizando la gama de colores 1 (negro, blanco, azul y magenta). La figura 8.4 muestra el resultado obtenido.

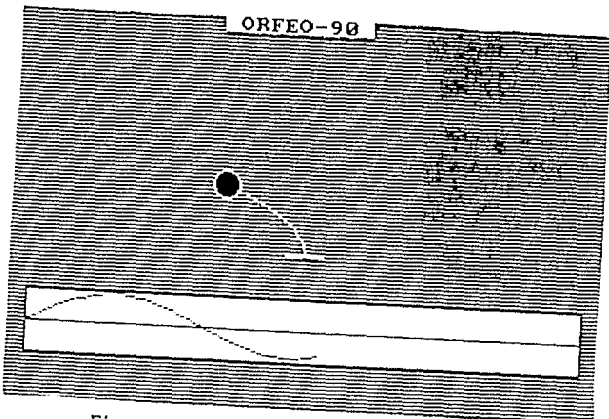


Figura 8.4 Animación de la matriz R1.

Como se puede observar, el programa ANIMA.EXE proporciona la gráfica de la respuesta en la parte inferior de la pantalla.

Matriz R2

ANIMA R2 2 1

Ahora, con los mismos atributos de <salto> y <color>, se anima comparativamente la respuesta de dos sistemas de un grado de libertad. El resultado se muestra en la Figura 8.5.

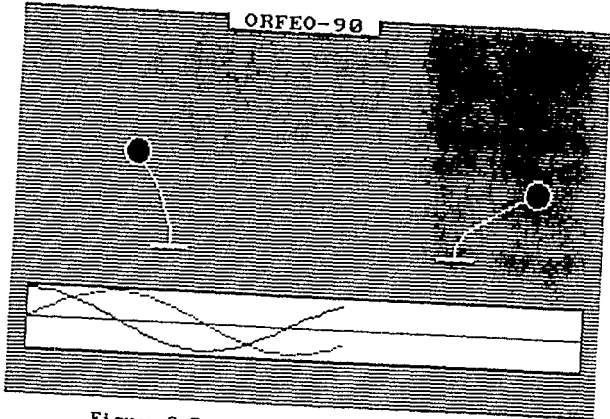


Figura 8.5 Animación de la matriz R2.

El código de color produce que la respuesta del primer sistema, situado a la izquierda de la pantalla, sea graficada en color azul, mientras que en el segundo la respuesta es mostrada en color blanco. Si el código hubiera sido 0, la gráfica presentaría, para el lado izquierdo el color verde y para el derecho amarillo.

Observaciones:

El programa ANIMA.EXE y la matriz a animar deben encontrarse en el mismo directorio.

8.5.2 SISMO.EXE

Para visualizar el efecto producido en sistemas de un grado de libertad por efecto de un acelerograma, se cuenta con el programa SISMO.EXE del disco UTILERIAS.

El modo de operación es el siguiente:

SISMO <señal> <#ptos> <dt> <T1> <T2> <X1> <X2> <salto> <ind> <color>

donde

<señal> Es el acelerograma de excitación en FORMATO BINARIO.

<#ptos> Número de puntos del acelerograma.

<dt> Intervalo de tiempo en el que está digitizada la señal.

<T1> Periodo del primer sistema de un grado de libertad, lado izquierdo.

<T2> Periodo del segundo sistema. Lado derecho.

<X1> <X2> Amortiguamiento para los sistemas izquierdo y derecho, respectivamente, expresado como fracción del amortiguamiento crítico.

<salto> Indica cada cuantos puntos se desea un cuadro de animación.

<ind> Tipo de gráfica obtenida en la parte inferior de la pantalla:

- 1 Sólo acelerograma.
- 2 Sólo respuestas de los péndulos.
- 3 Sismo y respuestas.

<color> Código de color:

0	1	
Rojo	Magenta	Acelerograma.
Verde	Azul	Respuesta péndulo izquierdo.
Amarillo	Blanco	Respuesta péndulo derecho.

Para estudiar los efectos sísmicos en este tipo de sistemas, se proporcionan en el disco UTILERIAS los archivos SD.BIN y SB.BIN que corresponden a dos sismos con las siguientes características:

SD.BIN

Registro típico de suelo duro. (figura 8.6)
Número de puntos de la señal 1209
Intervalo de muestreo dt 0.02 seg
Unidades gals^{8.1}

SB.BIN

Registro típico de suelo blando. (figura 8.7)
Número de puntos de la señal 1947
Intervalo de muestreo dt 0.02 seg
Unidades gals

Así mismo, como apoyo se proporcionan los espectros de respuesta para estos sismos considerando 5 por ciento del amortiguamiento crítico (figuras 8.8, 8.9 y tabla 8.2).

Ejemplo:

Tomemos el registro de zona dura, SD. De acuerdo con el espectro de la figura 8.8, se tienen altas respuestas para periodos cortos. Para comparar visualmente esta situación, se puede animar esta respuesta de la siguiente manera:

SISMO SD.BIN 1200 0.02 0.5 2.0 0.05 0.05 4 2 1

Donde se ha considerado que el primer sistema, situado en el lado izquierdo de la pantalla, tiene un periodo $\langle T1 \rangle = 0.5$ seg (alta respuesta), y el segundo, un periodo $\langle T2 \rangle = 2.0$ seg (baja respuesta). Para ambos sistemas, el amortiguamiento considerado es del 5 por ciento del crítico.

De acuerdo con la sintaxis se animarán los péndulos cada 4 intervalos (0.08 seg), y la gráfica reportada será la correspondiente a las respuestas $\langle ind \rangle = 2$. En la figura 8.10 se muestra el resultado obtenido.

Observaciones:

El registro y el programa SISMO.EXE deben estar en el mismo directorio.

Al proporcionar el nombre del archivo que contiene al acelerograma, se debe escribir la extensión.

8.5.3 BINARIO.EXE

Se ha mencionado en el punto anterior que los acelerogramas empleados en el programa SISMO.EXE deben estar dados en binario. Esto se debe a que por lo general los sismos son almacenados de esta manera, para optimar

^{8.1} 1 gal = 1 cm / seg²

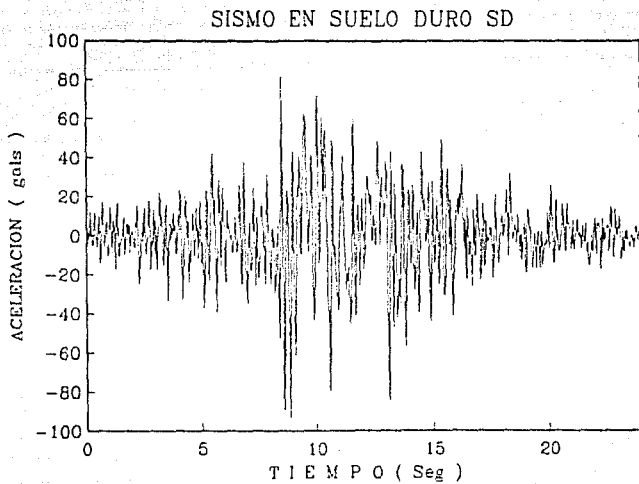


Figura 8.6

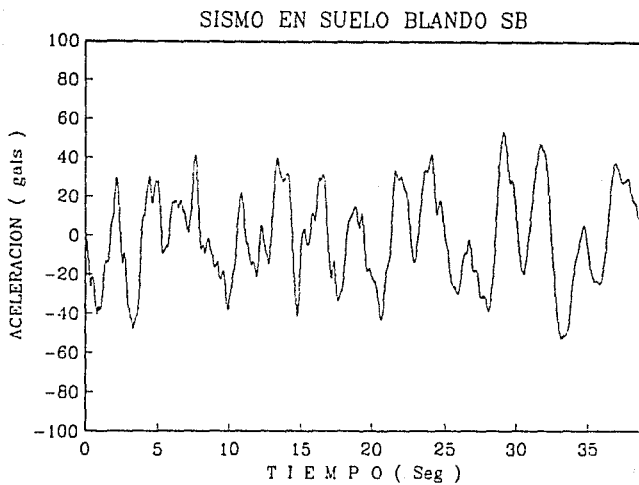


Figura 8.7

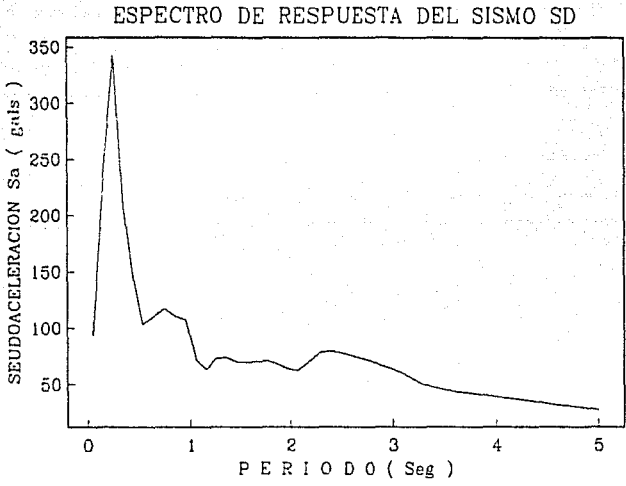


Figura 8.8

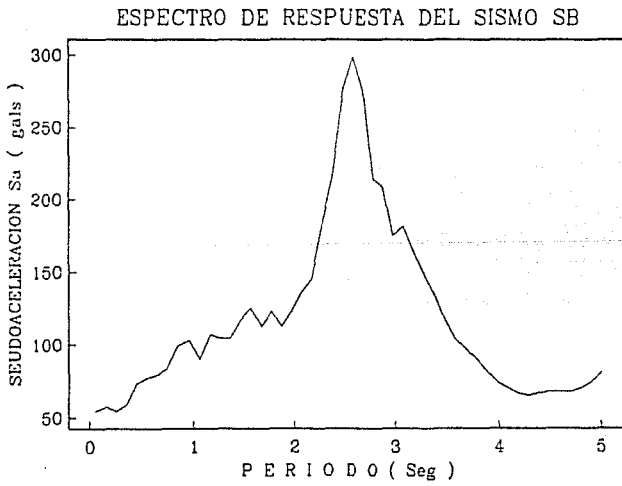


Figura 8.9

Periodo (seg)	Suelo duro (gals)	Suelo blando (gals)
0.05	9.357	5.427
0.15	24.376	5.738
0.25	34.311	5.445
0.35	20.816	5.942
0.45	14.430	7.366
0.56	10.322	7.706
0.66	11.056	7.892
0.76	11.748	8.338
0.86	11.057	9.938
0.96	10.776	10.310
1.06	7.176	9.029
1.16	6.331	10.738
1.26	7.311	10.471
1.36	7.369	10.437
1.46	7.015	11.727
1.57	6.972	12.517
1.67	7.070	11.257
1.77	7.079	12.302
1.87	6.805	11.262
1.97	6.416	12.249
2.07	6.256	13.593
2.17	6.982	14.539
2.27	7.815	18.127
2.37	7.978	21.603
2.47	7.794	27.672
2.58	7.548	29.881
2.68	7.342	27.517
2.78	7.080	21.426
2.88	6.699	20.899
2.98	6.427	17.472
3.08	6.075	18.179
3.18	5.616	16.444
3.28	5.062	14.955
3.38	4.814	13.505
3.48	4.574	11.713
3.59	4.361	10.397
3.69	4.258	9.696
3.79	4.154	8.966
3.89	4.050	8.088
3.99	3.941	7.413
4.09	3.826	7.016
4.19	3.705	6.633
4.29	3.576	6.493
4.39	3.440	6.692
4.49	3.303	6.795
4.60	3.193	6.788
4.70	3.091	6.798
4.80	2.985	6.981
4.90	2.874	7.414
5.00	2.758	8.061

Tabla 8.2 Valores de los espectros de respuesta.

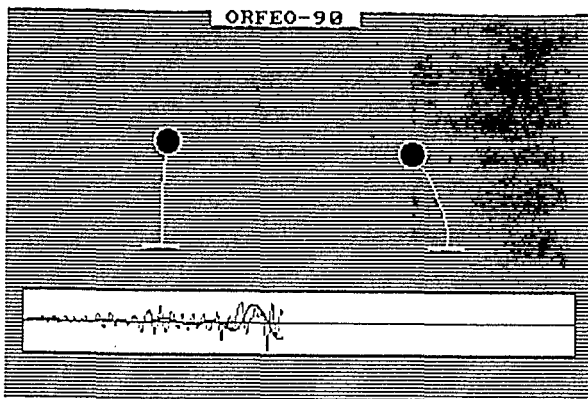


Figura 8.10

memoria. Si no cuenta con señales grabadas binariamente, es posible resolver la incompatibilidad con SISMO.EXE mediante el programa de conversión BINARIO.EXE. El modo de empleo es:

`BINARIO <archivo formato ASCII> <#ptos> <archivo formato binario>`

En donde:

- | | |
|--|--|
| <code><archivo formato ASCII></code> | Archivo compuesto por un vector columna con los valores de las aceleraciones en código ASCII (caracteres). |
| <code><#ptos></code> | Número de puntos del archivo en formato ASCII que se desea sean almacenados en formato binario. |
| <code><archivo formato binario></code> | Archivo de salida. Contiene los valores almacenados en formato binario. |

Ejemplo:

Supóngase que se tiene un archivo llamado SISMO1.DAT con 750 aceleraciones dispuestos en la forma:

```
0.0128
0.0259
0.0295
```

```
⋮
⋮
```

Para cambiar el formato a binario y almacenarlo en el nuevo archivo SISMO1.BIN basta con aplicar:

BINARIO SISMO1.DAT 750 SISMO1.BIN

Observaciones:

BINARIO.EXE y el archivo a convertir deben encontrarse en el mismo directorio.

El número de datos a transferir en binario puede ser menor al número de datos del archivo ASCII, pero nunca mayor.

**PROGRAMAS DE EJEMPLO
PARA EL MANEJO DE
ORFEO-90**

A

En el disco de UTILERIAS se encuentran nueve programas con ejemplos acerca del manejo de ORFEO-90. A continuación se muestra el directorio de los mismos junto con una pequeña explicación del tópico que tratan.

PROGRAMA	T O P I C O
E-01.PRG	Asignación de variables, funciones matemáticas y comandos relacionados.
E-02.PRG	Funciones de entrada y salida para matrices.
E-03.PRG	Modificación y generación de datos almacenados en matrices.
E-04.PRG	Aritmética de matrices.
E-05.PRG	Solución de sistemas de ecuaciones lineales.
E-06.PRG	Vectores y valores característicos. Función JACOBI.
E-07.PRG	Análisis dinámico modal. Función @DIN y comandos relacionados.
E-08.PRG	Análisis paso a paso. Función @PASO y comandos relacionados.
E-09.PRG	Manejo de gráficas.

NOTA: Los listados mostrados a continuación carecen de acentos por encontrarse escritos en el procesador de textos de ORFEO-90, el cual no los soporta.

```

*****
* ORFEO-90 PROCESADOR NUMERICO DIDACTICO PARA ESTRUCTURAS *
* Ver 1,5 *
* E-01 *
* *
* Objetivo: Mostrar al usuario del paquete el uso de las *
* funciones matematicas y comandos relacionados. *
*****

```

*** FUNCIONES Y CONSTANTES MATEMATICAS ***

Constantes:

#E - Proporciona el valor del numero e = 2.718281828459...
 #PI - Proporciona el valor del numero π = 3.141592653589...

Funciones:

ABS(n) - Proporciona el valor absoluto de n.
 ACOS(n) - Calcula el angulo cuyo coseno es n.
 ALN(n) - Calcula el antilogaritmo Neperiano de n.
 ALOG(n) - Calcula el antilogaritmo (base 10) de n.
 ASEN(n) - Calcula el angulo cuyo seno es n.
 ATAN(n) - Calcula el angulo cuya tangente es n.
 COS(n) - Calcula el coseno de n.
 FAC(n) - Calcula el factorial de n.
 LN(n) - Calcula el logaritmo Neperiano de n.
 LOG(n) - Calcula el logaritmo (base 10) de n.
 MAXV(V) - Proporciona el maximo valor absoluto de un vector
 columna V. La funcion solo acepta arreglos de tipo
 R (real). En el caso de que V tenga varias columnas
 (matriz), el comando opera sobre la primera.
 SEN(n) - Calcula el seno de n.
 TAN(n) - Calcula la tangente de n.

Funciones @:

@SUM(V) - Calcula la suma de los valores del vector V. Todas
 las observaciones hechas a MAXV se aplican a esta
 funcion.
 @E(M,i,j) - Proporciona el elemento i,j del arreglo M. La funcion
 acepta todo tipo de matrices exceptuando a las enteras
 @INT(EXPRESION,VARIABLE,INFERIOR,SUPERIOR,M,TIPO)
 - Realiza la integracion numerica de la funcion EXPRESION
 con respecto a VARIABLE de INFERIOR a SUPERIOR con el
 metodo definido por TIPO en M intervalos.

TIPO:

S2 - Regla de Simpson 1/2 <=> M - cualquiera.
 S3 - Regla de Simpson 1/3 <=> M - numero par.
 S8 - Regla de Simpson 3/8 <=> M - multiplo de tres.
 G1 - Cuadratura de Gauss-Legendre grado 1.
 G2 - Cuadratura de Gauss-Legendre grado 2.
 G3 - Cuadratura de Gauss-Legendre grado 3.
 G4 - Cuadratura de Gauss-Legendre grado 4.
 G5 - Cuadratura de Gauss-Legendre grado 5.

Aunque no se necesita el valor de M en la integracion de
 Gauss-Legendre, se debe proporcionar un numero de intervalos
 M>3 por motivos de sintaxis. En todos los casos se revisa que
 el valor de M sea valido.

Para efectuar integrales dobles o triples, se pueden anidar
 las funciones (@INT(@INT(@INT(...))...))), sin importar
 que se utilicen diferentes metodos de integracion para cada
 una de ellas.

*** FUNCIONES RELACIONADAS ***

Funcion: LISTAV
Formato: LISTAV;

Funcion: LISTAVI
Formato: LISTAVI;

EJEMPLOS:"

A=#E; Z=0.0; "<= Asignaciones sencillas"

B=C=D=#PI; "<= Asignacion multiple"

" Aplicación de las funciones..."

X=ABS(ACOS(1)-ALN(Z)-ALOG(Z)-ASEN(Z)-ATAN(Z)+COS(B/Z)-FAC(1)+LN(A)
+LOG(10)+SEN(Z)+TAN(D/4)+1); "El resultado es X=1.00"

" Se listan las variables:" LISTAV;

" Aplicacion de las funciones @..."

Se cargan las matrices A y B algebraica de orden 1x1 y real de 1x5
respectivamente para mostrar el uso de las funciones restantes."

LEMAT A A 0=1; X^2+X+1; IMPMAT A; "<= Se muestra la matriz A."

LEMAT B R R=5 C=1;

X; "<= X vale 1.00 por los calculos anteriores."
2.00;
3.00;
4.00;
5.00; IMPMAT B; "<= Se muestra la matriz B."

Se asigna en la variable A el valor de X^2+X+1; para el valor que en ese
momento contiene la variable X (1.00);"

A=@E(A,1,1); "X^2+X+1=3" LISTAV;

" Se almacena en la variable B el máximo valor absoluto del vector B..."

B=MAXV(B); "B=5.00" LISTAV;

" Se almacena en C la suma de todos los elementos del vector B..."

C=@SUM(B); "C=1+2+3+4+5=15;" LISTAV;

" *** Ejemplos de integracion numerica *** "

I=@INT(X^2+X+1,X,0,10,24,S3); LISTAV;

"Se integra la funcion X^2+X+1 con respecto a X de 0 a 10 utilizando el
criterio de Simpson 3/8 en 24 intervalos."

I=@INT(@INT(X^2+Y^2,X,0,10,24,S3),Y,0,5,4,G5); LISTAV;

"Se realiza la integral doble de la expresion X^2+Y^2 sobre la region
X(0,10) y Y(0,5). Note que se pueden emplear diferente metodos de
integracion para cada sentido."

" NOTA: Si posee impresora conectada y en servicio, substituya el comando
LISTAV por LISTAVI y obtendra la salida de datos por este dispo-
sitivo."

```

*****
* DRFEO-90 PROCESADOR NUMERICO DIDACTICO PARA ESTRUCTURAS *
* Ver 1.5 *
* E-02 *
* *
* Objetivo: Mostrar al usuario del paquete las funciones *
* de entrada y salida para matrices. *
*****

```

*** FUNCIONES DE ENTRADA ***

Funcion: LEMAT
 Formato: LEMAT <nombre> <tipo> R=<# de renglones> C=<# de columnas>;

- <nombre> - Cualquiera con extension maxima de 4 caracteres
 <tipo>
- R Matriz de reales
 - E Matriz de enteros
 - S Matriz simetrica de reales
 - D Matriz diagonal de reales
 - SD Matriz pseudodiagonal de reales
 - B Matriz en banda de reales
 - A Matriz simetrica algebraica
 (numeros, variables y expresiones)
 - RC Matriz de reales inicializada a cero
 automaticamente.
 - BC Matriz en banda de reales inicializada a cero
 automaticamente.
 - DI Matriz identidad de numeros reales inicializada
 automaticamente.

D=<orden de la matriz> opcional en lugar de R=<...> y C=<...>

*** FUNCIONES DE SALIDA ***

Funcion: LISTAM
 Formato: LISTAM;

Funcion: LISTAMI
 Formato: LISTAMI;

Funcion: IMPMAT
 Formato: IMPMAT <nombre de la matriz>;

Funcion: IMPMATI
 Formato: IMPMATI <nombre de la matriz>;

EJEMPLOS:

"Se carga una matriz M1 de numeros reales de orden 3x3..."

LEMAT M1 R O=3; "=< se pudo escribir LEMAT M1 R R=3 C=3;"

```

1; 2; 3;
4; 5; 6; "=< Datos por renglones"
A=B=COS(0)+6; 15-A; 2*B; "=< Se pueden utilizar expresiones, variables
o asignaciones multiples"

```

"Se carga una matriz M2 de numeros enteros de orden 3x2..."

```

LEMAT M2 E R=3 C=2;
1; 1;
2; 2; "=< Datos por renglones"
3; 3;

```

"Se carga una matriz M3 simetrica de numeros reales de orden 3x3..."

```

LEMAT M3 S O=3;
COS(0); 1; 1;
2; 2; "=< Solo se deben proporcionar los elementos"
3; 3; " requeridos."

```


"Se carga una matriz M4 diagonal de numeros reales de orden 5x5..."

LEMAT M4 D O=5;

1; 2; 3; 4; 5; "<= Solo se deben proporcionar los elementos"
" de la diagonal principal."

"Se carga una matriz M5seudodiagonal de numeros reales de orden 5x5..."

LEMAT M5 SD O=5;

1; 2; 3; 4; 5; "<= Elementos de la diagonal principal"
-1; "<= Valor de los elementos fuera de la diagonal principal"

"Se carga una matriz M6 en banda de 3x2 en donde 3 es el numero de ecuaciones y 2 es el ancho de banda (o semibanda MB)..."

LEMAT M6 B R=3 C=2;

1; 2; " | MB | ancho de banda"
" 1.0 2.0 0.0"
1; 2; "<= Matriz cargada =>" " 0.0 1.0 2.0"
2; 0; " 0.0 0.0 2.0"

"Se carga una matriz M7 algebraica simetrica (matriz de rigideces) de orden 2x2..."

LEMAT M7 A O=2;

4*E*1/L; 2*E*1/L; "<= Solo los elementos requeridos"
4*E*1/L;

"Se carga una matriz M8 de numeros reales inicializada a cero de orden 5x3..."

LEMAT M8 RC R=5 C=3; "No se requieren datos puesto que se inicializan los valores del arreglo a cero de manera automatica."

"Se carga una matriz M9 en banda inicializada a cero de orden 5x3..."

LEMAT M9 BC R=5 C=3;

"Se carga una matriz identidad (diagonal) M10 de orden 4x4..."

LEMAT M10 DI O=5; "No se requieren datos. Generacion automatica."

"Se realiza un listado de las matrices definidas..."

LISTAM;

"Se imprimen las matrices anteriormente definidas en pantalla..."

IMP MAT M1; IMP MAT M2; IMP MAT M3;
IMP MAT M4; IMP MAT M5; IMP MAT M6;
IMP MAT M7; IMP MAT M8; IMP MAT M9;
IMP MAT M10;

"Si posee impresora conectada y en servicio, sustituya las funciones IMPMAT y LISTAM por IMPMATI y LISTAMI respectivamente para obtener la salida por impresora."

```

*****
* ORFEO-90 PROCESADOR NUMERICO DIDACTICO PARA ESTRUCTURAS *
*                               Ver 1.5                               *
* E-03                                                                    *
*                                                                    *
* Objetivo: Mostrar al usuario del paquete las funciones            *
*           para modificacion y generacion de datos en              *
*           matrices.                                                *
*****

```

*** FUNCIONES DE MODIFICACION ***

```

Funcion: MODIF
Formato: MODIF <nombre> <renglon> <columna> <expresion>;

Funcion: @GENC
Formato: @GENC <nombre> <renglon> <columna> <inicio>,<intervalo>;

Funcion: @GENR
Formato: @GENR <nombre> <renglon> <columna> <inicio>,<intervalo>;

Funcion: @FUNC
Formato: @FUNC <nombre> <renglon> <columna> <variable>,<expresion>,<ren x>;

Funcion: @FUNR
Formato: @FUNR <nombre> <renglon> <columna> <variable>,<expresion>,<col x>;

```

EJEMPLOS:

"Se cargan la matrices reales M1 y M2 de orden 10x4 y 4x10 respectivamente para los ejemplos..."

```

LEMAT M1 RC R=10 C=4;
LEMAT M2 RC R=4 C=10; "Matrices inicializadas a cero"

```

"Se genera en la columna 1, a partir del renglon 3 de la matriz M1, una secuencia corrida de numeros reales comenzando en 1.0 y con intervalos de 2.0..."

```
@GENC M1 3 1 1.0,2.0; "Resultado:" IMPMAT M1;
```

"Se genera en el renglon 1, a partir de la columna 2 de la matriz M2, una secuencia de numeros reales comenzando en cero e incrementandose en $\cos(2*\pi)$..."

```
@GENR M2 1 2 0,COS(2*PI); "Resultado:" IMPMAT M2;
```

"Se calculan las funciones $\text{SEN}(X)$ $\text{COS}(T)$ y $\text{LN}(Y+1)$ con los datos de la columna 1 de la matriz M1, colocando el resultado en las columnas 2, 3 y 4 respectivamente..."

```
@FUNC M1 1 2 X,SEN(X),1; "Funcion SEN(X)..."
@FUNC M1 1 3 T,COS(T),1; "Funcion COS(T)..."
@FUNC M1 1 4 Y,LN(Y+1),1; "Funcion LN(Y+1)..."

```

```
"Resultado:" IMPMAT M1;
```

"Se calculan las mismas funciones para la matriz M2 pero tomando como variable independiente los numeros generados para el renglon 1..."

```
@FUNR M2 2 1 X,SEN(X),1; "Funcion SEN(X)..."
@FUNR M2 3 1 T,COS(T),1; "Funcion COS(T)..."
@FUNR M2 4 1 Y,LN(Y+1),1; "Funcion LN(Y+1)..."

```

```
"Resultado:" IMPMAT M2;
```

```
*****
* ORFEO-90 PROCESADOR NUMERICO DIDACTICO PARA ESTRUCTURAS *
*                               Ver 1.5                               *
* E-04                                                                    *
*                                                                           *
* Objetivo: Mostrar al usuario del paquete el uso de los *
* operadores matematicos para matrices *
*****
```

*** OPERADORES ***

Funcion: SUM

Formato: SUM <M1> <M2> <M3>;

- Se realiza la operacion $M1 + M2$. El arreglo $M3$ es opcional. Si es proporcionado, la suma es asignada a $M3$; $M3 = M1 + M2$, de lo contrario es asignada a $M1$; $M1 = M1 + M2$.

Funcion: RES

Formato: RES <M1> <M2> <M3>;

- Realiza la operacion $M1 - M2$. El funcionamiento es similar al operador SUM.

Funcion: MULT

Formato: MULT <M1> <M2> <M3>;

- Realiza el producto matricial de $M1$ y $M2$ asignando el resultado a $M3$.

Funcion: TMULT

Formato: TMULT <M1> <M2> <M3>;

- Realiza el producto matricial de $M1$ traspuesta por $M2$ igual a $M3$.

Funcion: TTMULT

Formato: TTMULT <M1> <M2> <M3>;

- Realiza el producto matricial de $M1$ traspuesta por $M2$ por $M1$, asignando el resultado a $M3$.

EJEMPLOS:

Se cargan los arreglos $M1$ y $M2...$

LEHAT M1 R R=5 C=5;

```
1; 1; 1; 1; 1;
2; 2; 2; 2; 2;
3; 3; 3; 3; 3;
4; 4; 4; 4; 4;
5; 5; 5; 5; 5;
```

LEHAT M2 R R=5 C=5;

```
1; 1; 1; 1; 1;
1; 1; 1; 1; 1;
1; 1; 1; 1; 1;
1; 1; 1; 1; 1;
1; 1; 1; 1; 1;
```

```
"M3 = M1 + M2" SUM M1 M2 M3; IMPMAT M3;
"M3 = M1 - M2" RES M1 M2 M3; IMPMAT M3;
"M3 = M1 * M2" MULT M1 M2 M3; IMPMAT M3;
"M3 = M1** M2" TMULT M1 M2 M3; IMPMAT M3; "M1" es la traspuesta de M1"
"M3 = M1** M2 * M1" TTMULT M1 M2 M3; IMPMAT M3;
```

```
*****
* ORFEO-90 PROCESADOR NUMERICO DIDACTICO PARA ESTRUCTURAS *
* Ver 1.5 *
* E-05 *
* *
* Objetivo: Mostrar al usuario del paquete las funciones *
* para resoluci3n de sistemas de ecuaciones y *
* comandos relacionados. *
*****
```

**** FUNCIONES DE RESOLUCION DE SISTEMAS DE ECUACIONES LINEALES [A] [X]=[Y] ****

Funci3n: GAUSS
Formato: GAUSS <[A]> <[Y]>;

- [A] - Puede ser cualquier tipo de matriz excepto en banda y entera
- [Y] - Cualquier tipo de matriz excepto en banda y entera. Pueden ser varios vectores de t3rminos independientes.

La soluci3n se almacena en la matriz [Y].

Funci3n: CHOLESKI
Formato: CHOLESKI <[A]> <[Y]>;

- [A] - Debe ser matriz en banda exclusivamente.
- [Y] - Matriz tipo R (real). Se aceptan varios vectores de t3rminos independientes por columnas.

En el caso de no proporcionarse la matriz [Y] el comando TRIANGULA a la matriz [A] siempre y cuando se encuentre sin alteraci3n, es decir; en BANDA.

Si se proporcionan los dos arreglos, [A] y [Y] pueden presentarse dos opciones:

- 1.- Si la matriz [A] se encuentra sin alteraci3n (en BANDA), el comando TRIANGULA y RESUELVE (sustituciones hacia adelante e inversa sobre [Y]).
- 2.- Si la matriz [A] ya ha sido TRIANGULADA anteriormente, el comando solo RESUELVE (sustituciones sobre [Y].)

Todas las asignaciones y revisiones de estado (matriz en banda o triangulada) se realizan autom3ticamente. Si se encuentra algun error, el programa lo detectara y reportara oportunamente.

***** INVERSION DE MATRICES *****

Funci3n: INVERTIR
Formato: INVERTIR <[A]> <[B]>;

- [A] - Matriz de cualquier tipo, excepto en banda y entera, que se desea invertir.
- [B] - Opcional. Si se proporciona, la inversa de [A] se almacena en [B]. De lo contrario la matriz [A] es substituida por su inversa.

En caso de singularidad, el programa lo reportara inmediatamente, sin que se presente alteraci3n alguna en la matriz [A].

EJEMPLOS:

Se carga una matriz de HILBERT cuya inversa es perfectamente conocida, lo que permite, adem3s de mostrar el manejo de las funciones; evaluar la bondad de los resolvidores implementados en el paquete.

Considere" N=6;
LEMAT [A] R O=6;

" * MATRIZ DE HILBERT *****

```
[A] =
" (N+2)/(2*N+2); -1/2; 0; 0; 0; 0; 1/(2*N+2); "
" -1/2; 1; -1/2; 0; 0; 0; 0; "
" 0; -1/2; 1; -1/2; 0; 0; 0; "
" 0; 0; -1/2; 1; -1/2; 0; 0; "
" 0; 0; 0; -1/2; 1; -1/2; 0; "
" 1/(2*N+2); 0; 0; 0; -1/2; (N+2)/(2*N+2); "
```

LEMAT [A] R 0=6;

" *** MATRIZ INVERSA ***

```
[A1]= | " N; N-1; N-2; N-3; N-4; N-5; " |
      | " N-1; N; N-1; N-2; N-3; N-4; " |
      | " N-2; N-1; N; N-1; N-2; N-3; " |
      | " N-3; N-2; N-1; N; N-1; N-2; " |
      | " N-4; N-3; N-2; N-1; N; N-1; " |
      | " N-5; N-4; N-3; N-2; N-1; N; " |
```

" Se imprime la matriz [A] y su inversa [A1]..."

IMPMAT [A]; IMPMAT [A1];

" Se comprueba que [A1] es en verdad inversa de [A]..."

Sea X la inversa de [A]. Entonces si [A1] es tambien inversa de [A],
obviamente [A1]=X . Comprobando..."

INVERTIR [A] X; IMPMAT X;

" Suponga ahora que se resuelve el sistema [A] X = I donde I es la matriz
identidad. Es evidente que bajo estas circunstancias X asumira el valor
de [A1], de tal forma que [A1]=X . Comprobando..."

LEMAT I DI 0=6; "<= Se carga la matriz identidad I (diagonal)"

GAUSS [A] I; "NOTA: Recuerde que ahora I se ha convertido en matriz
real y almacena la solucion del sistema (X)."

IMPMAT I;

" Realizando el razonamiento anterior resolvamos el sistema [AB] X = I
pero ahora siendo [AB] la matriz [A] almacenada en BANDA."

LEMAT [AB] B R=6 C=6; "<= El ancho de banda es 6"

" *** MATRIZ DE HILBERT EN BANDA ***

```
[AB]= | " (N+2)/(2*N+2); -1/2; 0; 0; 0; 1/(2*N+2); " |
      | " 1; -1/2; 0; 0; 0; 0; " |
      | " 1; -1/2; 0; 0; 0; 0; " |
      | " 1; -1/2; 0; 0; 0; 0; " |
      | " 1; -1/2; 0; 0; 0; 0; " |
      | " (N+2)/(2*N+2); 0; 0; 0; 0; 0; " |
```

" Se carga la matriz identidad I..."

LEMAT I R 0=6; "<= La matriz de terminos independientes debe ser real"

```
1; 0; 0; 0; 0; 0;
0; 1; 0; 0; 0; 0;
0; 0; 1; 0; 0; 0;
0; 0; 0; 1; 0; 0;
0; 0; 0; 0; 1; 0;
0; 0; 0; 0; 0; 1;
```

" Para hacer mas ilustrativo el ejemplo se TRIANGULA [AB] y se imprime..."

CHOLESKI [AB]; IMPMAT [AB];

" Se resuelve el sistema con la matriz ya TRIANGULADA y se imprime el
resultado..."

CHOLESKI [AB] I; IMPMAT I;

```

*****
* ORFEO-90 PROCESADOR NUMERICO DIDACTICO PARA ESTRUCTURAS *
* Ver 1.5 *
* E-06 *
* *
* Objetivo: Mostrar al usuario del paquete el uso de la *
* función JACOBI en la resolución de problemas *
* de vectores y valores característicos. *
*****

```

*** VECTORES Y VALORES CARACTERISTICOS ***

Función: JACOBI
Formato: JACOBI <K> <M> <VEC> <VAL>;

Datos:

K - Matriz de rigideces tipo R (real) simétrica
POSITIVA DEFINIDA.
M - Matriz de masas tipo R o D (real o diagonal)

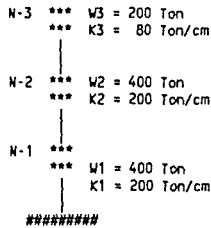
Resultados:

VEC - Matriz que contiene los vectores característicos
por columnas y ordenados de menor a mayor frecuencia.

VAL - Vector que contiene los valores característicos
en forma ascendente.

EJEMPLO:

Considere el sistema de la figura:



Se carga la matriz de rigideces [K] de orden 3x3..."

LEMAT [K] R 0=3;

```

"K1+K2=" 200+200;    " -K2=" -200;          0;
" -K2=" -200;        "K2+K3=" 200+80;    "-K3=" -80;
0;                  " -K3=" -80;          " K3=" 80;

```

" Se carga la matriz de masas [M] diagonal de orden 3x3..."

LEMAT [M] D 0=3;

```

"Masa del primer nivel =" 400/981;
"Masa del segundo nivel =" 400/981;
"Masa del tercer nivel =" 200/981;

```

"Se imprimen las matrices de rigideces y masas..."

```

IMPHAT [K];
IMPHAT [M];

```

"Se calculan los vectores y valores característicos (configuraciones modales y frecuencias)..."

```

JACOBI [K] [M] [F] {f}; " <= [F] matriz de formas modales, {f} vector de
frecuencias circulares."

```

"Se imprimen las matrices resultantes..."

```
IMPHAT (F); "<= Modos normalizados respecto a la matriz de masas"  
IMPHAT (F);
```

"Se calculan e imprimen los periodos de cada configuracion modal en el vector (T)..."

```
DUP (F) (T);  
RAIZ (T);  
INVERTER (T);  
ESCALA (T) (2*PI);  
IMPHAT (T);
```

"Se crea la grafica de las configuraciones modales en la matriz GRAF..."

```
LEMAT GRAF RC R=4 C=7;  
ALMSM (F) GRAF 2 3;  
GGENC GRAF 1 1 0,1;  
GGENC GRAF 1 6 -2,0; "<= Para centrar la imagen..."  
GGENC GRAF 1 7 2,0;  
TITULO(1)= CONFIGURACIONES MODALES (F);  
SUBTITULO(1)= NORMALIZACION RESPECTO A LA MATRIZ DE MASAS;  
EJEX(1)= CONFIGURACION;  
EJEY(1)= N I V E L;  
GRAFMY GRAF 1;
```

"NOTA: Como a continuacion se muestra, las matrices de masas y rigideces se modifican al aplicarse el comando JACOBI."

```
IMPHAT (K);  
IMPHAT (M);
```

```

*****
* ORFEO-90 PROCESADOR NUMERICO DIDACTICO PARA ESTRUCTURAS *
*                               Ver 1.5                               *
* E-07                                                                    *
*                               *
* Objetivo: Mostrar al usuario del paquete el uso de la *
*           funcion @DIN y comandos relacionados. *
*****

```

*** ANALISIS DINAMICO MODAL ***

Funcion: @DIN

Formato: @DIN W C Fp excitacion respuesta ninter intervalo;

Datos:

- W - Vector de frecuencias angulares.
- C - Vector de amortiguamiento. Cada renglon i contiene el amortiguamiento del modo i expresado como una fraccion del amortiguamiento critico.
- Fp - Vector de factores de participacion.
- excitacion - Arreglo que contiene la excitacion que se aplicara al sistema. La primera columna contiene al tiempo y la segunda a los valores de la aceleracion. No son necesarios intervalos constantes de tiempo.
- ninter - Numero de intervalos en los que se desea conocer la respuesta.

Resultados:

respuesta - Matriz real de orden ngl x ninter que contiene la respuesta del sistema.

ngl = Numero de grados de libertad.

Funcion: MAX

Formato: MAX respuesta vector clave;

- respuesta - Matriz que contiene la respuesta.
- vector - Contiene el maximo valor absoluto de las columnas (los renglones) de acuerdo con el tipo de clave empleada.
- clave - R para efectuar la busqueda de maximos sobre los renglones de la matriz respuesta. Genera un vector columna.
- C para efectuar la busqueda de maximos sobre las columnas de la matriz respuesta.

Funcion: @RCSC

Formato: @RCSC respuesta vector clave;

- respuesta - Matriz que contiene la respuesta.
- vector - Contiene la raiz cuadrada de la suma de los cuadrados de las columnas (los renglones) de acuerdo con el tipo de clave empleada.
- clave - R para efectuar el calculo sobre los renglones de la matriz respuesta. Genera un vector columna.
- C para efectuar el calculo sobre las columnas de la matriz respuesta. Genera un vector renglon.

EJEMPLO:

Considere los siguientes datos calculados a partir del sistema de tres grados de libertad del programa ejemplo E-06:

	Primer modo	Segundo modo	Tercer modo"
LEMAT W R R=3 C=1; "Frecuencias: "	11.05;	23.71;	37.08;
LEMAT C R R=3 C=1; "Amortiguamiento:"	0.20;	0.20;	0.20;
LEMAT Cp R R=3 C=1; "Coeficientes de participacion: "	0.5513;	0.2369;	0.2108;

LEMAT So R R=6 C=2;

**** Excitacion ****

"Tiempo	Aceleracion"
0.0;	0.000;
0.5;	0.500;
1.0;	0.500;
2.0;	-0.500;
2.5;	-0.500;
3.0;	0.000;

" Graficando la excitacion:"

TITULO(4)= ACELEROGRAMA;
SUBTITULO(4)= INTERVALOS VARIABLES;
EJEX(4)= T I E M P O;
EJEY(4)= ACELERACION;

GRAFMX So 4;

" Se calcula la respuesta U(t) del sistema para 20 intervalos con dt=0.15
(3 segundos del acelerograma);"

@DIN W C Cp So U(t) 20 0.15;

" Se grafica la respuesta:"

TITULO(4)= RESPUESTA;
SUBTITULO(4)= ANALISIS DINAMICO MODAL;
EJEY(4)= DESPLAZAMIENTOS;

LEMAT RES RC R=21 C=5;
@GENC RES 1 1 0,0.15;
TRANS U(t) UAUX;
ALMSH UAUX RES 1 3;
GRAFMX RES 4;

" Se obtiene la maxima respuesta con el criterio del maximo valor absoluto:"

MAX U(t) Mabs R;

" Se obtiene la maxima respuesta con el criterio de la raiz cuadrada de la
suma de los cuadrados:"

@RCSC U(t) Mrsc R;

" Se imprimen los resultados en pantalla..."

IMPHAT Mabs; IMPHAT Mrsc;

```
*****
* ORFEO-90 PROCESADOR NUMERICO DIDACTICO PARA ESTRUCTURAS *
*                               Ver 1.5                               *
* E-08                           *
* *                               *
* Objetivo: Mostrar al usuario del paquete el uso de la *
*          funcion @PASO y comandos relacionados. *
*****
```

*** ANALISIS PASO A PASO ***

Funcion: @PASO

Formato: @PASO K M C UVA U FAC F inter npnts dt,delta,alfa,theta;

Datos:

- K - Matriz de rigideces tipo B (real en banda).
- M - Matriz de masas tipo R o D (real o diagonal).
- C - Matriz de amortiguamiento tipo R o D.
- UVA - Matriz de real condiciones iniciales por columnas (desplazamiento U, velocidad V y aceleracion A) de orden grados de libertad x 3.
- FAC - Vector que contiene los factores que afectan a la excitacion de cada grado de libertad.
- F - Matriz de excitacion. Puede tener una o dos columnas. En el primer caso se condidera $F(i,1)=f(i)$, y en el segundo $F(i,1)=t(i)$ $F(i,2)=f(i)$. Los valores de a(i) deben ser a intervalos dt IGUALES. Utilice @ITP.
- inter - Indica cada cuantos puntos debe ser almacenada la respuesta.
- npnts - Numero de puntos de la respuesta que se desean almacenar.
- dt - Incremento de tiempo.

delta, alfa,theta - Parametros de integracion numerica. En caso de proporcionarse NINGUNO, se asume delta=1/2, alfa=1/6 y theta=1.00 (Newmark 1/6, variacion lineal).

Resultados:

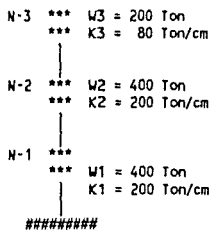
U - Matriz que contiene la respuesta (desplazamientos).

Funcion: @ITP

Formato: @ITP <Interpolacion> <Funcion a interpolar> <#intervalos> <incr>;

EJEMPLO:

Considere el sistema de la figura:



Se carga la matriz de rigideces K de orden 3x2 en banda..."

LEMAT K B R=3 C=2;

```
"K1+K2=" 200+200;    " -K2=" -200;
"K2+K3=" 200+80;    " -K3=" -80;
" K3=" 80;          " 0.0;
```

" Se carga la matriz de masas M diagonal de orden 3x3..."

LEMAT M D O=3;

"Masa del primer nivel =" 400/981;

"Masa del segundo nivel =" 400/981;

"Masa del tercer nivel =" 200/981;

"Se carga la matriz de amortiguamiento C = aM + bK simetrica de orden 3x3"

a=0.4; b=0.005;

LEMAT C S O=3;

a*B(E(M,1,1)+b*B(E(K,1,1); b*B(E(K,1,2); 0.0;

a*B(E(M,2,2)+b*B(E(K,2,1); b*B(E(K,2,2);

a*B(E(M,3,3)+b*B(E(K,3,1);

"Se carga la matriz UVA de condiciones iniciales, considerandolas nulas..."

LEMAT UVA RC O=3;

"Se carga el vector de factores FAC..."

LEMAT FAC R R=3 C=1; 1; 1; 1; "<= Sin alteracion"

"Se genera una excitacion con intervalos irregulares para mostrar el uso de la funcion @ITP..."

LEMAT F R R=3 C=2; "Puntos: Tiempo Fuerza"
0.0; 0.0;
0.05; 10.0; "<= Impulso"
0.1; 0.0;

"Se imprime la excitacion generada a intervalos variables..."

TITULO[1]= EXCITACION;
SUBTITULO[1]= INTERVALOS VARIABLES;
EJEX[1]= TIEMPO Seg ;
EJEY[1]= FUERZA Ton;

GRAFMX F 1;

"Se interpola la excitacion A en 100 intervalos iguales de 0.05 seg (5 segundos de excitacion) en el arreglo AUX..."

@ITP F AUX 100 0.05; "<= AUX contiene la funcion interpolada"

"Se construye y grafica una matriz FC con la excitacion interpolada..."

LEMAT FC RC R=100 C=2;
@GENC FC 1 1 0,0.05;
ALMSM AUX FC 1 2;

SUBTITULO[1]= INTERVALOS CONSTANTES DE 0.05 Segundos;
GRAFMX FC 1;

"Se realiza el analisis paso a paso obteniendo la respuesta cada 0.10 segundos en la matriz U (generada automaticamente de orden 3x50)..."

@PASO K M C UVA U FAC FC 2 50 0.05,1/2,1/6,1.0;

"Se construye la grafica de la respuesta..."

LEMAT RES RC R=50 C=5; @GENC RES 1 1 0,0.10; TRANS U; ALMSM U RES 1 3;
TITULO[1]= RESPUESTA DEL SISTEMA;
SUBTITULO[1]= INTERVALO 0.05 Segundos;
EJEY[1]= DESPLAZAMIENTO Cm;
GRAFMX RES 1;

```
*****
* ORFEO-90 PROCESADOR NUMERICO DIDACTICO PARA ESTRUCTURAS *
* Ver 1.5 *
* E-09 *
* *
* Objetivo: Mostrar al usuario del paquete las funciones *
* para graficacion. *
*****
```

*** FUNCIONES DE CONFIGURACION ***

Funcion: TITULO
Formato: TITULO[n]= <Titulo de la grafica>;

Funcion: SUBTITULO
Formato: SUBTITULO[n]= <Subtitulo de la grafica>;

En ambas funciones el numero maximo de caracteres es de 50

Funcion: EJEX
Formato: EJEX[n]= <Titulo del eje de las abscisas>;

Funcion: EJEY
Formato: EJEY[n]= <Titulo del eje de las ordenadas>;

En ambas funciones el numero maximo de caracteres es de 20
El parametro n indica cual es el formato definido n=1,2,3,4,5

*** FUNCIONES DE GRAFICACION ***

Funcion: GRAF2D
Formato: GRAF2D <F(var)>,<var>,<liminf>,<limsup>,<#intervalos>,<n>;

F(var) - Funcion que se desea graficar.
var - Variable independiente.
liminf - Valor minimo de la variable independiente.
limsup - Valor maximo de la variable independiente.
#intervalos - Numero de intervalos de graficacion.
n - Numero de formato (configuracion).
Si es omitido se coloca como titulo la funcion graficada y como subtitulo el intervalo de graficacion.

Funcion: GRAFMX
Formato: GRAFMX <matriz> <n>;

matriz - Arreglo que genera una grafica multiple tomando como variable INDEPENDIENTE a la primera columna, y como funciones TODAS las columnas restantes.
n - Numero de formato. Si es omitido se coloca como titulo el nombre de la matriz.

Funcion: GRAFMY
Formato: GRAFMY <matriz> <n>;

matriz - Arreglo que genera una grafica multiple tomando como variable DEPENDIENTE a la primera columna, y como diferentes variables INDEPENDIENTES a TODAS las columnas restantes.
n - Numero de formato. Si es omitido se coloca como titulo el nombre de la matriz.

Funcion: GRAF3D
Formato: GRAF3D <F(var1,var2)>,<var1>,<var2>,<liminf1>,<limsup1>,<liminf2>,<limsup2>,<#intervalos>,<n>;

F(var1,var2) - Funcion (superficie) que se desea graficar.
var1,var2 - Variables independientes.
liminf1,limsup1 - Limites inferior y superior de los valores que toma la variable independiente var1.
liminf2,limsup2 - Limites inferior y superior de los valores que toma la variable independiente var2.
#intervalos - Numero de intervalos en los que se evaluaran var1 y var2.
n - Numero de formato. Si es omitido se coloca como

titulo la funcion graficada y como subtítulo los intervalos de graficacion.

EJEMPLOS:

" Se definen configuraciones..."

```
TITULO[1]= ORFEO 90;
SUBTITULO[1]= GRAFICACION EN DOS DIMENSIONES;
EJEX[1]= A B S C I S A S;
EJEY[1]= O R D E N A D A S;
```

" Se grafica la funcion $\text{SEN}(X)$ para $[-\#P1,+\#P1]$ en 45 intervalos..."

```
GRAF2D SEN(X),X,-#P1,+#P1,45,1;
```

" Se grafica una matriz M de orden 29x4 en cuyas columnas de almacenan X, valores nulos, $\text{SEN}(X)$ y $\text{COS}(X)$..."

```
LEMAT M RC R=29 C=4;
@GENC M 1 1 -#P1,#P1/14;
@FUNC M 1 3 X,SEN(X),1; " <= Se construye la matriz Mx..."
@FUNC M 1 4 X,COS(X),1;
```

```
TITULO[2]= ORFEO 90;
SUBTITULO[2] FUNCIONES SEÑO Y COSEÑO  $[-\#P1,\#P1]$ ;
EJEX[2]= R A D I A N E S;
EJEY[2]= " <= Pueden ser nulos los formatos"
```

```
GRAFMX M 2;
```

" Se grafica la matriz M pero considerando ahora a las abscisas como las funciones definidas y sin formato grafico..."

```
GRAFMY M;
```

"Se grafica la funcion $\text{SEN}(X^2+Y^2)/(X^2+Y^2)$ para $[-\#P1 < X <+\#P1]$ y $[-\#P1 < Y <+\#P1]$ en una reticula de 15x15 intervalos con formato libre..."

```
GRAF3D SEN(X^2+Y^2)/(X^2+Y^2),X,Y,-#P1,+#P1,-#P1,+#P1,15;
```

**PROGRAMAS DE APLICACION
DE ORFEO-90**

B

En el disco de UTILERIAS se encuentran cinco programas de aplicación en el análisis estructural empleando ORFEO-90.

- A-01.PRG Se resuelve la armadura plana de la figura B.1 empleando un método matricial tradicional.
- A-02.PRG Generación de la matriz de rigideces asociada al sistema global de referencia para barras sometidas a fuerza axial. Librería KA2D.
- A-03.PRG Se resuelve la armadura plana del programa A-01.PRG empleando la librería KA2D.
- A-04.PRG Generación de la matriz de rigideces asociada al sistema global de referencia para barras sometidas a fuerza axial y momento flexionante. Librería KASG.
- A-05.PRG Generación de la matriz de rigideces en coordenadas locales para barras sometidas a fuerza axial y momento flexionante. Librería KASL.

NOTA: Los listados mostrados a continuación carecen de acentos por encontrarse escritos en el procesador de textos de ORFEO-90, el cual no los soporta.

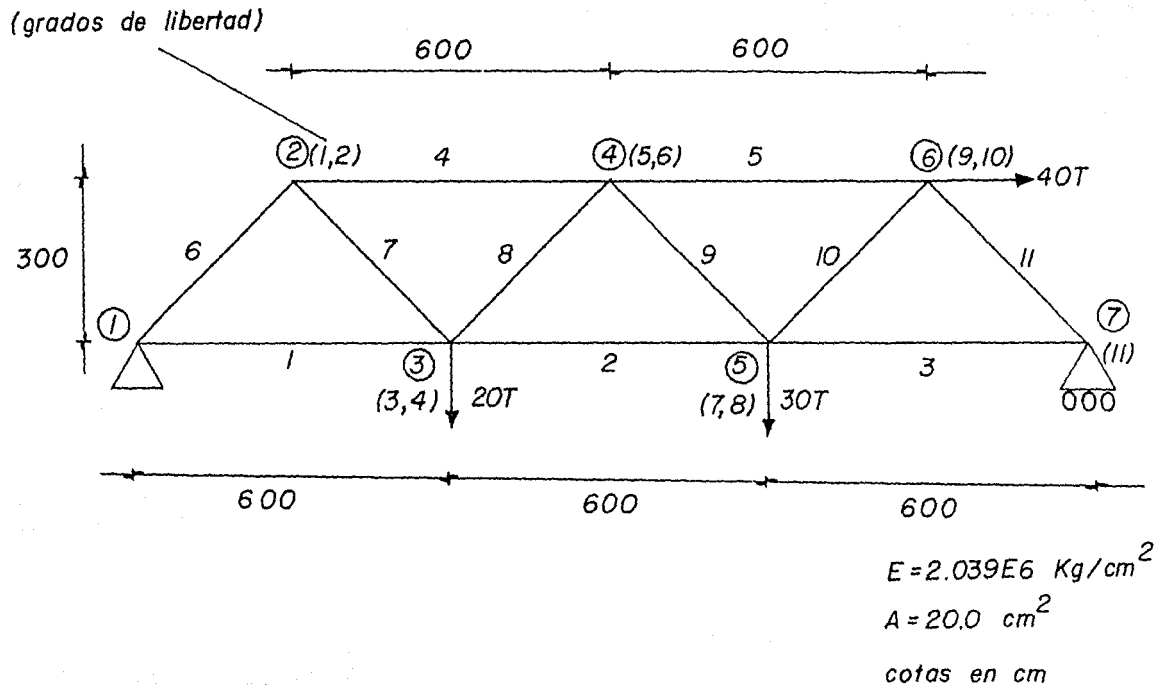


Figura B.1 Armadura plana utilizada en los ejemplos.


```

*****
* ORFEO-90 PROCESADOR NUMERICO DIDACTICO PARA ESTRUCTURAS *
* Ver 1.5 *
* A-01 *
* Ejemplo de aplicacion. Solucion de una armadura plana. *
*****

```

```

"           A R M A D U R A   2 - D           "
" cotas en metros.                            "
"           2 (1,2)   4 (5,6)   6 (9,10)      "
"           =====>                          "
"           / \      / \      / \             "
"          4   5   40 ton                      "
"         / \ / \ / \                          "
"        3.00 6 7 8 9 10 11                     "
"         \ / \ / \ /                          "
"          1   2   3                             "
"       +-----+ 7 (11)                         "
"      / \      / \      / \                     "
"     === 20 ton 3 (3,4) 5 (7,8) \ \          "
"      \ / \ / \ / \ \ \ \ \ \ \ \ \ \ \ \ \ "
"      + 6.00 + 6.00 + 6.00 +                  "

```

"Datos:"
E = 2.039E6; "kg/cm²"
A = 20.00; "cm²"
"#Nudos: 7"
"#Barras: 11"
"NOTA: Entre parentesis"
"los grados de libertad"
"para ensamble..."

" Matriz de transformacion [A]:

$$[A] = \begin{bmatrix} \cos(t) & \text{sen}(t) & 0 & 0 \\ 0 & 0 & \cos(t) & \text{sen}(t) \end{bmatrix}$$

donde t es el angulo de inclinacion de la barra.

Matriz de rigideces [k] local:

$$[k] = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} EA/L$$

Matriz de rigideces acoplada al sistema global [K]:

$$[K] = [A]^t [k] [A]$$

Se calcula las matrices de transformacion [A] y rigideces para cada barra

Barras 1 2 3 4 y 5:" t = 0.0;

LEMAT Aa R R=2 C=4;

$$\begin{bmatrix} \cos(t) & \text{sen}(t); & 0; & 0 \\ 0; & 0; & \cos(t); & \text{sen}(t); \end{bmatrix}$$

"Matriz de rigideces k1:" L = 600; "cm"

LEMAT k1 SD O=2; 1; 1; -1; ESCALA k1 E*A/L;

"Barras 6 8 y 10:" t = #PI/4;

LEMAT Ab R R=2 C=4;

$$\begin{bmatrix} \cos(t); & \text{sen}(t); & 0; & 0 \\ 0; & 0; & \cos(t); & \text{sen}(t); \end{bmatrix}$$

"Barras 7 9 y 11:" t = -#PI/4;

LEMAT Ac R R=2 C=4;

$$\begin{bmatrix} \cos(t); & \text{sen}(t); & 0; & 0 \\ 0; & 0; & \cos(t); & \text{sen}(t); \end{bmatrix}$$

"Matriz de rigideces k2 para barras 6 7 8 9 10 y 11:" L = 300/COS(#PI/4);

LEMAT k2 SD O=2; 1; 1; -1; ESCALA k2 E*A/L;

"Calculando la matriz de rigideces asociada al sistema global de referencia"

TTMULT Aa k1 KA;
TTMULT Ab k2 KB;
TTMULT Ac k2 KC;

"Se carga la matriz indicadora de ecuacion IND por renglones..."

LEMAT IND E R=11 C=4;

"	dx _i	dy _i	dx _j	dy _j	"
"Barra 1:"	0;	0;	3;	4;	
"Barra 2:"	3;	4;	7;	8;	
"Barra 3:"	7;	8;	11;	0;	
"Barra 4:"	1;	2;	5;	6;	
"Barra 5:"	5;	6;	9;	10;	
"Barra 6:"	0;	0;	1;	2;	
"Barra 7:"	1;	2;	3;	4;	
"Barra 8:"	3;	4;	5;	6;	
"Barra 9:"	5;	6;	7;	8;	
"Barra 10:"	7;	8;	9;	10;	
"Barra 11:"	9;	10;	11;	0;	

"Se inicializa la matriz global de rigideces KG de orden (11x11)..."

LEMAT KG RC O=11;

"Se ensambla la contribucion de cada barra en la matriz de rigideces KG..."

@ENSB KA KG IND R=1;
@ENSB KA KG IND R=2;
@ENSB KA KG IND R=3;
@ENSB KA KG IND R=4;
@ENSB KA KG IND R=5;
@ENSB KB KG IND R=6;
@ENSB KC KG IND R=7;
@ENSB KB KG IND R=8;
@ENSB KC KG IND R=9;
@ENSB KB KG IND R=10;
@ENSB KC KG IND R=11;

"Se genera el vector de cargas D..." LEMAT D R R=11 C=1;

0; 0; 0; -20000; 0; 0; 0; -30000; 40000; 0; 0;

"Se resuelve el sistema de ecuaciones..."

GAUSS KG D;

"Se calculan las fuerzas de cada barra..."

@FRZ F1 Aa k1 D IND R=1;
@FRZ F2 Aa k1 D IND R=2;
@FRZ F3 Aa k1 D IND R=3;
@FRZ F4 Aa k1 D IND R=4;
@FRZ F5 Aa k1 D IND R=5;
@FRZ F6 Ab k2 D IND R=6;
@FRZ F7 Ac k2 D IND R=7;
@FRZ F8 Ab k2 D IND R=8;
@FRZ F9 Ac k2 D IND R=9;
@FRZ F10 Ab k2 D IND R=10;
@FRZ F11 Ac k2 D IND R=11;

"Se imprimen los desplazamientos D..."

IMPHAT D;

"Se imprimen las fuerzas de cada barra (ACCION DE BARRA SOBRE NUDO)..."

IMPHAT F1; IMPHAT F2; IMPHAT F3; IMPHAT F4; IMPHAT F5; IMPHAT F6;
IMPHAT F7; IMPHAT F8; IMPHAT F9; IMPHAT F10; IMPHAT F11;

```
*****
* ORFEO-90 PROCESADOR NUMERICO DIDACTICO PARA ESTRUCTURAS *
*                               Ver 1.50                      *
* A-02                                                                    *
*                                                                    *
* Generacion de la matriz de rigideces asociada al sistema *
* global de referencia de una barra sometida a fuerza axial *
* para utilizacion posterior.                                          *
*                                                                    *
*                               Libreria KA2D                    *
*****
```

LEHAT KA2D A 0=4;

$E^*A/L^*COS(X)^2$; $E^*A/L^*COS(X)*SEN(X)$; $-E^*A/L^*COS(X)^2$; $-E^*A/L^*COS(X)*SEN(X)$;
 $E^*A/L^*SEN(X)^2$; $-E^*A/L^*COS(X)*SEN(X)$; $-E^*A/L^*SEN(X)^2$;
 $E^*A/L^*COS(X)^2$; $E^*A/L^*COS(X)*SEN(X)$;
 $E^*A/L^*SEN(X)^2$;

IMPHAT KA2D;
SALVA KA2D;

```

*****
* ORFEO-90 PROCESADOR NUMERICO DIDACTICO PARA ESTRUCTURAS *
* Ver 1.5 *
* A-03 *
* Aplicacion de la libreria KA2D. *
*****

```

```

" ARMADURA 2 - D "
" "
" cotas en metros. "
" "
" 2 (1,2) 4 (5,6) 6 (9,10) "
" + -----> "
" "
" 3.00 / \ 4 / \ 5 / \ 40 ton "
" / 6 \ 7 \ 8 \ 9 \ 10 \ 11 "
" / 1 \ 2 \ 3 \ 7 (11) "
" + -----> "
" / \ 3 (3,4) 5 (7,8) / \ "
" === 20 ton 30 ton 0000 "
" "
" + 6.00 + 6.00 + 6.00 + "
" "
"Datos:"
" E = 2.039E6; "kg/cm^2"
" A = 20.00; "cm^2"
"#Nudos: 7"
"#Barras: 11"
"NOTA: Entre parentesis"
"los grados de libertad"
"para ensemble..."

```

```

" *** BARRA TIPICA *** "
" "
" y nudo j "
" * dxj,dyj "
" "
" | / barra orientada de i a j "
" | / ) angulo X "
" |-----> x "
" nudo i "
" dxj,dyj "

```

"Se carga la matriz de rigideces para armaduras planas..."

CARGA KA2D;

"Se carga la matriz indicadora de ecuacion IND por renglones..."

LEMAT IND E R=11 C=4;

```

" dxi dyi dxj dyj "
"Barra 1:" 0; 0; 3; 4;
"Barra 2:" 3; 4; 7; 8;
"Barra 3:" 7; 8; 11; 0;
"Barra 4:" 1; 2; 5; 6;
"Barra 5:" 5; 6; 9; 10;
"Barra 6:" 0; 0; 1; 2;
"Barra 7:" 1; 2; 3; 4;
"Barra 8:" 3; 4; 5; 6;
"Barra 9:" 5; 6; 7; 8;
"Barra 10:" 7; 8; 9; 10;
"Barra 11:" 9; 10; 11; 0;

```

"Se inicializa la matriz global de rigideces KG de orden (11x11)..."

LEMAT KG RC 0=11;

"Se evaluan las matrices de rigideces y transformacion para barras tipo."

"Barras 1 2 3 4 y 5: Angulo" X=0.0; "radianes, Longitud" L=600; "cm"

VALUA KA2D Ka; " <= Matriz de rigideces asociada al sistema global"

LEMAT Aa R R=2 C=4; " <= Matriz de transformacion"

```

COS(X); SEN(X); 0.0; 0.0;
0.0; 0.0; COS(X); SEN(X);

```

```

LEMAT KlA SD O=2;  "=<= Matriz de rigideces en coordenadas locales"
      E*A/L; E*A/L; -E*A/L;

"Barras 6 8 y 10: Angulo" X=#PI/4; "radianes, Longitud" L=300/COS(X);"cm"

VALUA KA2D Kb;
LEMAT Ab R R=2 C=4;  COS(X);  SEN(X);  0.0;  0.0;
                    0.0;  0.0;  COS(X);  SEN(X);

LEMAT KlB SD O=2; 1; 1; -1; ESCALA KlB E*A/L;

"Barras 7 9 y 11: Angulo" X=-#PI/4; "radianes, Longitud idem anterior"

VALUA KA2D Kc;
LEMAT Ac R R=2 C=4;  COS(X);  SEN(X);  0.0;  0.0;
                    0.0;  0.0;  COS(X);  SEN(X);

"La matriz de rigideces local es la misma que la anterior"

"Se ensambla la contribucion de cada barra en la matriz de rigideces KG..."

@ENSB Ka KG IND R=1;
@ENSB Ka KG IND R=2;
@ENSB Ka KG IND R=3;
@ENSB Ka KG IND R=4;
@ENSB Ka KG IND R=5;
@ENSB Kb KG IND R=6;
@ENSB Kc KG IND R=7;
@ENSB Kb KG IND R=8;
@ENSB Kc KG IND R=9;
@ENSB Kb KG IND R=10;
@ENSB Kc KG IND R=11;

"Se genera el vector de cargas D..."

LEMAT D R R=11 C=1;

0; 0; 0; -20000; 0; 0; 0; -30000; 40000; 0; 0;

"Se resuelve el sistema de ecuaciones..."

GAUSS KG D;

"Se calculan las fuerzas de cada barra..."

@FRZ F1 Aa KlA D IND R=1;
@FRZ F2 Aa KlA D IND R=2;
@FRZ F3 Aa KlA D IND R=3;
@FRZ F4 Aa KlA D IND R=4;
@FRZ F5 Aa KlA D IND R=5;
@FRZ F6 Ab KlB D IND R=6;
@FRZ F7 Ac KlB D IND R=7;
@FRZ F8 Ab KlB D IND R=8;
@FRZ F9 Ac KlB D IND R=9;
@FRZ F10 Ab KlB D IND R=10;
@FRZ F11 Ac KlB D IND R=11;

"Se imprimen los desplazamientos D..."

IMPAT D;

"Se imprimen las fuerzas de cada barra (ACCION DE BARRA SOBRE MUDDO)..."

IMPAT F1; IMPAT F2; IMPAT F3; IMPAT F4; IMPAT F5; IMPAT F6;
IMPAT F7; IMPAT F8; IMPAT F9; IMPAT F10; IMPAT F11;

```

```

*****
* ORFEO-90 PROCESADOR NUMERICO DIDACTICO PARA ESTRUCTURAS *
*                               Ver 1.5                               *
* A-04                                                                    *
*                                                                    *
* Generacion de la matriz de rigideces asociada al sistema *
* global de referencia de una barra prismatica de seccion *
* constante sometida a flexion y fuerza axial para *
* utilizacion posterior. *
*                                                                    *
*                               Libreria KASG.MAT                       *
*****

```

Nomenclatura:

E - Modulo de elasticidad.
A - Area de la seccion transversal.
I - Momento de inercia.
L - Longitud de la barra.
X - Angulo de la barra.

LEMAT KASG A 0=6;

```

"Elemento 1,1:" E*A/L*COS(X)^2+12*E*I/L^3*SEN(X)^2;
"Elemento 1,2:" (E*A/L-12*E*I/L^3)*SEN(X)*COS(X);
"Elemento 1,3:" -6*E*I/L^2*SEN(X);
"Elemento 1,4:" -(E*A/L*COS(X)^2+12*E*I/L^3*SEN(X)^2);
"Elemento 1,5:" -(E*A/L-12*E*I/L^3)*SEN(X)*COS(X);
"Elemento 1,6:" -6*E*I/L^2*SEN(X);

"Elemento 2,2:" E*A/L*SEN(X)^2+12*E*I/L^3*COS(X)^2;
"Elemento 2,3:" 6*E*I/L^2*COS(X);
"Elemento 2,4:" -(E*A/L-12*E*I/L^3)*COS(X)*SEN(X);
"Elemento 2,5:" -(E*A/L*SEN(X)^2+12*E*I/L^3*COS(X)^2);
"Elemento 2,6:" 6*E*I/L^2*COS(X);

"Elemento 3,3:" 4*E*I/L;
"Elemento 3,4:" 6*E*I/L^2*SEN(X);
"Elemento 3,5:" -6*E*I/L^2*COS(X);
"Elemento 3,6:" 2*E*I/L;

"Elemento 4,4:" E*A/L*COS(X)^2+12*E*I/L^3*SEN(X)^2;
"Elemento 4,5:" (E*A/L-12*E*I/L^3)*SEN(X)*COS(X);
"Elemento 4,6:" 6*E*I/L^2*SEN(X);

"Elemento 5,5:" E*A/L*SEN(X)^2+12*E*I/L^3*COS(X)^2;
"Elemento 5,6:" -6*E*I/L^2*COS(X);

"Elemento 6,6:" 4*E*I/L;

```

IMPAT KASG;
SALVA KASG;

```

*****
* DRFEO-90 PROCESADOR NUMERICO DIDACTICO PARA ESTRUCTURAS *
*                               Ver 1.5                      *
* A-05                                                                    *
* *                                                                    *
* Generacion de la matriz de rigideces asociada al sistema *
* Local de referencia de una barra prismatica de seccion *
* constante sometida a flexion y fuerza axial para        *
* utilizacion posterior.                                       *
* *                                                                    *
*                               Libreria KASL.MAT              *
*****

```

Nomenclatura:

E - Modulo de elasticidad.
A - Area de la seccion transversal.
I - Momento de inercia.
L - Longitud de la barra.

LEMAT KASL A 0=6;

```

"Elemento 1,1:" E*A/L;
"Elemento 1,2:" 0.0;
"Elemento 1,3:" 0.0;
"Elemento 1,4:" -E*A/L;
"Elemento 1,5:" 0.0;
"Elemento 1,6:" 0.0;

"Elemento 2,2:" 12*E*I/L^3;
"Elemento 2,3:" 6*E*I/L^2;
"Elemento 2,4:" 0.0;
"Elemento 2,5:" -12*E*I/L^3;
"Elemento 2,6:" 6*E*I/L^2;

"Elemento 3,3:" 4*E*I/L;
"Elemento 3,4:" 0.0;
"Elemento 3,5:" -6*E*I/L^2;
"Elemento 3,6:" 2*E*I/L;

"Elemento 4,4:" E*A/L;
"Elemento 4,5:" 0.0;
"Elemento 4,6:" 0.0;

"Elemento 5,5:" 12*E*I/L^3;
"Elemento 5,6:" -6*E*I/L^2;

"Elemento 6,6:" 4*E*I/L;

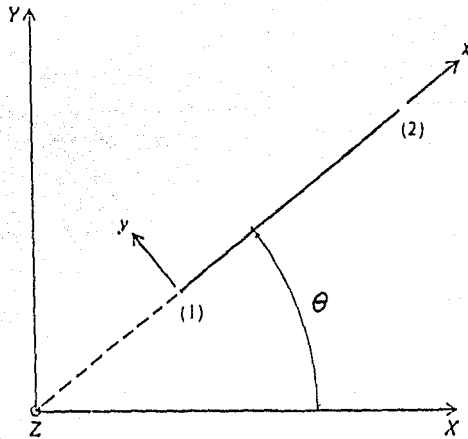
```

IMPAT KASL;
SALVA KASL;

MATRICES DE RIGIDECES Y TRANSFORMACION

C

C.1 BARRA SOMETIDA A FUERZA AXIAL EN EL PLANO



Matriz de rigideces referida al sistema local de coordenadas:

$$[k] = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \frac{EAx}{L}$$

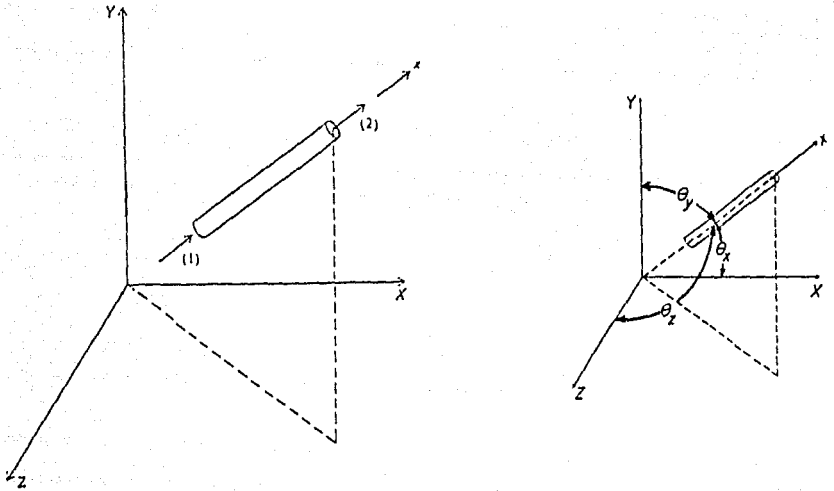
Matriz de transformación:

$$[A] = \begin{bmatrix} \cos \theta & \text{sen } \theta & 0 & 0 \\ 0 & 0 & \cos \theta & \text{sen } \theta \end{bmatrix}$$

Matriz de rigideces asociada al sistema global:

$$[K] = [A]^t [k] [A]$$

C.2 BARRA SOMETIDA A FUERZA AXIAL EN EL ESPACIO



Matriz de rigideces referida al sistema local de coordenadas:

$$[k] = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \frac{EAx}{L}$$

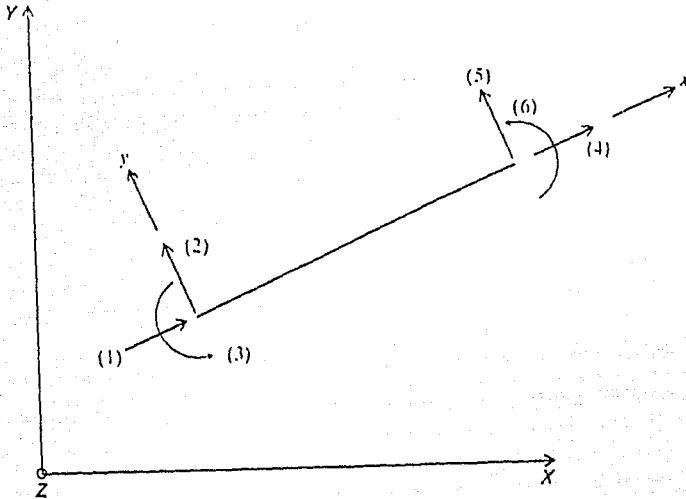
Matriz de transformación:

$$[A] = \begin{bmatrix} \cos \theta_x & \cos \theta_y & \cos \theta_z & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos \theta_x & \cos \theta_y & \cos \theta_z \end{bmatrix}$$

Matriz de rigideces asociada al sistema global:

$$[K] = [A]^t [k] [A]$$

C.3 BARRA SOMETIDA A FUERZA AXIAL Y MOMENTO FLEXIONANTE EN EL PLANO



Matriz de rigideces referida a coordenadas locales:

$$[k] = \begin{bmatrix} \frac{EAx}{L} & 0 & 0 & -\frac{EAx}{L} & 0 & 0 \\ 0 & \frac{12EIz}{L^3} & \frac{6EIz}{L^2} & 0 & -\frac{12EIz}{L^3} & \frac{6EIz}{L^2} \\ 0 & \frac{6EIz}{L^2} & \frac{4EIz}{L} & 0 & -\frac{6EIz}{L^2} & \frac{2EIz}{L} \\ -\frac{EAx}{L} & 0 & 0 & \frac{EAx}{L} & 0 & 0 \\ 0 & -\frac{12EIz}{L^3} & -\frac{6EIz}{L^2} & 0 & \frac{12EIz}{L^3} & -\frac{6EIz}{L^2} \\ 0 & \frac{6EIz}{L^2} & \frac{2EIz}{L} & 0 & -\frac{6EIz}{L^2} & \frac{4EIz}{L} \end{bmatrix}$$

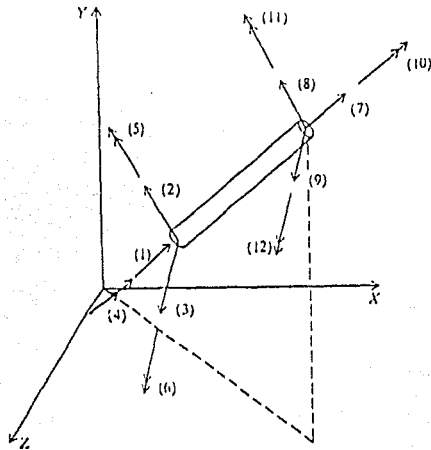
Matriz de transformación de coordenadas:

$$[A] = \begin{bmatrix} \cos \theta & \text{sen } \theta & 0 & 0 & 0 & 0 \\ -\text{sen } \theta & \cos \theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos \theta & \text{sen } \theta & 0 \\ 0 & 0 & 0 & -\text{sen } \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Matriz de rigideces asociada al sistema global:

$$[K] = [A]^t [k] [A]$$

C.4 BARRA SOMETIDA A FUERZA AXIAL Y MOMENTOS FLEXIONANTE Y TORSIONANTE EN EL ESPACIO



Matriz de rigideces [k] en coordenadas locales:

$$\begin{bmatrix}
 \frac{EA_x}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{EA_x}{L} & 0 & 0 & 0 & 0 & 0 \\
 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} & 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \\
 0 & 0 & \frac{12EI_y}{L^3} & 0 & 0 & \frac{6EI_y}{L^2} & 0 & 0 & -\frac{12EI_y}{L^3} & 0 & 0 & \frac{6EI_y}{L^2} \\
 0 & 0 & 0 & \frac{IxG}{L} & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{IxG}{L} & 0 \\
 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 & 0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\
 0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \\
 -\frac{EA_x}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA_x}{L} & 0 & 0 & 0 & 0 & 0 \\
 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} & 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} \\
 0 & 0 & -\frac{12EI_y}{L^3} & 0 & 0 & -\frac{6EI_y}{L^2} & 0 & 0 & \frac{12EI_y}{L^3} & 0 & 0 & -\frac{6EI_y}{L^2} \\
 0 & 0 & 0 & -\frac{IxG}{L} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{IxG}{L} & 0 \\
 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 & 0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 \\
 0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L}
 \end{bmatrix}$$

Matriz de transformación [A]:

$$[A] = \begin{bmatrix} [A]' & & & \\ & [A]' & & \\ & & [A]' & \\ & & & [A]' \end{bmatrix}$$

$$[A]' = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix}$$

donde

$$a_1 = \cos \theta_x$$

$$a_2 = \cos \theta_y$$

$$a_3 = \cos \theta_z$$

$$a_4 = \frac{-\cos \theta_x \cos \theta_y}{(\cos^2 \theta_x + \cos^2 \theta_z)^{1/2}}$$

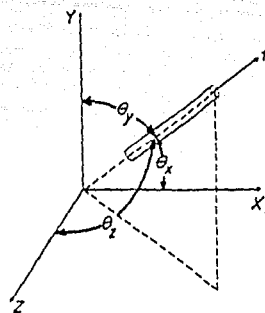
$$a_5 = (\cos^2 \theta_x + \cos^2 \theta_z)^{1/2}$$

$$a_6 = \frac{-\cos \theta_y \cos \theta_z}{(\cos^2 \theta_x + \cos^2 \theta_z)^{1/2}}$$

$$a_7 = \frac{-\cos \theta_z}{(\cos^2 \theta_x + \cos^2 \theta_z)^{1/2}}$$

$$a_8 = 0.0$$

$$a_9 = \frac{\cos \theta_x}{(\cos^2 \theta_x + \cos^2 \theta_z)^{1/2}}$$



Valido sólo en el caso de que θ_y sea distinto de 0 y 180 grados. Cuando sucede lo contrario, los valores de a_n son:

$$a_1 = a_3 = a_5 = a_6 = a_7 = a_8 = 0$$

$$a_2 = \cos \theta_y$$

$$a_4 = -\cos \theta_y$$

$$a_9 = 1.0$$

Para todos los casos la matriz de rigideces asociada al sistema global de referencia está dada por:

$$[K] = [A]^t [k] [A]$$

TABLA DE ERRORES

D

TABLA DE ERRORES

CODIGO

- 1 Formato o asignación inválida.
- 2 Paréntesis incompletos.
- 3 División entre cero.
- 4 Fuera de dominio en función $ASEN(n)$. $n \notin [-1.0, 1.0]$
- 5 Fuera de dominio en función $ACOS(n)$. $n \notin [-1.0, 1.0]$
- 6 Fuera de rango en función $ALN(n)$ $n > 709$
- 7 Fuera de dominio en función $LN(n)$ $n < 0$
- 8 Fuera de rango en función $LN(n)$ $n = 0$
- 9 Fuera de dominio en función $LOG(n)$ $n < 0$
- 10 Fuera de rango en función $LOG(n)$ $n = 0$
- 11 Fuera de rango en función $ALOG(n)$ $n > 307$
- 12 Fuera de dominio en exponenciación X^n $X = 0.0$ y $n = 0.0$
- 13 Fuera de dominio en exponenciación X^n $X = 0.0$ y n fracción.
- 14 No se proporciona expresión alguna.
- 15 Función desconocida.
- 16 Argumento inválido en la función factorial $FAC(n)$
 $n < 0$ o $n > 170$
- 17 Tipo de integración numérica inválido.
- 18 Número de intervalos inválido en función @INT
- 20 No es posible acceder al directorio indicado.
- 21 El archivo de lectura no existe.
- 22 No es posible acceder al archivo indicado.
Extensión inválida.
- 23 Editor programado vacío.
- 24 No es posible acceder al archivo indicado.
Archivo protegido contra escritura.
- 25 Fallo en gestión de memoria para un procedimiento interno.
- 26 La ejecución sólo es posible en modo programado.
- 27 Puerto abierto o unidad de disco inexistente.

- 28 Inicialización imposible por incompatibilidad del sistema.
No se encuentra el manejador del sistema gráfico.
El manejador de video no se encuentra en el directorio activo.
- 29 Editor interactivo vacío. No se ha definido arreglo alguno.
- 30 No se proporciona el nombre o nombres de los arreglos requeridos por la función.
- 31 Fallo en gestión de memoria para una matriz.
- 32 Editor programado lleno. Se sobrepasa el número máximo de líneas permitidas (150).
- 33 No se proporciona el nombre del arreglo en la función LEMAT
- 34 El nombre o la sentencia supera el número máximo de caracteres permitidos.
- 35 La matriz gestionada no se encuentra en memoria.
- 36 Tipo de matriz indefinido.
- 37 Tipo de matriz inválido.
- 38 Número de parámetros primarios inválido en función LEMAT
- 39 La matriz supera el máximo tamaño permitido.
- 40 Operación irrealizable. Incongruencia en definición de matrices.
- 41 Inconformabilidad de matrices en el procedimiento.
Asignación ilógica.
- 42 Aplicación de operadores o funciones inválidos al tipo de matriz proporcionado.
- 43 Impresora inexistente o apagada.
- 44 Elemento nulo en la matriz.
- 45 Elemento negativo en la matriz.
- 47 El elemento gestionado a la matriz sobrepasa el número de renglones y/o columnas. Fuera de orden.
- 48 Expresión que supera al número máximo de caracteres permitidos en modo programado.
- 50 No es posible salvar la matriz indicada.
El arreglo existe como librería.
- 51 No es posible cargar la matriz indicada.
Matriz no salvada con anterioridad.
- 52 Número de formato inválido. Consulte las funciones GRAF...
- 53 El número máximo de caracteres para TITULO o SUBTITULO ha sido superado.

- 54 El número máximo de caracteres para los ejes coordenados ha sido superado.
- 55 El número mínimo de intervalos es 2. Consulte la función GRAF2D
- 56 El número mínimo de intervalos es 3 y el máximo 50. Consulte la función GRAF3D
- 57 La matriz proporcionada debe tener al menos dos columnas.
- 58 Matriz singular.
- 59 Duplicación de matrices.
- 60 Número de intervalos y/o intervalo inválido.
- 61 Parámetro(s) de integración inválido(s). Consulte la función @PASO.

**TECNICAS DE PROGRAMACION
UTILIZADAS EN EL
DESARROLLO
DE ORFEO-90**

E

E.1 LENGUAJE DE PROGRAMACION

Durante muchos años, y hasta la fecha, el lenguaje más utilizado en el ámbito de las ciencias físico - matemáticas ha sido el FORTRAN. Debido a ello los fabricantes de compiladores no han escatimado esfuerzos para ofrecer, sin romper el esquema fundamental del lenguaje, las mayores prestaciones en computadoras personales. Sin embargo, recordemos que el nacimiento del FORTRAN se remonta a 1957, cuando la IBM lo implantó a su modelo 704, y desde entonces ha tenido que ser modificado en múltiples ocasiones para seguir siendo competitivo. Así, hoy en día se cuenta con un sinúmero de librerías e interfases para realizar operaciones para las cuales no se pensó fuera usado. Un ejemplo de ello se encuentra en algunos paquetes de funciones orientadas a la representación gráfica de resultados. Entonces, ¿porqué cambiar de lenguaje?

El FORTRAN es un lenguaje de alto nivel, lo que implica cierta rigidez de programación. Supongamos por un momento que se desea hacer una función para calcular el punto medio entre dos pares de coordenadas dentro de un dispositivo de graficación. Obviamente, estos números son enteros. No se entrará en detalles, pero resulta claro que esta tarea involucra un par de divisiones entre dos, una de las operaciones más costosas si no se cuenta con la posibilidad de un manejo óptimo de los recursos de la computadora. Recordemos que la arquitectura de cualquier ordenador está basada en la presencia y ausencia de corriente, es decir, un sistema binario. Entonces si se requiere dividir entre dos un número entero representado en binario resultará tan fácil como hacerlo entre diez en un sistema decimal, corriendo el punto. De esta manera, si pudiéramos de alguna forma indicar al procesador de la computadora que corriera los bits de un número en binario, dividiríamos o multiplicáramos por potencias de dos, con una rapidez sumamente alta.

Pongamos otro ejemplo. Tomemos un proceso recursivo como el factorial. La definición de éste es $n! = n(n-1)!$, considerando que el factorial de cero es uno. Entonces realizando una función que se llame a sí misma atendiendo a la definición del proceso, el problema se resolvería de una manera sencilla. Lo malo es que el FORTRAN no permite llamadas recursivas. En este punto seguramente pensará usted que la función factorial puede ser programada de muchas formas. Tiene toda la razón. Sin embargo, uno de los pilares más importantes en la inteligencia artificial es la recursividad, y si queremos contar con programas altamente eficientes que desarrollen tareas más allá del simple cálculo numérico, debemos utilizar estas bondades.

Se podrían dar muchos más ejemplos, sobre todo cuando entráramos a tópicos como el manejo del entorno del sistema y la velocidad de procesamiento, pero no es el objetivo de este apartado. Por lo tanto, ¿cuál lenguaje nos permite un control tan minucioso de la computadora, sin llegar al ensamblador ?

La respuesta es C.

Desarrollado por Dennis Ritchie a partir del lenguaje BCPL en la década del 70, el lenguaje C se ha convertido en uno de los más populares de nuestros tiempos. Inicialmente se creó para la programación de sistemas operativos pero su carácter estructurado, padre de la flexibilidad, lo ha hecho ideal para una programación sumamente eficiente. Por otra parte, la generación de código ejecutable compilado de un programa escrito en C es mucho más compacto, altamente portable y rápido, lo que imprime un mayor atractivo para su utilización.

Estos últimos son algunos de los puntos más importantes por los que diversos programas para análisis matricial de estructuras y elemento finito se

han empezado a escribir en lenguaje C, y razón para haber empleado en el desarrollo de ORFEO-90 este lenguaje.

E.2 ENTORNO DE LAS COMPUTADORAS PERSONALES

Cuando la informática comenzó a dar frutos con la aparición de la primera computadora, la gente reaccionó con elucubraciones que iban, desde temores inherentes a lo desconocido, hasta las ideas más descabelladas en el umbral de la ciencia ficción. Sin embargo, con el desarrollo de las computadoras personales, poco a poco nos hemos ido habituando a la gran utilidad que representa el empleo de los ordenadores.

Esto ha producido que, en la elaboración de software orientado a personas que no les interesa la programación en sí, sino la utilización de paquetes para un propósito específico, se hayan implementado funciones, que permitan un manejo rápido, cómodo y sencillo, sobretodo en las interfases para comunicación entre el usuario y el ordenador.

Cuando usted compara dos programas que hagan exactamente lo mismo, y uno de ellos, sin importar que sea menos eficiente en el desarrollo de sus tareas internas, cuenta con pantallas de presentación a color y con efectos visuales agradables en la entrada y salida de datos, seguramente tendrá una mayor demanda. De esto se desprende la necesidad de conocer a fondo, no sólo el tópico sobre el cual se desarrollará un programa, sino también el entorno que envuelve a la computadora, desde el sistema operativo, hasta los periféricos.

En el caso que nos atañe, la creación de ORFEO-90 contempló el entorno de los equipos personales basados en la familia del procesador 8086, los monitores con tarjetas CGA, MCGA, EGA, VGA y MDA (Hércules) y los equipos de impresión EPSON serie MX. Así mismo, el sistema operativo que sirvió de marco para el programa fue el MS-DOS versión 3.30 de Microsoft.

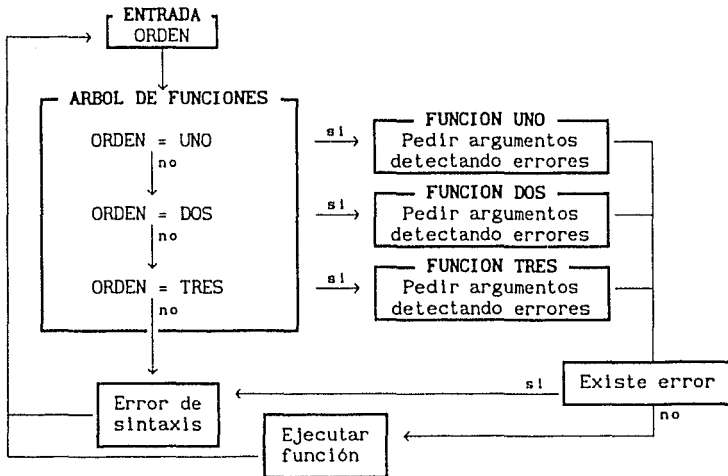
E.3 ANALIZADORES DE LENGUAJE

Uno de los aspectos fundamentales de los programas que interpretan algún tipo de sentencias proporcionadas por el usuario es el analizador. ORFEO-90 posee dos analizadores. El primero obedece a la tarea de interpretar el lenguaje propio del programa y el otro se encarga del análisis y evaluación de sentencias matemáticas.

En el primer caso, por las características inherentes al manejo de comandos e instrucciones, la programación no implica grandes dificultades. Únicamente es necesario analizar de izquierda a derecha y de arriba a abajo, cada una de las palabras empleadas en la definición de un procedimiento y compararlas con un árbol de funciones. Supongamos por un momento que se tienen tres funciones, cada una de las cuales requiere un número determinado de argumentos para realizar un procedimiento cualquiera tal y como se muestra a continuación:

FUNCION	ARGUMENTOS
UNO	U1, U2, U3
DOS	D1, D2
TRES	Sin argumentos.

Bajo estas circunstancias el diagrama de bloques del analizador estará dado por:



El mecanismo es a todas luces simple. Mediante el dispositivo de entrada es proporcionado el nombre del procedimiento a ejecutar, por ejemplo DOS D1 D2. El sistema transfiere esta línea al árbol de funciones en el primer punto, tratándose de instanciar DOS=UNO lo que conduce a una falla; por lo tanto, se pasa a la segunda comparación en donde la instanciación tiene éxito lo que dispara la función DOS. En ella se detectan, tanto los argumentos, como su validez. Si todo está en orden, el procedimiento es ejecutado y el sistema retorna al punto de partida. En caso contrario se comunica un error de sintaxis (o cualquier otro, según sea el caso) y se retorna al inicio.

De lo anterior se desprende el carácter no recursivo de este analizador, es decir, resulta inválido anidar comandos para realizar funciones específicas. Sin embargo, cuando se trata con expresiones matemáticas, la recursividad toma un papel preponderante por lo que resulta de suma importancia considerar este aspecto.

Existen muchas formas de analizar sentencias matemáticas por computadora. Entre ellas se encuentran los analizadores conducidos por tabla y los descendentes recursivos. Los primeros son de poco uso en la actualidad debido a que, como su nombre lo indica, emplean tablas para conducción de operaciones, resultando en algunos casos tan complejas, que casi se requiere de un programa para generarlas.

En el segundo caso, empleado en ORFEO-90, el analizador es mucho más simple en cuanto a su concepción. Esto se debe a que trabaja basándose directamente en las estrictas reglas del álgebra, además de utilizar el mismo tipo de razonamiento que las personas suelen seguir al analizar y evaluar expresiones matemáticas.

E.4 ANALIZADOR DESCENDENTE RECURSIVO

En el contexto de este tipo de analizadores, las expresiones matemáticas se definen a partir de estructuras recursivas, es decir, estructuras que se definen a sí mismas.

EXPRESION \longrightarrow TERMINO [+TERMINO][-TERMINO]
 TERMINO \longrightarrow FACTOR [*FACTOR][^FACTOR]
 FACTOR \longrightarrow VARIABLE, NUMERO, FUNCION(EXPRESION) O EXPRESION

donde los nombres entre paréntesis rectangulares son opcionales, y el símbolo \longrightarrow significa produce.

De esta manera, podemos decir que una expresión es producida por un término o por la suma algebraica de dos o más términos. Un término está compuesto por un factor multiplicado, dividido o elevado a un factor. Un factor es producido por una variable, un número, una expresión o una función de una expresión.

A estas reglas se les llama *reglas de producción*. Es importante notar que la jerarquización de los operadores +, -, *, /, ^, así como funciones matemáticas, queda implícita en la definición de la expresión.

Para aclarar ideas consideremos la expresión:

$$A + 5 * (B + 8)$$

De acuerdo con lo anterior, la expresión está formada por dos términos: $A + 5 * (B + 8)$. El primero de ellos, está formado por un factor A y el segundo por los factores 5 y $(B + 8)$, el último de los cuales es producido recursivamente por una expresión con dos términos, B y 8 , de un factor cada uno.

A estas alturas, seguramente se preguntará ¿cómo se implanta un motor de inferencia de este tipo en una computadora?; pues bien, en lo subsecuente trataremos de dar respuesta a ello.

E.5 TECNICAS DE PROGRAMACION PARA ANALIZADORES DESCENDENTES RECURSIVOS

Volviendo a nuestra expresión $A + 5 * (B + 8)$, podemos distinguir una serie de componentes atómicos; A, +, 5, *, (, B, +, 8,). A cada una de estas unidades indivisibles se le llama *token*. De esto se desprende la existencia de varios tipos de token. En la programación del analizador de ORFEO-90 se tomó la siguiente clasificación:

TIPO DE TOKEN	COMPONENTES DEL TOKEN
DELIMITADOR	()+-*\/^=
VARIABLE	ABCDEFGHIJKLMNPOQRSTUVWXYZ
NUMERO	0123456789.E+- (notación científica)
FUNCION	Secuencia de dos o más caracteres.

Por lo anterior, se infiere la necesidad de una primera función que obtenga el token y lo clasifique de acuerdo con su tipo. Además, es necesario que ésta sea lo suficientemente "inteligente" para saltar los espacios en blanco comúnmente introducidos en una expresión para imprimir claridad en su sintaxis.

Ahora bien, para introducir la filosofía de la definición de expresiones al programa, es necesario elaborar una serie de funciones mutuamente recursivas encadenadas de acuerdo con la importancia de las operaciones. El encadenamiento debe estar dado de menor a mayor jerarquía, de tal manera que sea lanzada una expresión desde el punto de entrada hasta el nivel más alto, para que a su retorno ejecute las operaciones definidas atendiendo al riguroso orden establecido por el álgebra. Las funciones que contemplan el orden requerido son:

- EXPRESION - Función que dispara la secuencia de niveles.
- NIVEL 1 - Orientada a la asignación del valor de una expresión a una variable.
- NIVEL 2 - Cumple con la detección y ejecución de la suma o resta de dos términos.
- NIVEL 3 - Detecta y ejecuta la multiplicación o división de dos factores.
- NIVEL 4 - Detecta y ejecuta la exponenciación de un factor.
- NIVEL 5 - Aplica el operador monario + ó - (signo de un número).
- NIVEL 6 - Detección de paréntesis.
- PRIMITIVA - Función que devuelve los valores últimos de un token de acuerdo con su tipo; VARIABLE, NUMERO o FUNCION.

Finalmente, se requiere de una función que efectúe las operaciones aritméticas. Sus argumentos deben ser dos operandos y un operador, y su respuesta el resultado de ejecutarlos.

Con lo anterior es posible construir el esquema básico de un analizador descendente recursivo en pseudocódigo, tal y como se presenta a continuación.

- CONSIDERACIONES

FUNCIONES PRIMARIAS:

OBTENER TOKEN Función que obtiene el token (símbolo) y tipo dentro de una expresión matemática.

ARITMETICA(N1, N2, OPERACION)

Función que regresa el valor obtenido de ejecutar (N1) (OPERACION) (N2).

VARIABLES:

TOKEN Variable global. Almacena el token obtenido en la última llamada a la función **OBTENER TOKEN**.

TIPO Variable global. Define el tipo de token almacenado en **TOKEN** de acuerdo a la clasificación antes discutida.

RESULTADO Almacena el resultado de la evaluación de la expresión.

MANTENER Almacena el valor del término, factor o expresión siguiente al calculado hasta el momento para aplicar un operador o función determinada.

Las funciones reciben el valor de los argumentos por referencia; es decir, reciben la dirección de memoria en donde se almacena el valor, para poderlo alterar sin ningún problema.

Salvo en el caso de **TOKEN** y **TIPO**, todas las variables son locales, por lo que se generan tantas como veces sea llamada una función, y se destruyen al abandonar ésta. Se considera que cada llamada a la función **OBTENER TOKEN** actualiza las variables **TOKEN** y **TIPO**. Así mismo, la función **ARITMETICA**, efectúa el cálculo entre **RESULTADO** y **MANTENER** con el operador definido, asignando el valor obtenido a **RESULTADO**.

FUNCIONES EN SEUDOCODIGO

```
EXPRESION (RESULTADO)
{
  OBTENER TOKEN
  NIVEL 1(RESULTADO)
}
```

```
NIVEL 1 (RESULTADO)
{
  SI EXISTE ASIGNACION(
    NIVEL 2 (RESULTADO)
    ASIGNAR EL RESULTADO A LA VARIABLE
  )
  NIVEL 2 (RESULTADO)
}
```

NIVEL 2 (RESULTADO)

```
{ DEFINIR MANTENER, OP
  NIVEL 3 (RESULTADO)
  OP = TOKEN
  MIENTRAS QUE TOKEN = + 0 - {
    OBTENER TOKEN
    NIVEL 3 (MANTENER)
    ARITMETICA (RESULTADO, MANTENER, OP)
  }
}
```

NIVEL 3 (RESULTADO)

```
{ DEFINIR MANTENER, OP
  NIVEL 4 (RESULTADO)
  OP = TOKEN
  MIENTRAS QUE TOKEN = * 6 / {
    OBTENER TOKEN
    NIVEL 4 (MANTENER)
    ARITMETICA (RESULTADO, MANTENER, OP)
  }
}
```

NIVEL 4 (RESULTADO)

```
{ DEFINIR MANTENER
  NIVEL 5 (RESULTADO)
  SI TOKEN = ^ {
    NIVEL 4 (MANTENER)
    ARITMETICA (RESULTADO, MANTENER, ^)
  }
}
```

NIVEL 5 (RESULTADO)

```
{ DEFINIR OP = 1
  SI TIPO = DELIMITADOR Y TOKEN = + 6 - {
    OP = TOKEN
    OBTENER TOKEN
  }
  NIVEL 6 (RESULTADO)
  SI OP <> 1 → APLICAR OPERADOR MONARIO
}
```

NIVEL 6 (RESULTADO)

```
{ SI TOKEN = ( {
  OBTENER TOKEN
  NIVEL 1 (RESULTADO)
  SI TOKEN <> ) { REPORTAR ERROR DE SINTAXIS }
  OBTENER TOKEN
}
DE LO CONTRARIO { PRIMITIVA (RESULTADO) }
}
```

PRIMITIVA (RESULTADO)

```
{ DEFINIR MANTENER
  SI TIPO = FUNCION {
    OBTENER TOKEN
    NIVEL 6 (MANTENER)
    RESULTADO = FUNCION (MANTENER)
  }
}
```

```

        RETORNAR
    }
SI TIPO = NUMERO {
    RESULTADO = VALOR DEL NUMERO
    OBTENER TOKEN
    RETORNAR
}
SI TIPO = VARIABLE {
    RESULTADO = VALOR DE LA VARIABLE
    OBTENER TOKEN
    RETORNAR
}
REPORTAR ERROR DE SINTAXIS
}

```

Para comprender el proceso de evaluación, apliquémoslo paso a paso a nuestra expresión $A + 5 * (B + 8)$, considerando que $A=1.0$ y $B=2.0$. Para imprimir la mayor claridad posible, consulte el mapa de trayectoria de la figura E.5.1 en el punto indicado entre paréntesis a lo largo de la explicación.

Al entrar al analizador por la función EXPRESION (1), se obtiene el primer token con lo que las variables globales se inicializan a $TOKEN = A$ y $TIPO = VARIABLE$. Bajo estas circunstancias se hace un recorrido a través de los seis niveles hasta la función PRIMITIVA (2), la cual asigna el valor de A; es decir 1.0, a la variable RESULTADO. Así mismo, se hace una llamada a la función OBTENER TOKEN renovándose las variables $TOKEN = +$ y $TIPO = DELIMITADOR$. Al retornar de la función PRIMITIVA, ninguna de las condiciones es cumplida sino hasta regresar a la función NIVEL 2 (3). En este punto se define la operación suma, se obtiene el siguiente token, $TOKEN = 5$ y $TIPO = NUMERO$, y se lanza la función NIVEL 3 con el argumento MANTENER. De esta manera se sube nuevamente a través de todos los niveles hasta la función PRIMITIVA (4) en donde MANTENER toma el valor de 5, $TOKEN = *$ y $TIPO = DELIMITADOR$. El camino de retorno hasta la función NIVEL 2 se ve interrumpido al tener éxito la comparación efectuada en NIVEL 3 (5) en donde se define la operación multiplicación. Nuevamente es llamada la función OBTENER TOKEN que actualiza $TOKEN = ($ y $TIPO = DELIMITADOR$. Ahora la función NIVEL 3 llama a la función NIVEL 4 con el argumento MANTENER. Con los valores actuales de las variables globales se sube hasta la función NIVEL 6 (6) en donde la comparación tiene éxito. De esta forma, se vuelve a modificar el contenido de $TOKEN$ y $TIPO$ a B y VARIABLE, respectivamente. En este punto se lanza nuevamente la función NIVEL 1 (7) con el argumento "RESULTADO" pasado a NIVEL 6; es decir, MANTENER de la función NIVEL 3, y comienza una nueva escalada a través del analizador hasta la función PRIMITIVA (8). De manera análoga, la variable MANTENER toma el valor de $B=2.0$ y las variables globales se modifican a $TOKEN = +$ y $TIPO = DELIMITADOR$. Al descender por cada uno de los niveles se satisface la comparación de la función NIVEL 2 (9). Se define la operación suma y se obtiene el siguiente token, actualizándose $TOKEN = 8$ y $TIPO = NUMERO$. Subiendo nuevamente hasta la función PRIMITIVA (10), la variable MANTENER de la última llamada a NIVEL 2, toma el valor de 8, se alteran las variables globales mediante la función OBTENER TOKEN a $TOKEN =)$ y $TIPO = DELIMITADOR$ y se empieza a descender a través de los niveles recursivos generados por esta expresión hasta la función NIVEL 2 (11) en donde se ejecuta la suma entre 2 y 8. El resultado es devuelto al NIVEL 6 (12) en donde se revisa un posible error de sintaxis por paréntesis incompletos, y luego al NIVEL 3 (13) donde se ejecuta el producto entre 5 y 10. Con 50 como resultado asignado a MANTENER continúa el descenso hasta NIVEL 2 (14) de la primera

llamada efectuándose la última operación entre RESULTADO = 1 y MANTENER, tomando así RESULTADO el valor 51 y retornándolo a la función EXPRESION (15).

$$\text{EXPRESION } A + 5 * (B + 8)$$

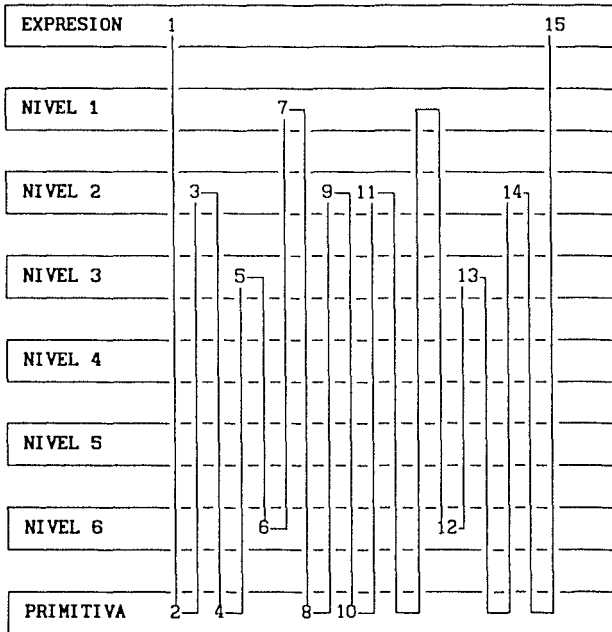


Figura E.5.1 Mapa de trayectoria en los niveles de recursividad.

E.6. MANEJO DE LA MEMORIA

Uno de los aspectos que requiere de especial cuidado en la elaboración de programas de cómputo es, sin duda alguna, el manejo óptimo de memoria. En esta tarea se involucra, aunque parezca extraño, la velocidad de procesamiento. Para entender claramente esta relación, es necesario hacer una breve explicación, tanto de la arquitectura de las computadoras personales, como de la flexibilidad que ofrece el lenguaje C, en cuanto a manejo de memoria se refiere.

Como es sabido, las computadoras trabajan con sistema binario. A cada posible estado, 1 ó 0, se le llama *bit*, y a cada grupo de ocho bits se le llama *byte*. De esta manera, el rango de valores que puede tomar un byte es de 00000000 a 11111111 en binario, de 0 a 255 en decimal o de 00 a FF en hexadecimal. Cada uno de estos bytes ordenados en una secuencia determinada, define el código y los datos empleados en todo programa. Por esto, resulta lógico pensar que el rango de cada uno de los tipos de datos, depende de la cantidad de bytes empleados en su definición.

El lenguaje C permite el manejo de cuatro tipos de datos (*char*, *int*, *float* y *double*) cada uno de los cuales acepta cuatro modificadores (*signed*, *unsigned*, *long* y *short*).

TIPO	LONGITUD EN BITS	RANGO
<i>char</i>	8	Caracteres ASCII
<i>unsigned char</i>	8	0 a 255
<i>signed char</i>	8	-128 a 127
<i>int</i>	16	-32768 a 32767
<i>unsigned int</i>	16	0 a 65535
<i>signed int</i>	16	igual que <i>int</i>
<i>short int</i>	8	-128 a 127
<i>unsigned short int</i>	8	0 a 255
<i>signed short int</i>	8	igual que <i>short int</i>
<i>long int</i>	32	-2147483648 a 2147483649
<i>signed long int</i>	32	igual que <i>long int</i>
<i>unsigned long int</i>	32	0 a 4294967296
<i>float</i>	32	6 dígitos de precisión
<i>double</i>	64	12 dígitos de precisión
<i>long double</i>	128	24 dígitos de precisión

En el caso de los datos afectados por el modificador *signed*, el llamado *bit mas significativo*, situado en la extrema izquierda representa el signo mediante la convención 1 negativo 0 positivo.

Con lo anterior es posible inferir el primer paso en la optimización de memoria; seleccionar adecuadamente el tipo de datos que se van a emplear. Cada byte almacenado en la memoria de la PC recibe una *etiqueta* que define su posición llamado *apuntador*. De esta forma, cada tipo de dato tiene un *apuntador específico*. Por ejemplo, consideremos la existencia de un arreglo de enteros *E[i]* y un *apuntador a entero* *Ap*. Mediante la sintaxis del lenguaje C es posible asignar la dirección de memoria del primer elemento del arreglo *E* (*E[0]*) al *apuntador* *Ap* de la siguiente manera.

$Ap = \&E[0]$

donde el operador & retorna la dirección de memoria de la variable. Dado que es posible la suma y resta de cantidades a los punteros, la expresión $Ap + 1$ (que también se puede escribir como $Ap++$ o $Ap+=1$) incrementará el apuntador en una unidad; es decir, 2 octetos debido a que un entero tiene 16 bits. Si el apuntador fuera de tipo float, el incremento sería de 4 octetos.

Ahora bien, los punteros están codificados de acuerdo con la arquitectura "segmentada" de las computadoras basadas en el procesador 8086 - 8088. El término segmentada se aclara en seguida.

El total de localidades de memoria disponibles en una computadora personal es de un megabyte (1048576 bytes). Para direccionar a estas posiciones se utilizan 2 octetos (16 bits). Aun y cuando estos bytes no tienen signo el número máximo de direcciones que se pueden etiquetar con 16 bits es de 65535. Entonces surge la pregunta inevitable, ¿ cómo acceder al resto de la memoria ?.

Los fabricantes de computadoras resolvieron el problema utilizando dos palabras de 16 bits. El *segmento* (segment) y el *desplazamiento* (offset) los cuales se representan en una unidad, de acuerdo con la sintaxis *segmento:desplazamiento*. El primero se comporta como los capítulos de un libro, avanzando en múltiplos de 16, y el segundo como las hojas. De esta manera es posible acceder a un total de $65535 * 16 + 15 = 1048575$ posiciones, es decir, al megabyte antes mencionado.

Por la metodología empleada en el direccionamiento de memoria se presenta una consecuencia inevitable. Una localidad puede ser referida por varias parejas *segmento:desplazamiento*.

Para ilustrar esta situación supongamos que nos interesa acceder a la dirección de memoria E000:A000 (las posiciones se manejan en base hexadecimal). Si calculamos la dirección absoluta de este punto obtendremos:

$$E000 (57344) * 10 (16) + A000 (40960) = (958464)$$

donde los números entre paréntesis indican la equivalencia en decimal.

Ahora bien, si consideramos la posición E1E1:81F0 y realizamos el mismo cálculo:

$$E1E1 (57825) * 10 (16) + 81F0 (33264) = (958464)$$

obtendremos la misma dirección absoluta.

Los compiladores de lenguaje C permiten definir la manera en la que se direccionarán las localidades de memoria para datos y código, atendiendo a los llamados " *Modelos de memoria* ". Existen seis modelos de memoria empleados en la compilación de programas escritos en lenguaje C; TINY, SMALL, MEDIUM, COMPACT, LARGE y HUGE. Cada uno de ellos impone ciertos límites en cuanto a la extensión del código, zonas de asignación dinámica y localización de variables se refiere.

Como se puede observar en la figura E.6.1 el modelo de memoria TINY ocupa en total 64 Kb, de donde se desprende que, tanto el código como los datos pueden ser direccionados con punteros (sinónimo de apuntador) de 16 bits, lo que se traduce en una mayor rapidez de ejecución y menor extensión de código. Este modelo de memoria fue empleado en la compilación de los programas ORFEO_A.COM, CNFORFEO.COM, PLIB.COM y DLIB.COM. Sólo utilizando este tipo de modelo es posible convertir, mediante el enlazador LINK o la utilería EXE2BIN del sistema operativo, los programas con extensión EXE a COM, cuya diferencia estriba en la rapidez con la que pueden ser cargados de disco a memoria

principal.

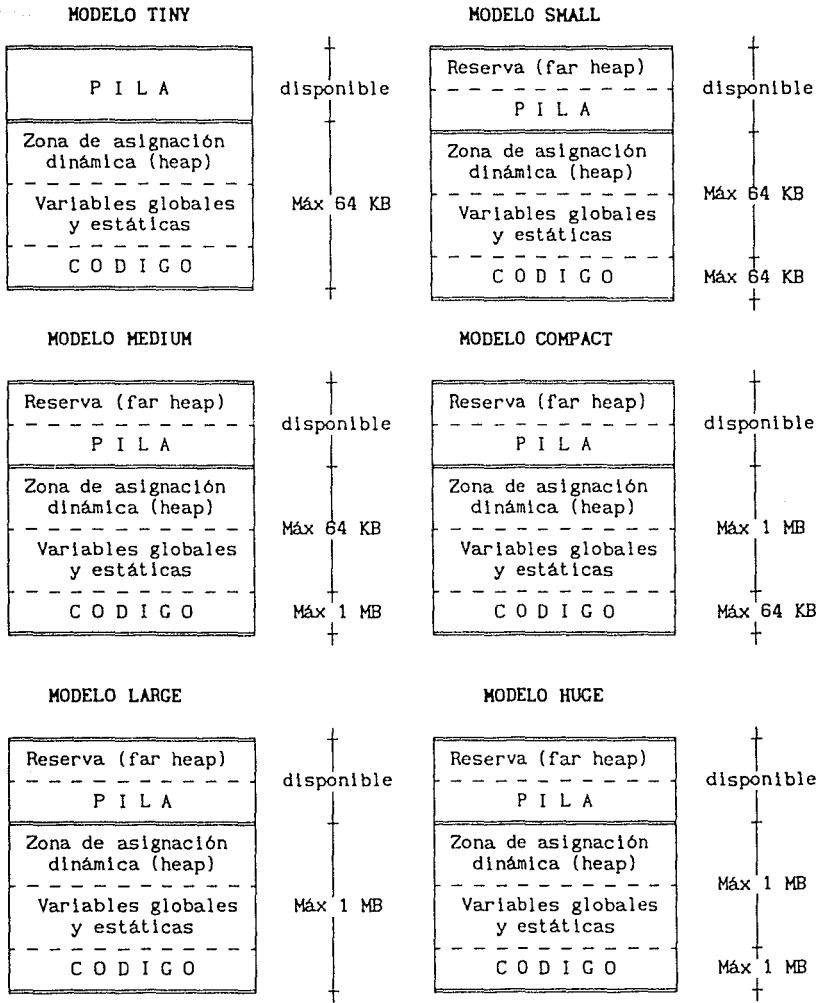


Figura E.6.1 Modelos de memoria disponibles en el compilador Turbo C de Borland.

El modelo de memoria SMALL fue utilizado para compilar los programas GRAFICAR.EXE, SISMO.EXE, BINARIO.EXE y ANIMA.EXE, que como se aprecia en la figura E.6.1 proporciona 64 Kb para cada una de las zonas destinadas a datos y código. La rapidez de ejecución no se ve afectada ya que se utilizan punteros de 16 bit pero iniciando en diferentes segmentos.

En los modelos TINY y SMALL los apuntadores destinados a los datos son los llamados near (cercanos) y esto es debido a que pueden recorrer todo su dominio con sólo 16 bits (máximo 64 Kb). Existen otros dos tipos de punteros: far (lejanos) y huge (enormes).

Estos apuntadores permiten acceder a localidades de memoria situadas por encima de la pila, debido a estar codificados en 32 bits. La diferencia fundamental entre ellos es que los punteros huge siempre trabajan con direcciones normalizadas, lo que produce un mayor código compilado y un substancial incremento en el tiempo de acceso a datos. Los punteros far son mucho más rápidos ya que no normalizan la dirección a la que apuntan cada vez que son gestionados; sin embargo, esto conduce a la imposibilidad de revisar límites. Es decir, no se sabe en que momento es superado el segmento de partida. Ejemplo, para un puntero tipo far:

$$E000:FFFF + 1 = E000:0000$$

Comúnmente los programadores de lenguajes de alto nivel como FORTRAN se desentienden de estos tópicos por resultarles completamente transparentes. Sin embargo, tomándolos en cuenta es posible reducir significativamente el tiempo de acceso a datos.

Existen algoritmos que permiten seccionar en bloques grandes problemas, por ejemplo, en la solución de sistemas de ecuaciones. Estos métodos no deben utilizarse cuando la cantidad de memoria disponible es rebasada sino cuando el apuntador de datos es insuficiente. Para aclarar ideas supongamos que un proceso de cálculo requiere 256 Kb de memoria pero sólo se cuenta con 128 Kb disponibles. Bajo estas circunstancias se podría pensar en particionar el problema en dos bloques de 128 Kb; sin embargo, esto no es óptimo, ya que para accederlos, sin revisar límites, sería necesario apuntadores de tipo huge que son extremadamente lentos. Mejor es utilizar cuatro bloques de 64 Kb y utilizar punteros de tipo far o inclusive near, incrementando en mucho la eficiencia del proceso.

ORFEO-90 utiliza esta técnica compilándolo en modelo COMPACT, manejando con punteros cercanos los datos matriciales y con apuntadores lejanos, tipo far, los datos definidos en el editor programado y procesos de cálculo interno.

Para asegurar que la dirección apuntada por un puntero far es realmente recorrible hasta 64 Kb es necesario prenormalizarla. Esta situación es importante. Supongamos que se hace una gestión de memoria lejana y es obtenido un puntero

$$Fp = 0000:FFFF$$

Aunque representa un ejemplo absurdo, el lector se percatará de que no es posible acceder a más de 15 localidades de memoria (FFFF a FFFF). La prenormalización llevará entonces al puntero

$$Fp = 0FFF:0000$$

que sí permite el acceso completo a los 64 Kb (0000 a FFFF) sin requerir

revisión de límites.

A continuación se proporciona el código en lenguaje C para este propósito.

```
void far * normalizar_puntero( void far *apuntador)
{
    unsigned int segmento, desplazamiento;
    segmento = ((unsigned) ((unsigned long)(apuntador) >> 16));
    desplazamiento = ((unsigned) (apuntador));

    segmento += desplazamiento<<4;
    desplazamiento %= 16;

    return( (((unsigned long)(segmento)<<16)|(unsigned)(desplazamiento)) )
}
```

IMPORTANTE: Debido a la normalización de apuntadores, existe la posibilidad de mermar en 15 bits el dominio de los punteros far. Es responsabilidad del programador tomar las precauciones pertinentes.

E.7 ESTRUCTURAS DE DATOS

Cuando se diseñan programas con entidades de datos referidos por un mismo nombre, el uso de estructuras de datos es de primordial importancia. El lenguaje C permite realizar estructuras de datos complejas y de muy alta eficiencia. En el caso de ORFEO-90 se utilizaron las siguientes estructuras para manejo de matrices.

```
union PMAT {
    int      *e;
    double   *d;
    char     *t;
}

type struct {
    char nombre[5];
    char t;
    int r,c;
    union PMAT p;
} MAT;

MAT matriz[MAXMAT];
```

De esta manera se definieron **MAXMAT** (50) estructuras **matriz** con los atributos **nombre[5]** (nombre del arreglo), **t** (tipo de matriz), **r** (número de renglones), **c** (número de columnas) y unión **PMAT p** (apuntadores a distintos tipos de datos referidos en la misma localidad de memoria).

Con lo anterior es posible manejar el total de datos que definen a un arreglo bajo el sólo nombre genérico de **matriz**.

Ejemplo:

La definición del arreglo 24 dentro de ORFEO-90 podría estar dado por:

```
matriz[24].nombre = "U(t)"
matriz[24].t      = 1      (tipo real)
matriz[24].r      = 20     (20 renglones)
matriz[24].c      = 2      (2 columnas)
matriz[24].p.d    = AB01   (apuntador cercano al primer elemento)
```

lo cual es interpretado internamente por el programa como matriz **U(t)** de orden 20 x 2 de tipo real.

E.8 ALMACENAMIENTO Y REFERENCIA DE DATOS MATRICIALES EN VECTORES

Cuando se utilizan algoritmos que operan con arreglos, a menudo se emplea una notación índice para facilitar su representación. Comúnmente esta nomenclatura toma la concepción representativa que tenemos de una matriz; una serie de renglones y columnas manejadas por coordenadas. Las computadoras evidentemente no operan así. Para ellas todos los datos se encuentran secuencialmente almacenados en un vector.

Con base en lo anterior, y sobre todo en el concepto del "manejo dinámico de memoria", es frecuente encontrar algoritmos para diversas tareas convertidos en procedimientos vectoriales. Esto resulta, en la mayoría de los

casos un trabajo inútil debido a la complejidad que encierra esta modificación y la difícil interpretación del código generado. En lugar de esto, la alternativa se encuentra en una interfase de gestión y colocación de datos matriciales mediante sencillos algoritmos de ubicación sobre vectores y emplear los métodos de cálculo con su notación índice tal y como fueron concebidos.

Además, esta técnica facilita la optimación de memoria si se sabe a priori el tipo de matrices que se utilizarán.

Considere que R es el número de renglones de un arreglo y C el número de columnas. De esta manera la cantidad de memoria (expresada en unidades de tipo de dato) de acuerdo con el tipo de matriz es:

Matriz normal	$R \times C$
Matriz simétrica	$R (R + 1) / 2$
Matriz diagonal	R
Matriz identidad	1

Así, para almacenar o recuperar información posicionada en una matriz a partir de los índices i, j y su tipo se utilizan los siguientes algoritmos en pseudocódigo:

donde

R	número de renglones
C	número de columnas
i	renglón gestionado ($i[0, R-1]$)
j	columna gestionada ($j[0, C-1]$)
V[]	vector que almacena a la matriz
valor	valor del elemento i, j

FUNCIONES DE COLOCACION

```
COLOCA_NORMAL( valor, i, j )
{
    V[ C * i + j ] = valor;
    RETORNAR
}
```

```
COLOCA_SIMETRICA( valor, i, j )
{
    SI( i < j ) {
        aux = j
        j = i
        i = aux
    }

    V[ R * i + j - ( i * ( i + 1 ) / 2 ) ] = valor
    RETORNAR
}
```

```

COLOCA_DIAGONAL( valor, i, j )
{
    SI( i < > j ) RETORNAR
    V[ i ] = valor
    RETORNAR
}

```

FUNCIONES DE GESTION

```

GESTION_NORMAL( i, j )
{
    RETORNAR ( V[ C * i + j ] )
}

```

```

GESTION_SIMETRICA( i, j )
{
    SI( i < j ) {
        aux = j
        j = i
        i = aux
    }
    RETORNAR( V[ R * i + j - ( i * ( i + 1 ) / 2 ) ] )
}

```

```

GESTION_DIAGONAL( i, j )
{
    SI( i < > j ) RETORNAR( 0.0 )
    RETORNAR( V[ i ] )
}

```

```

GESTION_IDENTIDAD( i, j )
{
    SI( i = j ) RETORNAR( 1.0 )
    RETORNAR( 0.0 )
}

```