Capítulo 6
Implementación de UNDIN. Descripción de la subrutina principal

En este capítulo se da una descripción general de UNDIN, programa didáctico a código abierto de análisis dinámico de estructuras desarrollado en esta tesis, utilizando las metodologías de cálculo descritas en los capítulos precedentes.

En la primera parte del capítulo se indica cómo funciona el programa, sus posibilidades, limitaciones, así como la forma de acceder al código principal y a las subrutinas. También se describe cómo se encuentra organizada la subrutina principal denominada *UNDIN*, y se describe en detalle cada etapa de cálculo que se lleva a cabo en esta subrutina.

6.1. Descripción general del programa

6.1.1. Contenido del programa

El programa desarrollado consiste en una macro de Microsoft Excel, compuesta por una subrutina principal denominada *"**UNDIN**"* escrita en el módulo **"*Md*",** y unas subrutinas secundarias que realizan tareas especificas y que están almacenadas en otros módulos, en los que se agrupan las subrutinas existentes de acuerdo a su uso (Linero, 2008). La subrutina pincipal funciona a partir de las subrutinas secundarias.

En el desarrollo del programa se han adicionado subrutinas nuevas a los módulos "***MdMT***", "***MdNGL***", "***MdTR***", "***MdIO***" y "***MdPB***", las cuales permiten manipular datos de matrices, numerar grados de libertad, importar y exportar matrices, entre otras tareas. También se han complementado algunas de las subrutinas originales de PEFiCA, almacenadas en los módulos "***MdK***", "***MdF***" y "***MdVP***". En estos módulos se almacenan respectivamente las subrutinas para crear matrices elementales de rigidez, para crear vectores elementales de fuerzas y para resolver el problema de valores propios, cuya solución, para

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

los métodos de análisis dinámico elástico, corresponde a las frecuencias y a los modos de vibración natural de la estructura. Para la construcción de la subrutina principal y de las subrutinas secundarias se ha utilizado un buen número de subrutinas existentes en la versión de PEFiCA tomada como base.

Adicionalmente se han creado los módulos "***MdAD***" y "***MdAE***", que contienen subrutinas para realizar cálculos de análisis dinámico y de análisis estructural respectivamente, desarrolladas tomando como base la formulación matemática presentada en los Capítulos 3 a 5 para los diferentes métodos de análisis dinámico, y con las que se realizan tareas específicas, como por ejemplo la creación de las matrices de masa, de amortiguamiento o de rigidez, condensación y asociación de grados de libertad, cálculo de acciones internas, etc.

6.1.2. Estructura de la subrutina principal

La subrutina principal ***"UNDIN"*** se encuentra dividida en siete etapas de cálculo, en las que se agrupan las instrucciones que permiten realizar tareas comunes tales como la lectura de datos o la ejecución de alguno de los tipos de análisis dinámico disponibles. La estructura de UNDIN se muestra el diagrama de flujo de la Figura 6.1

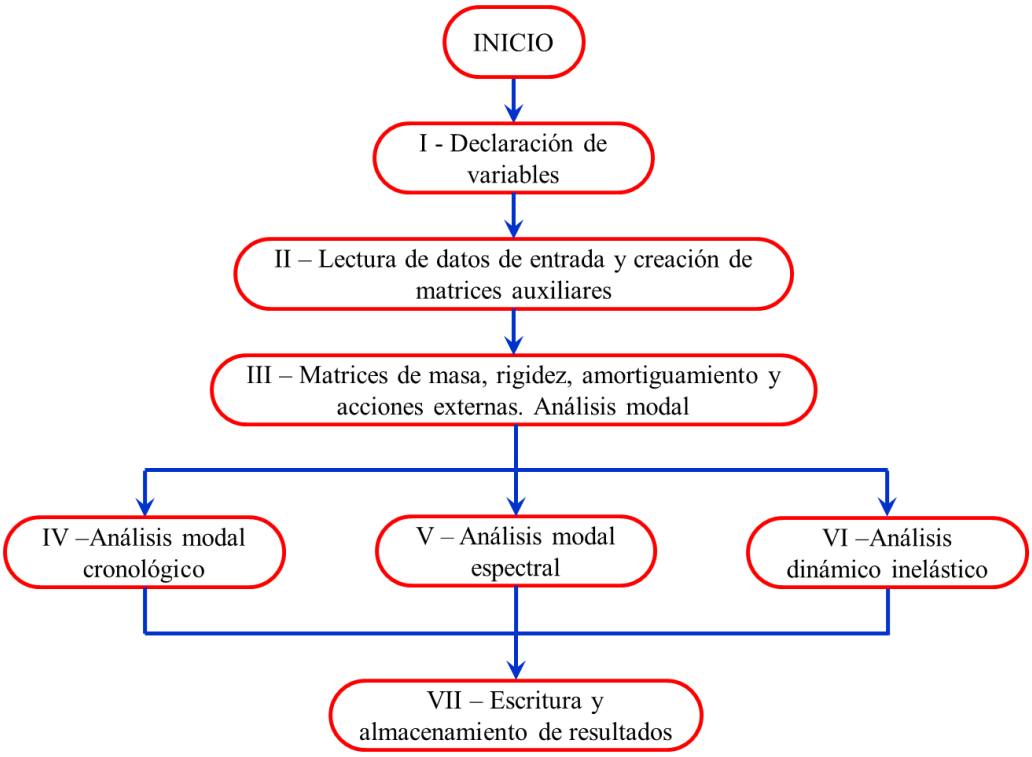


Figura 6.1. Estructura general del programa

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La parte I comprende la definición de las variables escalares y matriciales necesarias para la ejecución de la subrutina principal, así como la escritura del encabezado y la limpieza de la hoja en la que se escriben los resultados más relevantes del análisis.

El diagrama de flujo de la Figura 6.2 muestra las etapas de cálculo previstas en la parte de declaración de variables de la subrutina principal.

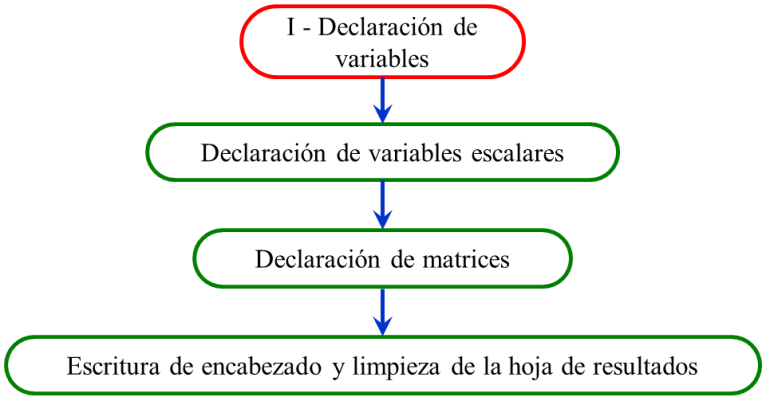


Figura 6.2. Parte I – declaración de variables

En la Parte II se realiza la lectura de los datos de entrada, que pueden ser de tipo escalar o matricial. Estos se leen desde las diferentes hojas de datos dispuestas para tal fin.

Adicionalmente en esta parte se definen algunos parámetros escalares relacionados con las cantidades de grados de libertad, y se crean unas tablas auxiliares que se utilizan en las etapas siguientes para facilitar el funcionamiento de otras subrutinas.

El diagrama de flujo de la Figura 6.3 muestra las etapas de cálculo previstas en la parte de lectura de datos de entrada y creación de matrices auxiliares.

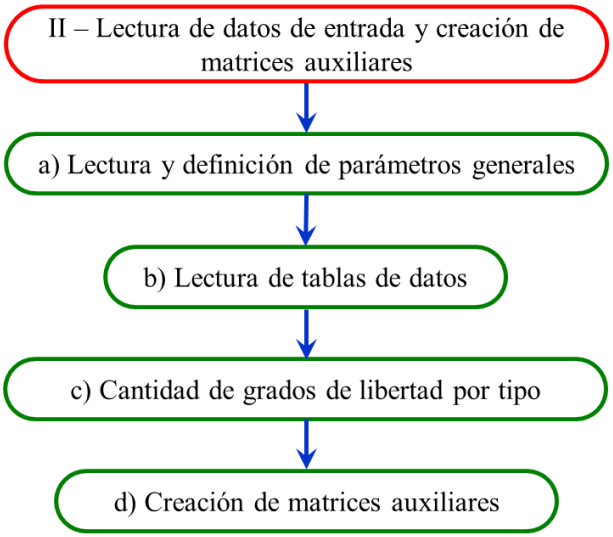


Figura 6.3. Parte II – lectura de datos de entrada y creación de matrices auxiliares

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

En la Parte III se crean las matrices de masa, rigidez, amortiguamiento y acciones externas equivalentes a la aceleración de la base, y se realiza el análisis modal de la estructura, del que se obtienen las frecuencias y los modos de vibración natural calculados por el método de Jacobi descrito en el Capítulo 3. Estas cantidades son necesarias para calcular la respuesta de la estructura por alguno de los métodos de análisis dinámico elástico descritos en el Capítulo 4.

En esta parte también se calculan otras cantidades de interés, útiles para entender el comportamiento y anticiparse a la respuesta de una estructura en el rango elástico, tales como los factores de participación y la masa modal efectiva.

El diagrama de flujo de la Figura 6.4 muestra las etapas de cálculo previstas para la creación de las matrices de masa, rigidez y amortiguamiento y para llevar a cabo el análisis modal de la estructura.

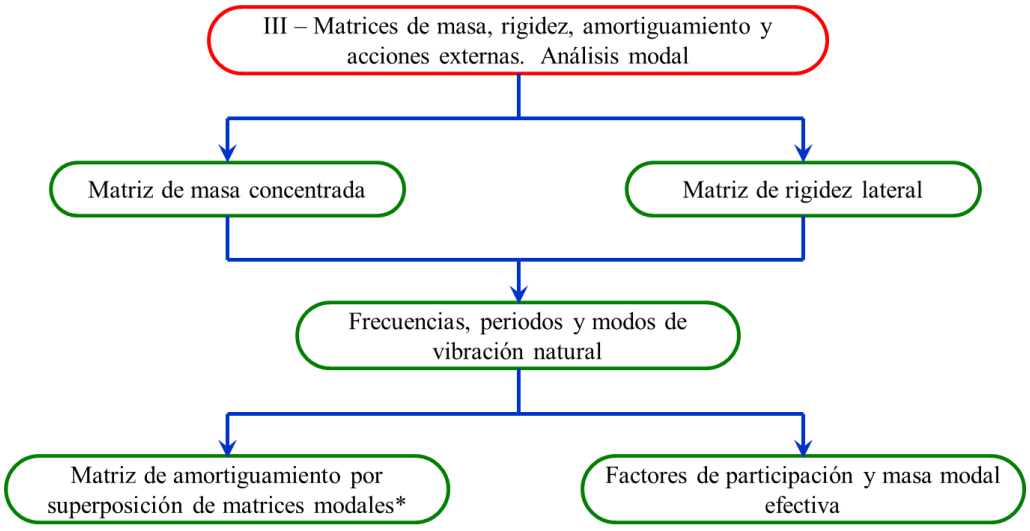


Figura 6.4. Parte III – Matrices de masa, rigidez y amortiguamiento & Análisis modal

En la Parte IV se calcula la respuesta de una estructura en rango elástico, con amortiguamiento clásico y sometida a una aceleración en la base, utilizando el método de análisis modal cronológico, el cual tiene como base los resultados del análisis modal, específicamente los modos de vibración natural.

En esta etapa se obtienen varios parámetros de respuesta que son de interés para entender el comportamiento y/o para realizar el diseño de la estructura, tales como los desplazamientos de los centros de masa, las derivas o las acciones internas en los elementos. También se verifican las ecuaciones de movimiento y de balance de energía. Específicamente el programa calcula la variación de cada parámetro en el intervalo de tiempo especificado.

El diagrama de flujo de la Figura 6.5 muestra las etapas de cálculo que considera UNDIN para calcular la respuesta de una estructura plana o espacial, utilizando el método de análisis modal cronológico.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

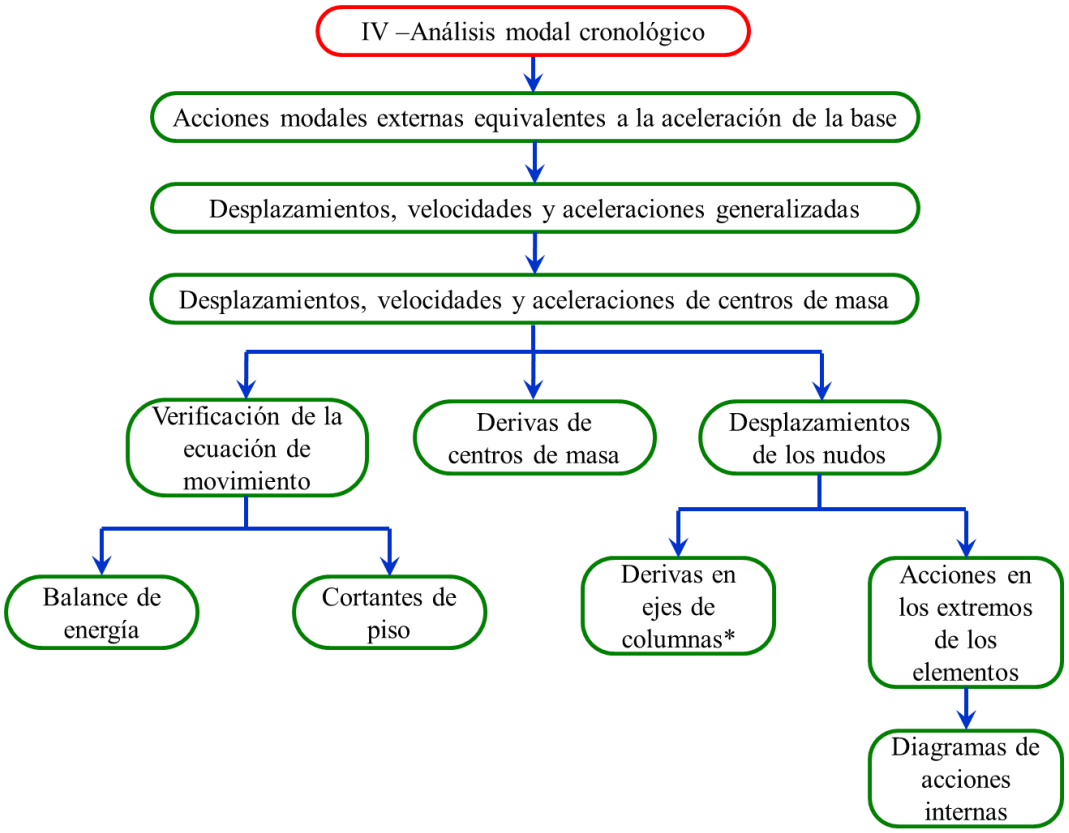


Figura 6.5. Parte IV – Análisis modal cronológico

En la Parte V se calcula la respuesta máxima probable de una estructura en rango elástico y con amortiguamiento clásico, a partir de un espectro elástico de pseudo – aceleraciones, en el que se resume el efecto máximo de una sola componente de aceleración en la base sobre todos los posibles sistemas de un grado de libertad dinámico. Se utiliza el método de análisis modal espectral, que también tiene tiene base el uso de los modos de vibración natural.

En esta etapa se obtienen los valores máximos probables de los mismos parámetros de respuesta estructural calculados por análisis modal cronológico (*desplazamientos de centros de masa, derivas, acciones internas en los elementos, etc.*) pero no se verifican las ecuaciones de movimiento ni de balance de energía. UNDIN calcula un solo valor máximo probable y sin signo para cada parámetro de respuesta.

El diagrama de flujo de la Figura 6.6 muestra las etapas de cálculo previstas en UNDIN para estimar la respuesta máxima de una estructura plana o espacial, utilizando el método de análisis modal espectral.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

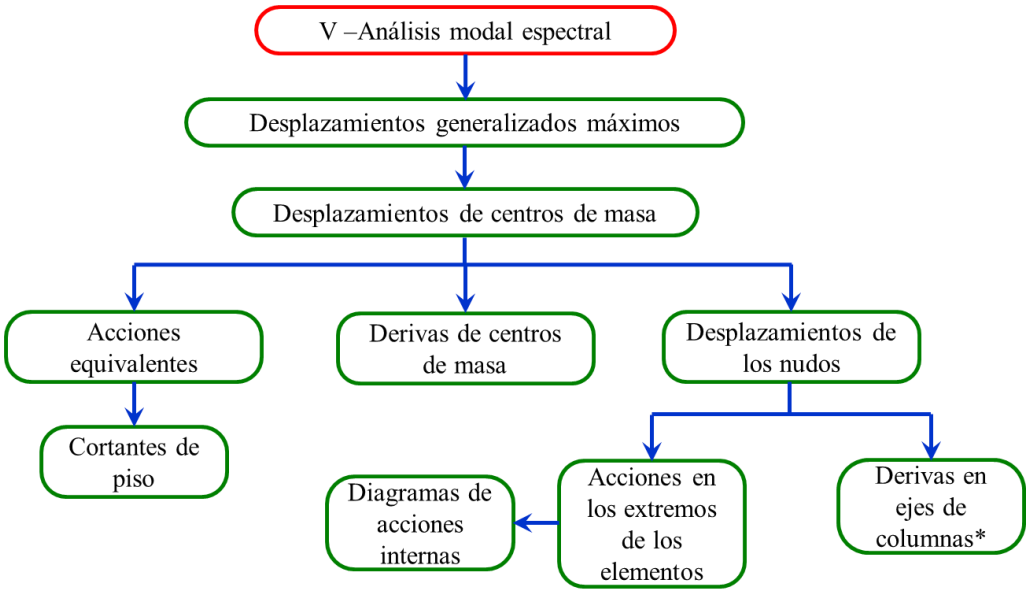


Figura 6.6. Parte V – Análisis modal espectral

En la Parte VI se calcula la respuesta de una estructura en rango inelástico, con amortiguamiento proporcional a la masa y a la rigidez inicial o a la rigidez tangente (*Amortiguamiento de Rayleigh*) y sometida a una o varias componentes de aceleración en la base, utilizando el método de análisis dinámico inelástico. En este método ya no se usan los resultados del análisis modal. Por el contrario las ecuaciones de movimiento se resuelven en su forma básica utilizando el método de Newmark combinado con el método de Newton – Raphson modificado, siguiendo la formulación matemática descrita en el Capítulo 5.

Unicamente se considera la no linealidad material producida por la formación de rótulas plásticas a flexión en los extremos de los elementos. Las rótulas plásticas se idealizan mediante resortes rotacionales de rigidez variable, cuya relación $M - \theta$ sigue la fomulacion matemática del modelo de Bouc – Wen descrito en el Capítulo 5. En esta versión de UNDIN no es posible considerar rótulas plásticas por fuerza axial, por fuerza cortante ni por flexo – compresión. Tampoco es posible considerar la no linealidad geométrica, dado que la formulación matemática utilizada se ha planteado desde la posición no deformada de la estructura.

En esta etapa del programa se calculan los mismos parámetros de respuesta estructural que se obtienen por análisis modal cronológico. También se verifican las ecuaciones de movimiento y de balance de energía, a esta última se adiciona la energía de deformación inelástica o disipada por histéresis. Nuevamente UNDIN calcula la variación de cada parámetro en el intervalo de tiempo especificado, el cual puede o no coincidir con el intervalo de tiempo en el que existe aceleración en la base.

El diagrama de flujo de la Figura 6.7 muestra las etapas previstas para calcular la respuesta de la estructura, utilizando el método de análisis dinámico inelástico.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

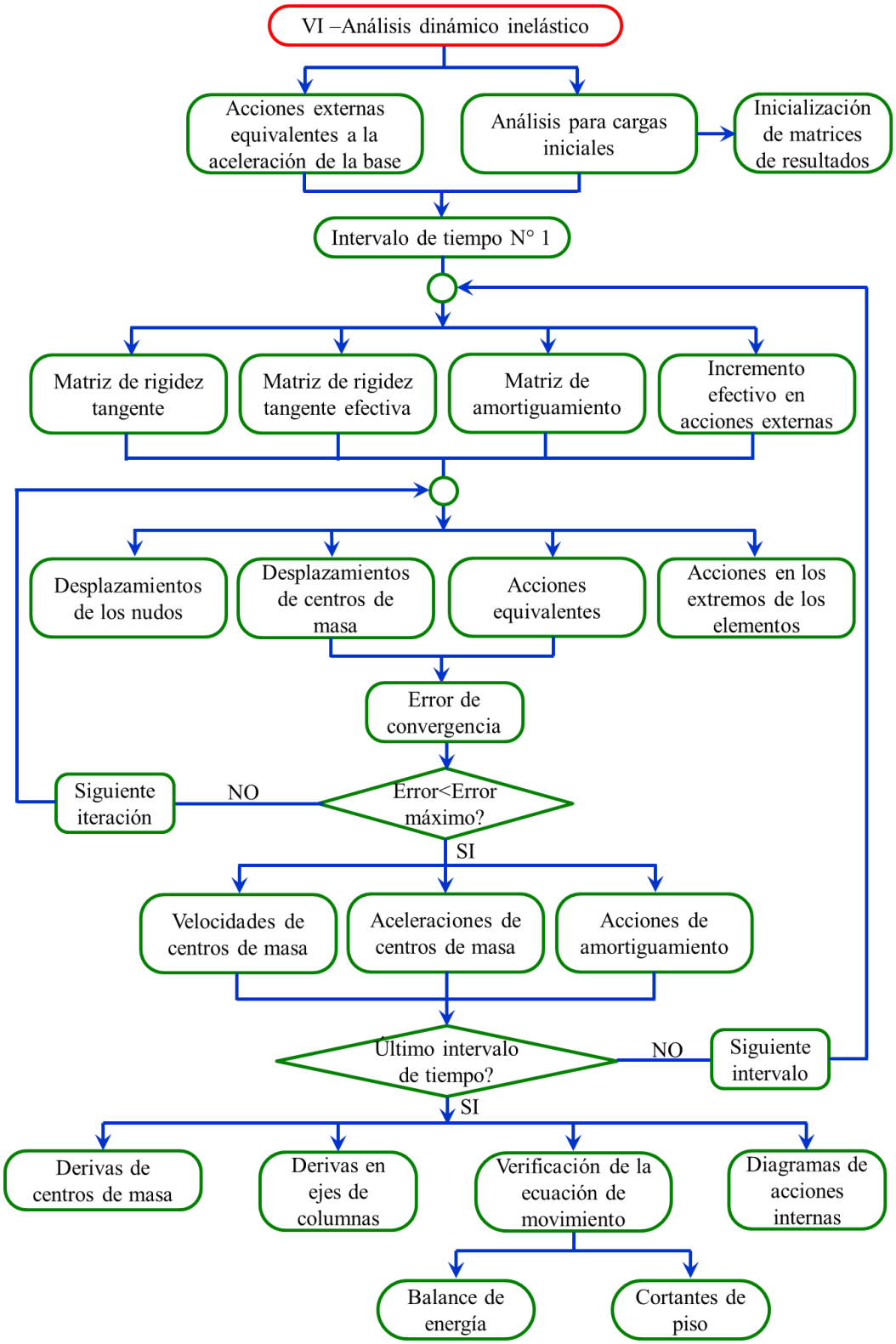


Figura 6.7. Parte VI –Análisis dinámico inelástico

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

Finalmente en la Parte VII se realiza la escritura, en la hoja de correspondiente, de los resultados más relevantes del análisis (*aunque es posible escribir cualquier cantidad deseada*), y se exporta a archivos de texto cualquier matriz de datos o de resultados, haciendo visible la totalidad de las variables involucradas en la solución del problema.

La escritura y el almacenamiento de los resultados obtenidos de los métodos de análisis modal cronológico y análisis dinámico inelástico, se agruparon en la misma etapa teniendo en cuenta la similitud de resultados que se producen con cada método. El diagrama de flujo de la Figura 6.8 muestra las etapas de cálculo previstas en esta parte de escritura y almacenamiento de resultados.

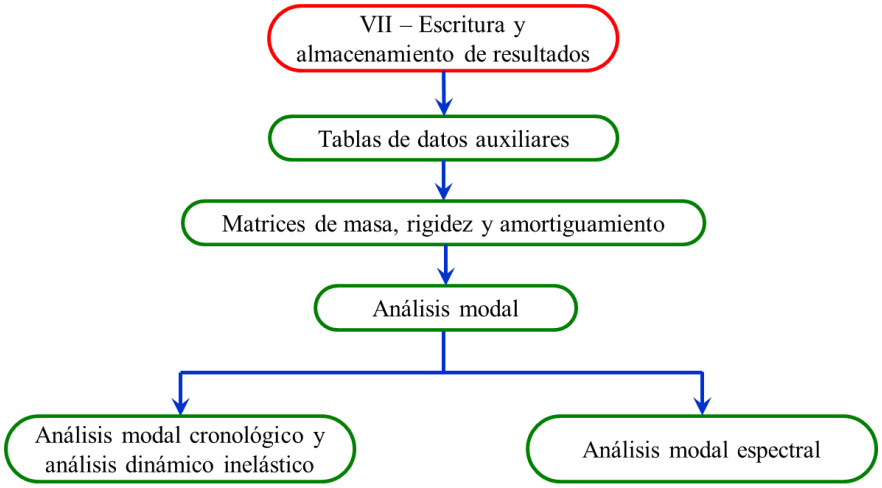


Figura 6.8. Parte VII – escritura y almacenamiento de resultados

Cada una de las partes que componen la subrutina principal UNDIN descritas previamente se detallan a continuación con una mayor profundidad, mencionando la forma en que trabajan las subrutinas secundarias involucradas en cada etapa especifica de cálculo.

6.1.3. Acceso a la subrutina principal y a las subrutinas secundarias

El acceso a la subrutina principal se hace según lo indicado en la Figura 6.9, seleccionando en la parte superior la pestaña "***Vista***", luego seleccionando la pestaña "***Macros***" y seleccionando la opción "***Ver Macros***", con lo que aparece un cuadro de dialogo en el que debe seleccionarse la macro "***UNDIN***" y finalmente debe seleccionarse la opción "***Modificar***". De esta forma se accede al *Editor de Visual Basic* (*VBA*) de Excel.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

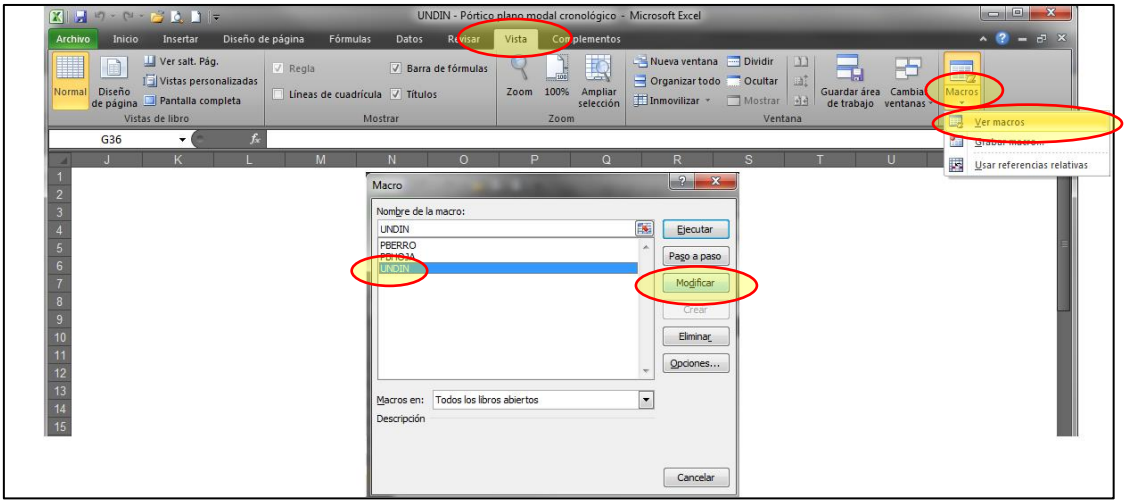


Figura 6.9. Acceso al editor de Visual Basic

Dentro de VBA se identifica, a la izquierda, la ventana denominada "*explorador de proyectos*" (Figura 6.10) en la que se listan las hojas de cálculo activas y los módulos donde se almacenan todas las subrutinas que componen el programa. Con doble clic sobre algún modulo se accede a las respectivas subrutinas.

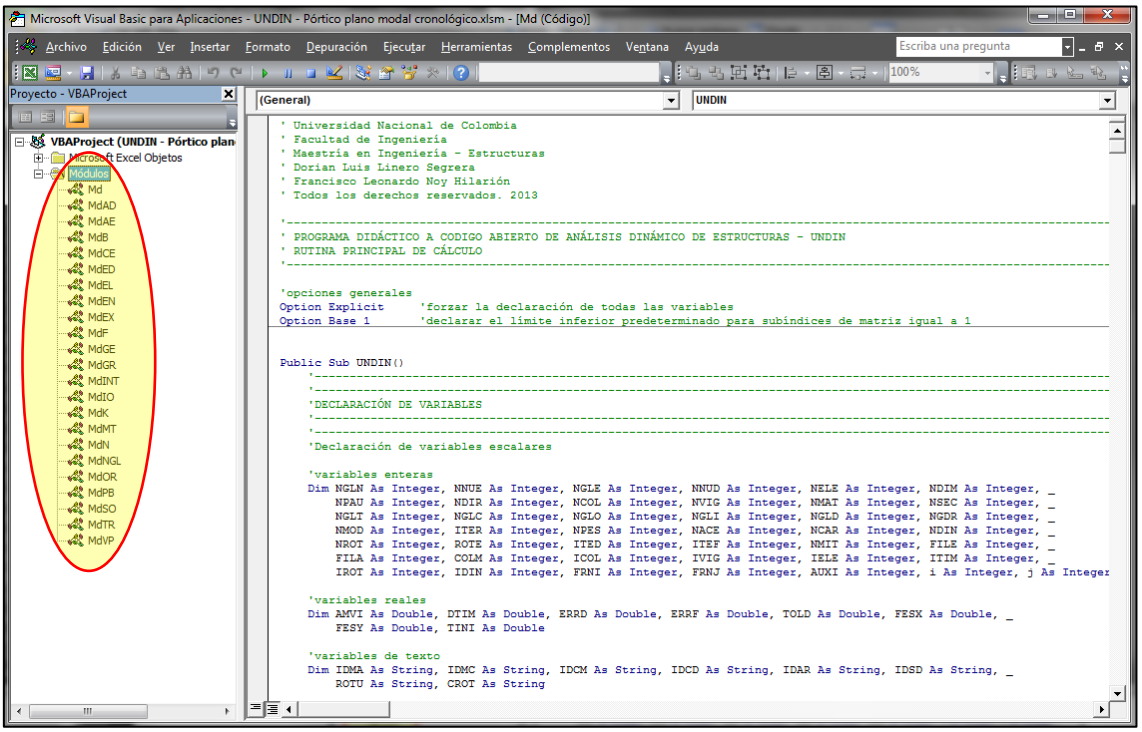


Figura 6.10. Acceso a las subrutinas

6.2. Parte I: Declaración de variables

A continuación se da una descripción general de cada una de las variables declaradas en la subrutina principal UNDIN.

6.2.1. Declaración de variables escalares

Las variables escalares declaradas se utilizaron para almacenar un solo valor de tipo entero, real o de texto.

En el código principal se requirieron 59 variables escalares, que se usan constantemente en las diferentes etapas de cálculo. Se utilizaron 43 variables escalares de tipo entero (*integer*), 9 variables escalares de tipo real de doble precisión (*double*) y 7 variables escalares tipo cadena (*string*).

En la Tabla 6.1 se indica el tipo y una descripción de cada una de las variables escalares declaradas para la ejecución de la subrutina principal.

N°	Nombre	Tipo	Descripción
1	NGLN	Entero	Número de grados de libertad por nudo
2	NNUE	Entero	Número de nudos por elemento
3	NGLE	Entero	Número de grados de libertad por elemento
4	NNUD	Entero	Número de nudos
5	NELE	Entero	Número de elementos
6	NDIM	Entero	Número de dimensiones de la estructura (2=plana, 3=espacial)
7	NPAU	Entero	Número de puntos auxiliares para definir ejes locales de elementos
8	NDIR	Entero	Número de diafragmas rígidos
9	NVIG	Entero	Número de ejes de vigas
10	NCOL	Entero	Número de ejes de columnas
11	NMAT	Entero	Número de materiales tipo
12	NSEC	Entero	Número de secciones
13	NGLT	Entero	Número de grados de libertad estáticos de la estructura
14	NGLC	Entero	Número de grados de libertad conocidos
15	NGLO	Entero	Número de grados de libertad desconocidos condensados
16	NGLI	Entero	Número de grados de libertad desconocidos de interés para el análisis dinámico
17	NGLD	Entero	Número de grados de libertad desconocidos
18	NGDR	Entero	Número de grados de libertad por diafragma rígido
19	NMOD	Entero	Número de modos de vibración de la estructura
20	ITER	Entero	Número de iteraciones para cálculo de modos por el método de Jacobi generalizado
21	NPES	Entero	Número de parejas de datos del espectro de pseudo-aceleraciones
22	NACE	Entero	Número de datos de cada acelerograma
23	NCAR	Entero	Número de cargas distribuidas aplicadas a la estructura
24	NDIN	Entero	Número de grados de libertad dinámicos de la estructura
25	NROT	Entero	Número de rótulas tipo
26	ROTE	Entero	Número de rótulas de la estructura
27	ITED	Entero	Número de iteraciones para convergencia de desplazamientos
28	ITEF	Entero	Número de iteraciones para convergencia de fuerzas

Tabla 6.1. Descripción de las variables escalares declaradas

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

N°	Nombre	Tipo	Descripción
29	NMIT	Entero	Número máximo de iteraciones permitido
30	FILE	Entero	Número auxiliar para indicar donde se almacenan las acciones en los extremos de cada elemento
31	FILA	Entero	Contador de filas para escritura de resultados en la hoja TB_OUT
32	COLM	Entero	Contador de columnas para escritura de resultados en la hoja TB_OUT
33	IVIG	Entero	Contador de ejes de vigas
34	ICOL	Entero	Contador de ejes de columnas
35	IELE	Entero	Contador de elementos
36	ITIM	Entero	Contador de intervalos de tiempo
37	IROT	Entero	Contador de rótulas
38	IDIN	Entero	Contador de grados de libertad dinámicos
39	FRNI	Entero	Frecuencia de vibración N° 1 (para amortiguamiento de Rayleigh)
40	FRNJ	Entero	Frecuencia de vibración N° 2 (para amortiguamiento de Rayleigh)
41	AUXI	Entero	Variable auxiliar (constantemente se sobrescribe)
42	i	Entero	Variable auxiliar (constantemente se sobrescribe)
43	j	Entero	Variable auxiliar (constantemente se sobrescribe)
44	AMVI	Real	Fracción de amortiguamiento viscoso respecto al amortiguamiento crítico
45	DTIM	Real	Tamaño del intervalo de tiempo para análisis modal cronológico ó análisis dinámico inelástico
46	ERRD	Real	Error calculado en desplazamientos
47	ERRF	Real	Error calculado en fuerzas
48	TOLD	Real	Tolerancia para convergencia de desplazamientos
49	TOLF	Real	Tolerancia para convergencia de fuerzas
50	FESX	Real	Factor de escala del movimiento sísmico (espectro o acelerograma) aplicado en dirección X global
51	FESY	Real	Factor de escala del movimiento sísmico (espectro o acelerograma) aplicado en dirección Y global
52	TINI	Real	Instante de tiempo en el que comienza el análisis
53	IDMA	Cadena	Indicador del método de análisis (modal espectral, modal cronológico o dinámico inelástico)
54	IDMC	Cadena	Indicador del método de solución de las ecuaciones modales desacopladas (Aceleración lineal, Excitación lineal o Newmark)
55	IDCM	Cadena	Indicador del método de combinación modal (RCSC, CCC)
56	IDCD	Cadena	Indicador del método de combinación direccional (100/30, RCSC)
57	IDAR	Cadena	Indicador de la forma de amortiguamiento de Rayleigh usada (constante o tangente)
58	IDSD	Cadena	Indicador del método de solución directa de las ecuaciones de movimiento (Aceleración constante o Aceleración lineal)
59	ROTU	Cadena	Indicador de elementos con rótulas (SI o NO)

Tabla 6.1. Descripción de las variables escalares declaradas (*cont.*)

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Declaración de variables escalares

'variables enteras
Dim NGLN As Integer, NNUE As Integer, NGLE As Integer, NNUD As Integer, NELE As Integer, NDIM As Integer, _
    NPAU As Integer, NDIR As Integer, NCOL As Integer, NVIG As Integer, NMAT As Integer, NSEC As Integer, _
    NGLT As Integer, NGLC As Integer, NGLO As Integer, NGLI As Integer, NGLD As Integer, NGDR As Integer, _
    NMOD As Integer, ITER As Integer, NPES As Integer, NACE As Integer, NCAR As Integer, NDIN As Integer, _
    NROT As Integer, ROTE As Integer, ITED As Integer, ITEF As Integer, NMIT As Integer, FILE As Integer, _
    FILA As Integer, COLM As Integer, ICOL As Integer, IVIG As Integer, IELE As Integer, ITIM As Integer, _
    IROT As Integer, IDIN As Integer, FRNI As Integer, FRNJ As Integer, AUXI As Integer, i As Integer, j As Integer

'variables reales
Dim AMVI As Double, DTIM As Double, ERRD As Double, ERRF As Double, TOLD As Double, FESX As Double, _
    FESY As Double, TINI As Double

'variables de texto
Dim IDMA As String, IDMC As String, IDCM As String, IDCD As String, IDAR As String, IDSD As String, ROTU As String
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.11. Declaración de variables escalares

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

6.2.2. Declaración de matrices

Las matrices se utilizaron para almacenar un conjunto de valores ordenados de forma secuencial, de tipo entero o real. Se utilizaron matrices de dos dimensiones.

El código principal requirió de 171 matrices para almacenar los datos de entrada y los resultados que se alcanzaban en cada etapa de cálculo. Se utilizaron 154 matrices reales de doble precisión (*double*) y 17 matrices enteras (*integer*). No se utilizaron matrices de tipo cadena (*string*).

En la Tabla 6.2 se indican el tipo y una descripción de cada una de las matrices declaradas en el código principal, así como el grupo al cual se asignó cada matriz para facilitar su identificación dentro del código.

N°	Nombre	Tipo	Descripción	Grupo
1	`ELE()`	Entero	Tabla de conectividades entre elementos	Geometría
2	`MGL()`	Entero	Tabla de numeración de grados de libertad por nudo	
3	`INC()`	Entero	Tabla de numeración de grados de libertad por elemento (*tabla de incidencias*)	
4	`PAU()`	Entero	Tabla de puntos auxiliares asociados a cada elemento para la definición de los ejes locales	
5	`MRE()`	Entero	Tabla de clasificación de grados de libertad	
6	`NDR()`	Entero	Tabla auxiliar de indicadores de nudos asociados a cada diafragma rígido	
7	`CND()`	Entero	Tabla auxiliar de cantidad de nudos asociados a cada diafragma rígido	
8	`AMS()`	Entero	Tabla de asignación de materiales y secciones tipo a elementos	
9	`NEC()`	Entero	Tabla auxiliar de cantidad de nudos asociados a cada eje de columnas	
10	`NNE()`	Entero	Tabla auxiliar de indicadores de nudos asociados a cada eje de columnas	
11	`VIG()`	Entero	Tabla auxiliar de indicadores de elementos asociados a cada eje de vigas	
12	`COL()`	Entero	Tabla auxiliar de indicadores de elementos asociados a cada eje de columnas	Geometría
13	`RIG()`	Entero	Tabla de asignación de rigidez tipo a elementos	
14	`CTS()`	Entero	Tabla de número de cortes por elemento, para calcular acciones internas	
15	`ARE()`	Entero	Tabla de asignación de rótulas plásticas a elementos	
16	`NRO()`	Entero	Tabla de numeración de rótulas plásticas de la estructura	
17	`XYZ()`	Real	Tabla de coordenadas de nudos y de puntos auxiliares	
18	`MDC()`	Real	Tabla de desplazamientos conocidos por nudo	
19	`DRX()`	Real	Tabla de coordenadas de centros de masa	
20	`CXY()`	Real	Tabla de coordenadas XY de ejes de columnas	
21	`TDR()`	Real	Matriz de asociación de desplazamientos	
22	`TDT()`	Real	Matriz transpuesta de asociación de desplazamientos	
23	`MAS()`	Real	Tabla de masas traslacional y rotacional por diafragma	Propiedades de la estructura
24	`MAT()`	Real	Tabla de materiales tipo	
25	`SEC()`	Real	Tabla de secciones transversales tipo	
26	`JRS()`	Real	Tabla de rigideces de conexiones elásticas	
27	`PAE()`	Real	Tabla de propiedades por elemento	
28	`TWE()`	Entero	Tabla de tipo de carga distribuida aplicada a elementos	Cargas distribuidas
29	`WEL()`	Real	Tabla con valores numéricos de las cargas distribuidas aplicadas a los elementos	
30	`FFE()`	Real	Vector de acciones fijas de la estructura en coordenadas globales, debidas a cargas iniciales	Acciones fijas
31	`RFF()`	Real	Tabla resumen de acciones fijas en coordenadas locales por elemento	

Tabla 6.2. Descripción de las matrices declaradas

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

N°	Nombre	Tipo	Descripción	Grupo
32	FNE()	Real	Vector de acciones externas aplicadas en los nudos de la estructura en coordenadas globales, debidas a cargas iniciales	Acciones aplicadas en los nudos
33	ROT()	Real	Tabla de propiedades de rótulas tipo	Rótulas plásticas
34	PRO()	Real	Tabla auxiliar de propiedades de las rótulas de la estructura	
35	DRO()	Real	Tabla de desplazamientos de rótulas plásticas en cada instante de tiempo	
36	ERO()	Real	Tabla de parámetros de desplazamiento elástico de rótulas plásticas en cada instante de tiempo	
37	FRO()	Real	Tabla de fuerzas en las rótulas plásticas en cada instante de tiempo	
38	KRO()	Real	Tabla de rigidez tangente de rótulas plásticas al comienzo de cada intervalo de tiempo	
39	MDR()	Real	Matriz de masa concentrada	Masa, rigidez y amorti-guamiento
40	AMR()	Real	Matriz de amortiguamiento	
41	KGL()	Real	Matriz de rigidez general considerando todos los grados de libertad estáticos	
42	KDI()	Real	Matriz de rigidez condensada a los grados de libertad de interés para el análisis dinámico	
43	KDP()	Real	Matriz de rigidez lateral (condensada a los grados de libertad de los diafragmas)	
44	KAA()	Real	Submatriz de KGL()	
45	KAG()	Real	Submatriz de KGL()	
46	KGA()	Real	Submatriz de KGL()	
47	KGG()	Real	Submatriz de KGL()	
48	KTE()	Real	Matriz de rigidez tangente efectiva	Masa, rigidez y amorti-guamiento
49	KEI()	Real	Matriz de rigidez tangente efectiva invertida	
50	KTR()	Real	Matriz de rigidez auxiliar, resta de matrices KTE() - KDP()	
51	VAP()	Real	Vector de frecuencias de vibración al cuadrado (valores propios)	Análisis modal
52	MOV()	Real	Matriz de modos de vibración ortonormalizados	
53	MOT()	Real	Matriz transpuesta de modos de vibración ortonormalizados	
54	FRN()	Real	Vector de frecuencias de vibración natural	
55	PER()	Real	Vector de periodos de vibración natural	
56	MMM()	Real	Matriz modal de masa	
57	MMK()	Real	Matriz modal de rigidez	
58	MMA()	Real	Matriz modal de amortiguamiento	
59	GAM()	Real	Matriz de influencia	
60	ALF()	Real	Matriz de factores de participación	Análisis modal
61	MME()	Real	Matriz de masa modal efectiva	
62	MMP()	Real	Matriz de masa total de la estructura	
63	MTE()	Real	Matriz de masa total de la estructura en porcentajes	
64	UGX()	Real	Tabla de aceleraciones de la base en dirección X global	Acelero-gramas y espectros
65	UGY()	Real	Tabla de aceleraciones de la base en dirección Y global	
66	ACE()	Real	Tabla resumen de aceleración de la base en función del tiempo	
67	TME()	Real	Vector de instantes de tiempo considerados en el análisis	
68	ESP()	Real	Espectro de pseudo-aceleraciones como fracción de la aceleración de la gravedad	
69	SAT()	Real	Pseudo-aceleraciones correspondientes a periodos de vibración de la estructura, interpoladas del espectro	
70	COR()	Real	Matriz de coeficientes de correlación (método CCC de combinación modal)	
71	MUO()	Real	Matriz de aceleraciones de la base	
72	TEI()	Real	Matriz de términos independientes de las ecuaciones modales desacopladas	
73	DCG()	Real	Desplazamientos generalizados en cada instante de tiempo	Coordena-das gene-ralizadas
74	VCG()	Real	Velocidades generalizadas en cada instante de tiempo	
75	ACG()	Real	Aceleraciones generalizadas en cada instante de tiempo	

Tabla 6.2. Descripción de las matrices declaradas (*cont.*)

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

N°	Nombre	Tipo	Descripción	Grupo
76	GEN()	Real	Desplazamientos generalizados máximos por modo y por componente de aceleración de la base	
77	CSD()	Real	Tabla de constantes de los métodos de solución directa de las ecuaciones de movimiento	Métodos de solución directa
78	AAA()	Real	Matriz auxiliar [a]	
79	BBB()	Real	Matriz auxiliar [b]	
80	FEI()	Real	Vector de incremento efectivo de acciones externas en el intervalo	
81	DRJ()	Real	Vector de acciones residuales que actúan en los centros de masa en la iteración j	
82	DCJ()	Real	Vector de incremento de desplazamientos de centros de masa en la iteración j	
83	DDC()	Real	Vector acumulado de incremento de desplazamientos de centros de masa en el intervalo	
84	DFS()	Real	Vector de incremento de acciones equivalentes en la iteración j	
85	DFJ()	Real	Vector de incremento efectivo de acciones equivalentes en la iteración j	
86	DCM()	Real	Desplazamientos de centros de masa en cada instante de tiempo	Desplazamientos, velocidades y aceleraciones de centros de masa
87	VCM()	Real	Velocidades de centros de masa en cada instante de tiempo	
88	ACM()	Real	Aceleraciones de centros de masa en cada instante de tiempo	
89	DCX()	Real	Desplazamientos máximos de centros de masa por modo, cuando actua la aceleración de la base solo en dirección X global	
90	DCY()	Real	Desplazamientos máximos de centros de masa por modo, cuando actua la aceleración de la base solo en dirección Y global	
91	DCR()	Real	Tabla resumen de desplazamientos máximos probables de centros de masa	
92	XCR()	Real	Tabla resumen de desplazamientos de centros de masa en dirección X global	
93	YCR()	Real	Tabla resumen de desplazamientos de centros de masa en dirección Y global	
94	ZCR()	Real	Tabla resumen de rotaciones de centros de masa en dirección Z global	
95	FEX()	Real	Acciones externas equivalentes a la aceleración de la base	Acciones trasladadas a centros de masa
96	FIN()	Real	Acciones inerciales trasladadas a centros de masa	
97	FAM()	Real	Acciones de amortiguamiento trasladadas a centros de masa	
98	FEQ()	Real	Acciones equivalentes trasladadas a centros de masa	
99	FCX()	Real	Acciones equivalentes máximas por modo, cuando actúa la aceleración de la base solo en dirección X global	
100	FCY()	Real	Acciones equivalentes máximas por modo, cuando actúa la aceleración de la base solo en dirección Y global	
101	FCR()	Real	Tabla resumen de acciones equivalentes máximas probables	
102	FXR()	Real	Tabla resumen de fuerzas equivalentes en dirección X global	
103	FYR()	Real	Tabla resumen de fuerzas equivalentes en dirección Y global	
104	MZR()	Real	Tabla resumen de momentos equivalentes en dirección Z global	
105	ECI()	Real	Vector de energía cinética	Energía
106	EAM()	Real	Vector de energía disipada por amortiguamiento natural	
107	ESS()	Real	Vector de energía de deformación	Energía
108	ESE()	Real	Vector de energía elástica de deformación	
109	ESY()	Real	Vector de energía inelástica de deformación, disipada por histéresis	
110	EEX()	Real	Vector de energía inducida por las acciones externas	
111	ENE()	Real	Tabla resumen de balance de energía de la estructura	
112	EXX()	Real	Derivas de centros de masa en dirección X global. Derivas de centros de masa en X máximas por modo, cuando actúa la aceleración de la base en dirección X global	Derivas de centros de masa
113	EYY()	Real	Derivas de centros de masa en dirección Y global. Derivas de centros de masa en Y máximas por modo, cuando actúa la aceleración de la base en dirección Y global	
114	EXY()	Real	Derivas de centros de masa en X máximas por modo, cuando actúa la aceleración de la base en dirección Y global	
115	EYX()	Real	Derivas de centros de masa en Y máximas por modo, cuando actúa la aceleración de la base en dirección X global	
116	EXR()	Real	Tabla resumen de derivas máximas probables de centros de masa en dirección X global	
117	EYR()	Real	Tabla resumen de derivas máximas probables de centros de masa en dirección Y global	

Tabla 6.2. Descripción de las matrices declaradas (*cont.*)

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

N°	Nombre	Tipo	Descripción	Grupo
118	VPX()	Real	Cortantes de piso en dirección X global	Cortantes de piso
119	VPY()	Real	Cortantes de piso en dirección Y global	
120	PXX()	Real	Cortantes de piso en dirección X global máximos por modo, cuando actúa la aceleración de la base en dirección X global	
121	PYY()	Real	Cortantes de piso en dirección X global máximos por modo, cuando actúa la aceleración de la base en dirección Y global	
122	PXY()	Real	Cortantes de piso en dirección Y global máximos por modo, cuando actúa la aceleración de la base en dirección X global	
123	PYX()	Real	Cortantes de piso en dirección Y global máximos por modo, cuando actúa la aceleración de la base en dirección Y global	
124	VXR()	Real	Tabla resumen de cortantes de piso máximos probables en dirección X global	
125	VYR()	Real	Tabla resumen de cortantes de piso máximos probables en dirección Y global	
126	DVX()	Real	Tabla resumen para graficar diagramas de cortantes de piso en dirección X global	
127	DVY()	Real	Tabla resumen para graficar diagramas de cortantes de piso en dirección Y global	
128	DBB()	Real	Vector de desplazamientos conocidos	Desplazamientos de nudos
129	DNG()	Real	Desplazamientos en coordenadas globales de nudos	
130	DNX()	Real	Desplazamientos de nudos máximos por modo, cuando actúa la aceleración de las base en dirección X global	
131	DNY()	Real	Desplazamientos de nudos máximos por modo, cuando actúa la aceleración de las base en dirección Y global	
132	DNR()	Real	Tabla resumen de desplazamientos máximos probables de nudos	
133	DEM()	Real	Tabla resumen de desplazamientos máximos de nudos y tiempos correspondientes	
134	DNJ()	Real	Incremento de desplazamientos de nudos en la iteración j	Desplazamientos de nudos
135	DDN()	Real	Incremento de desplazamientos de nudos en el intervalo, acumulado hasta la iteración j	
136	RDI()	Real	Tabla de desplazamientos de nudos debidos a cargas iniciales, organizada por nudo	
137	DEX()	Real	Derivas máximas por modo en un eje de columnas, cuando actúa la aceleración de la base en dirección X global	Derivas de ejes de columnas
138	DEY()	Real	Derivas máximas por modo en un eje de columnas, cuando actúa la aceleración de la base en dirección Y global	
139	DER()	Real	Derivas en un eje de columnas	
140	KEG()	Real	Matriz de rigidez elemental en coordenadas globales	Rigidez y transformación de coordenadas de elementos
141	KEL()	Real	Matriz de rigidez elemental en coordenadas locales	
142	TEL()	Real	Matriz elemental de transformación de coordenadas	
143	TET()	Real	Matriz elemental de transformación de coordenadas transpuesta	
144	UEG()	Real	Desplazamientos de nudos del elemento en coordenadas globales	Desplazamientos y acciones en los extremos de los elementos
145	UEL()	Real	Desplazamientos de nudos del elemento en coordenadas locales	
146	FEL()	Real	Acciones en coordenadas locales en los extremos del elemento	
147	FEG()	Real	Acciones en coordenadas globales en los extremos del elemento	
148	FBX()	Real	Acciones máximas por modo en los extremos del elemento, cuando actúa la aceleración de la base en dirección X global	
149	FBY()	Real	Acciones máximas por modo en los extremos del elemento, cuando actúa la aceleración de la base en dirección Y global	Desplazamientos y acciones en los extremos de los elementos
150	FEM()	Real	Tabla resumen de acciones máximas en coordenadas locales en los extremos de los elementos e instantes de tiempo correspondientes	
151	RFI()	Real	Tabla resumen de acciones en coordenadas locales en los extremos de los elementos producidas por las cargas iniciales	
152	FBL()	Real	Tabla resumen de acciones en coordenadas locales en los extremos de los elementos, calculadas por análisis dinámico inelástico	
153	NXX()	Real	Tabla para graficar diagramas de fuerza axial	Diagramas de acciones internas
154	VYY()	Real	Tabla para graficar diagramas de fuerza cortante en dirección Y local	
155	VZZ()	Real	Tabla para graficar diagramas de fuerza cortante en dirección Z local	
156	TXX()	Real	Tabla para graficar diagramas de momento torsor	
157	MYY()	Real	Tabla para graficar diagramas de momento flector en dirección Y local	

Tabla 6.2. Descripción de las matrices declaradas (*cont.*)

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

N°	Nombre	Tipo	Descripción	Grupo
158	MZZ()	Real	Tabla para graficar diagramas de momento flector en dirección Z local	
159	ITE()	Real	Tabla resumen de iteraciones realizadas en cada intervalo de tiempo	Iteraciones y errores
160	ERR()	Real	Tabla resumen de error final en iteraciones	
161	VF1()	Real	Tabla de verificación de la ecuación de movimiento en cada instante de tiempo	Verifica-ción de resultados
162	VF2()	Real	Tabla de verificación de la ecuación de balance de energía en cada instante de tiempo	
163	VF3()	Real	Tabla de verificación de acciones en las rótulas plásticas en cada instante de tiempo	
164	TM1()	Real	Matriz temporal (constantemente se sobrescribe)	Matrices temporales
165	TM2()	Real	Matriz temporal (constantemente se sobrescribe)	
166	TM3()	Real	Matriz temporal (constantemente se sobrescribe)	
167	TM4()	Real	Matriz temporal (constantemente se sobrescribe)	
168	TM5()	Real	Matriz temporal (constantemente se sobrescribe)	
169	TM6()	Real	Matriz temporal (constantemente se sobrescribe)	
170	TM7()	Real	Matriz temporal (constantemente se sobrescribe)	
171	TM8()	Real	Matriz temporal (constantemente se sobrescribe)	

Tabla 6.2. Descripción de las matrices declaradas (*cont.*)

La Figura 6.12 muestra las líneas del código principal en las que se declaran las matrices descritas en la Tabla 6.2, agrupadas como se muestra en la misma tabla.

```
'Declaración de matrices

'geometría
Dim ELE() As Integer, MGL() As Integer, INC() As Integer, PAU() As Integer, MRE() As Integer, NDR() As Integer, _
    CND() As Integer, AMS() As Integer, NEC() As Integer, NNE() As Integer, VIG() As Integer, COL() As Integer, _
    RIG() As Integer, CTS() As Integer, ARE() As Integer, NRO() As Integer

Dim XYZ() As Double, MDC() As Double, DRX() As Double, CXY() As Double, TDR() As Double, TDT() As Double

'propiedades de la estructura
Dim MAS() As Double, MAT() As Double, SEC() As Double, JRS() As Double, PAE() As Double

'cargas distribuidas
Dim TWE() As Integer, WEL() As Double

'acciones fijas
Dim FFE() As Double, RFF() As Double

'acciones aplicadas en los nudos
Dim FNE() As Double

'rótulas plásticas
Dim ROT() As Double, PRO() As Double, DRO() As Double, ERO() As Double, FRO() As Double, KRO() As Double

'masa, rigidez y amortiguamiento
Dim MDR() As Double, AMR() As Double, KGL() As Double, KDI() As Double, KDP() As Double, KAA() As Double, _
    KAG() As Double, KGA() As Double, KGG() As Double, KTE() As Double, KEI() As Double, KTR() As Double

'análisis modal
Dim VAP() As Double, MOV() As Double, MOT() As Double, FRN() As Double, PER() As Double, MMM() As Double, _
    MMK() As Double, MMA() As Double, GAM() As Double, ALF() As Double, MME() As Double, MMP() As Double, _
    MTE() As Double

'acelerogramas y espectros
Dim UGX() As Double, UGY() As Double, ACE() As Double, TME() As Double, ESP() As Double, SAT() As Double, _
    COR() As Double, MUO() As Double, TEI() As Double

'coordenadas generalizadas
Dim DCG() As Double, VCG() As Double, ACG() As Double, GEN() As Double

'métodos de solución directa
Dim CSD() As Double, AAA() As Double, BBB() As Double, FEI() As Double, DRJ() As Double, DCJ() As Double, _
    DDC() As Double, DFS() As Double, DFJ() As Double
```

Figura 6.12. Declaración de matrices

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'desplazamientos, velocidades y aceleraciones de centros de masa
Dim DCM() As Double, VCM() As Double, ACM() As Double, DCX() As Double, DCY() As Double, DCR() As Double, _
    XCR() As Double, YCR() As Double, ZCR() As Double

'acciones trasladadas a centros de masa
Dim FEX() As Double, FIN() As Double, FAM() As Double, FEQ() As Double, FCX() As Double, FCY() As Double, _
    FCR() As Double, FXR() As Double, FYR() As Double, MZR() As Double

'energía
Dim ECI() As Double, EAM() As Double, ESS() As Double, ESE() As Double, ESY() As Double, EEX() As Double, _
    ENE() As Double

'derivas de centros de masa
Dim EXX() As Double, EYY() As Double, EXY() As Double, EYX() As Double, EXR() As Double, EYR() As Double

'cortantes de piso
Dim VPX() As Double, VPY() As Double, PXX() As Double, PYY() As Double, PXY() As Double, PYX() As Double, _
    VXR() As Double, VYR() As Double, DVX() As Double, DVY() As Double

'desplazamientos de nudos
Dim DBB() As Double, DNG() As Double, DNX() As Double, DNY() As Double, DNR() As Double, DEM() As Double, _
    DNJ() As Double, DDN() As Double, RDI() As Double

'derivas de ejes de columnas
Dim DEX() As Double, DEY() As Double, DER() As Double

'rigidez y transformación de coordenadas de elementos
Dim KEG() As Double, KEL() As Double, TEL() As Double, TET() As Double

'desplazamientos y acciones en los extremos de elementos
Dim UEG() As Double, UEL() As Double, FEL() As Double, FEG() As Double, FBX() As Double, FBY() As Double, _
    FEM() As Double, RFI() As Double, FBL() As Double

'diagramas de acciones internas en ejes de vigas y columnas
Dim NXX() As Double, VYY() As Double, VZZ() As Double, TXX() As Double, MYY() As Double, MZZ() As Double

'iteraciones y errores
Dim ITE() As Double, ERR() As Double

'verificación de resultados
Dim VF1() As Double, VF2() As Double, VF3() As Double

'matrices temporales
Dim TM1() As Double, TM2() As Double, TM3() As Double, TM4() As Double, TM5() As Double, TM6() As Double, _
    TM7() As Double, TM8() As Double
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.12. Declaración de matrices (*cont.*)

6.2.3. Escritura del encabezado y limpieza de hoja de resultados

En esta etapa del código se realizan las siguientes instrucciones:

- Se imprime el encabezado de la Figura 6.13 en la ventana de inmediato, como se muestra en la Figura 6.14.
- Se inicializa la variable *TINI* con el instante de tiempo en el que comienza el análisis de la estructura.
- Se borra el contenido de la hoja de resultados "*TB_OUT*" con la subrutina *EDLIMH* original de PEFiCA
- Se inicializan las variables *FILA* y *COLM* en las posiciones 2 y 1 para comenzar a escribir los resultados en la hoja "*TB_OUT*".

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'-----------------------------------------------------------------------------------------------------------
'Escribir encabezado y linpiar hoja de resultados

'Encabezado hoja de inmediato
Debug.Print "_____________________________________________________________________________________________"
Debug.Print "Programa Didáctico a Código Abierto de Análisis Dinámico de Estructuras"
Debug.Print "Universidad Nacional de Colombia. 2012"
Debug.Print "_____________________________________________________________________________________________"
Debug.Print "calculando ..."

'instante de tiempo inicial
TINI = Timer

'limpiar la hoja TB_OUT
EDLIMH "TB_OUT"

'fila y columna iniciales para escritura en la hoja TB_OUT
FILA = 2
COLM = 1
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.13. Escritura de encabezado y limpieza de hoja de resultados

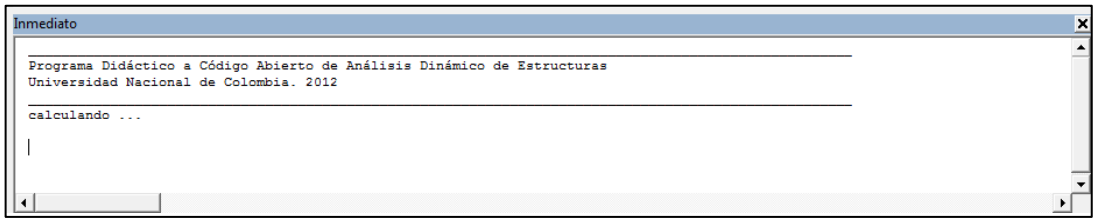


Figura 6.14. Escritura del encabezado en la ventana de inmediato

6.3. Parte II: Lectura de datos de entrada y creación de matrices auxiliares

A continuación se describen las líneas de la subrutina principal UNDIN con las que se leen los datos de entrada necesarios para el análisis dinámico, los cuales son de tipo escalar o matricial, y deben introducirse por el usuario en las diferentes hojas del libro de cálculo.

Se describe la forma como se definen parámetros generales relacionados con las cantidades de grados de libertad por tipo. También se describe la forma de las matrices auxiliares que se construyen en esta etapa de cálculo y que son utilizadas constantemente.

6.3.1. Lectura y definición de parámetros generales

Los parámetros generales que se describen a continuación se requieren posteriormente para definir el tamaño de las matrices de datos de entrada y de resultados, asi como para definir el tamaño de los ciclos que implican cálculos por nudo, por elemento, por intervalo de tiempo, etc.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

Los parámetros generales que son datos de entrada se leen de la hoja "*TB_GEN*". Se han agrupado de la siguiente forma:

- *Parámetros generales de la estructura:*

Estos parámetros están relacionados con las características generales de la estructura en cuanto a número de nudos, número de elementos, número de dimensiones del problema, número de puntos auxiliares para definir ejes locales de elementos, número de materiales y secciones tipo, número de diafragmas rígidos, número de ejes de vigas, número de ejes de columnas, número de rótulas plásticas tipo y número de cargas distribuidas.

Los dos últimos parámetros se requieren cuando se utiliza el método de análisis dinámico inelástico, mientras que los demás se requieren siempre, independientemente del método de análisis dinámico que se utilice.

- *Parámetros de amortiguamiento natural*

Estos parámetros están relacionados con la definición del tipo de amortiguamiento natural a considerar y con el valor de la fracción de amortiguamiento natural respecto al crítico ξ.

Cuando se utilizan métodos de análisis dinámico elástico (*análisis modal cronológico y análisis modal espectral*) UNDIN asocia un valor de amortiguamiento modal igual para todos los modos de vibración. En este caso solo se debe especificar el valor de la fracción de amortiguamiento modal ξ.

Cuando se utilizan métodos de análisis dinámico inelástico, UNDIN utiliza el criterio de amortiguamiento proporcional a la masa y a la rigidez de Rayleigh. En este caso se debe especificar si se utiliza la matriz de rigidez inicial o la matriz de rigidez tangente al comienzo de cada intervalo de tiempo, con lo cual el amortiguamiento se considera constante o variable. Asimismo deben especificarse las dos frecuencias de vibración con amortiguamiento conocido y el valor de la fracción de amortiguamiento modal ξ de las dos frecuencias seleccionadas.

- *Parámetros de movimientos sísmicos*

Estos parámetros están relacionados con los movimientos sísmicos que se aplican a la estructura y que, dependiendo el método de análisis dinámico utilizado, corresponden a la aceleración de la base o el espectro de pseudo – aceleraciones, los cuales se aplican en direcciones x e y globales.

Cuando se utiliza análisis dinámico inelástico o análisis modal cronológico deben especificarse el número de datos de cada acelerograma y el tamaño del intervalo de tiempo, constante entre datos sucesivos, este último parámetro será el tamaño del intervalo de tiempo requerido para la solución numérica de las ecuaciones de movimiento. Cuando se utiliza análisis modal espectral se requiere solamente el número de parejas de datos del espectro de

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

pseudo-aceleraciones. En los dos casos se deben especificar factores de escala para amplificar o reducir la magnitud de los movimientos sísmicos.

- *Parámetros de métodos de solución*

Estos parámetros están relacionados con la selección del método general de análisis dinámico y con la selección de los métodos particulares propios de cada tipo de análisis.

Cuando se utiliza análisis modal cronológico se debe especificar el método de solución de las ecuaciones modales desacopladas. El programa dispone de los tres procedimientos descritos en el Capítulo 4 de análisis dinámico elástico para tal fin.

Sí se utiliza análisis modal espectral se deben especificar los métodos de combinación modal y direccional, los cuales se requieren para estimar la respuesta máxima probable de la estructura. El programa trabaja con los métodos *RCSC* y *CCC* para combinación modal y con la regla *100/30* y el método *RCSC* para combinación direccional, descritos en el capítulo de análisis dinámico elástico.

Finalmente, cuando se utiliza análisis dinámico inelástico, se debe especificar el método de solución directa de las ecuaciones de movimiento en forma incremental, el número máximo de iteraciones y la tolerancia para la convergencia de resultados. El programa utiliza la familia de métodos de Newmark en combinación con el método de Newton – Raphson descrito en el Capítulo 5. Debe especificarse una de las dos formas del método de Newmark, aceleración promedio o aceleración lineal.

La Tabla 6.3 muestra los datos que se leen de la hoja "*TB_GEN*" y el valor que pueden tomar según las características de pórtico y del método de análisis dinámico utilizado.

N°	Grupo	Variable	Contenido	Tipo	Valor permitido
1	Parámetros generales de la estructura	NNUD	Número de nudos	Entero	Mínimo 2
2		NELE	Número de elementos	Entero	Mínimo 1
3		NDIM	Número de dimensiones	Entero	2 (*pórtico plano*) o 3 (*pórtico espacial*)
4		NPAU	Número de puntos auxiliares	Entero	Cualquier valor, incluido cero
5		NMAT	Número de materiales tipo	Entero	Mínimo 1
6		NSEC	Número de secciones tipo	Entero	Mínimo 1
7		NROT	Número de rótulas tipo	Entero	Mínimo 1
8		NCAR	Número de cargas iniciales distribuidas en la luz de los elementos	Entero	Mínimo 1
9		NDIR	Número de diafragmas rígidos	Entero	Mínimo 1
10		NCOL	Número de ejes de columnas	Entero	Mínimo 1
11		NVIG	Número de ejes de vigas	Entero	Mínimo 1
12	Amortiguamiento natural	AMVI	Fracción de amortiguamiento modal	Real	Cualquier valor entre cero y 0.99
13		IDAR	Forma de amortiguamiento de Rayleigh	Cadena	"*CONSTANTE*" o "*TANGENTE*"
14		FRNI	Frecuencia de vibración número ω_i	Entero	Cualquier valor menor o igual al número de modos de la estructura
15		FRNJ	Frecuencia de vibración número ω_j	Entero	Cualquier valor menor o igual al número de modos de la estructura
16	Movimientos sísmicos	NACE	Número de datos de cada acelerograma	Entero	Cualquier valor mayor a cero
17		DTIM	Tamaño del intervalo de tiempo	Real	Cualquier valor mayor a cero

Tabla 6.3. Parámetros generales leídos de la hoja "*TB_GEN*"

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

N°	Grupo	Variable	Contenido	Tipo	Valor permitido
18	Movimientos sísmicos	NPES	Número de datos del espectro de pseudo-aceleraciones	Entero	Cualquier valor mayor a cero
19		FESX	Factor de escala del movimiento sísmico en dirección X global	Real	Cualquier valor mayor a cero
20		FESY	Factor de escala del movimiento sísmico en dirección Y global	Real	Cualquier valor mayor a cero
21	Métodos de solución	IDMA	Método de análisis dinámico	Cadena	"*MODAL CRONOLÓGICO*", "*MODAL ESPECTRAL*" o "*DINÁMICO INELÁSTICO*"
22		IDMC	Método para solucionar las ecuaciones modales desacopladas	Cadena	"*ACELERACIÓN LINEAL*", "*EXCITACIÓN LINEAL*" o "*NEWMARK*"
23		IDCM	Método de combinación modal	Cadena	"*RCSC*" o "*CQC*"
24		IDCD	Método de combinación direccional	Cadena	"*100/30*" o "*RCSC*"
25		IDSD	Método de solución directa de las ecuaciones de movimiento	Cadena	"*ACELERACIÓN PROMEDIO*" o "*ACELERACIÓN LINEAL*"
26		NMIT	Número máximo de iteraciones permitido	Entero	Cualquier valor mayor a cero
27		TOLD	Tolerancia para convergencia de resultados	Real	Cualquier valor mayor a cero

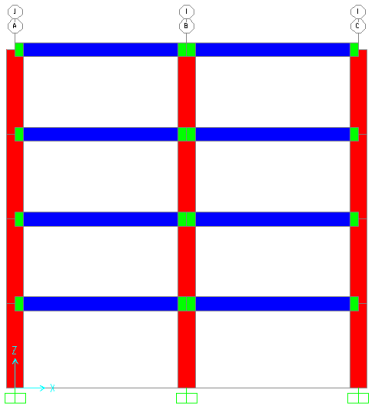
Tabla 6.3. Parámetros generales leídos de la hoja "*TB_GEN*" (cont.)

A modo de ejemplo la Figura 6.15 muestra los datos que se leen de la hoja "*TB_GEN*" para el análisis de un pórtico plano.

ANÁLISIS DINÁMICO INELÁSTICO-PÓRTICO PLANO

PÁRAMETROS GENERALES DE LA ESTRUCTURA

NOMBRE	PÓRTICO PLANO
NÚMERO DE NUDOS	31
NÚMERO DE ELEMENTOS	36
NÙMERO DE DIMENSIONES	2
NÚMERO DE PUNTOS AUXILIARES	0
NÚMERO DE MATERIALES	2
NÚMERO DE SECCIONES	2
NÚMERO DE RÓTULAS PLÁSTICAS TIPO	5
NÚMERO DE CARGAS DISTRIBUIDAS	24
NÚMERO DE DIAFRAGMAS RÍGIDOS	4
NÚMERO DE EJES DE COLUMNAS	3
NÚMERO DE EJES DE VIGAS	4



AMORTIGUAMIENTO NATURAL

FRACCIÓN DE AMORTIGUAMIENTO	0.05
FORMA DE AMORTIGUAMIENTO DE RAYLEIGH	TANGENTE
FRECUENCIA N° 1	1
FRECUENCIA N° 2	2

MOVIMIENTOS SÍSMICOS

NÚMERO DE PUNTOS DEL ACELEROGRAMA	5000
INTERVALO DE TIEMPO ENTRE DATOS DE ACELEROGRAMAS	0.010
NÚMERO DE PUNTOS DEL ESPECTRO	12
FACTOR DE ESCALA DEL MOVIMIENTO SÍSMICO EN DIRECCIÓN X	20
FACTOR DE ESCALA DEL MOVIMIENTO SÍSMICO EN DIRECCIÓN Y	0

MÉTODOS DE SOLUCIÓN

TIPO DE ANÁLISIS DINÁMICO	DINÁMICO INELÁSTICO
MÉTODO DE SOLUCIÓN DE LAS ECUACIONES MODALES DESACOPLADAS	EXCITACIÓN LINEAL
MÉTODO DE COMBINACIÓN MODAL	CQC
MÉTODO DE COMBINACIÓN DIRECCIONAL	100/30
MÉTODO DE SOLUCIÓN DIRECTA DE LAS ECUACIONES DE MOVIMIENTO	ACELERACIÓN LINEAL
NÚMERO MÁXIMO DE ITERACIONES	50
ERROR MÁXIMO	1E-05

Figura 6.15. Ejemplo de datos de entrada de la hoja "*TB_GEN*"

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

Finalmente se fijan los valores de las variables *NGLN*, *NNUE* y *NGLE*, las cuales almacenan valores relacionados con el número de grados de libertad por nudo y por elemento. Estos se fijan puesto que esta versión del programa solo considera elementos tipo pórtico de dos nudos y de seis grados de libertad por nudo, es decir, de 12 grados de libertad por elemento.

Esta forma de considerar los grados de libertad es general y se puede usar en estructuras planas y espaciales, con la única diferencia que en estructuras planas tres de los seis grados de libertad de cada nudo se restringen y por lo tanto valen cero. La Tabla 6.4 resume el nombre de cada parámetro general fijo y su valor numérico.

N°	Variable	Tipo	Parámetro	Valor numérico
1	*NGLN*	Escalar	Número de grados de libertad por nudo.	6
2	*NNUE*	Escalar	Número de nudos por nudo.	2
3	*NGLE*	Escalar	Número de grados de libertad por elemento.	12

Tabla 6.4. Parámetros generales fijos

En la Figura 6.16 y en la Figura 6.17 se muestran las líneas de la subrutina principal con las que se leen de la hoja "*TB_GEN*" los datos de entrada de tipo escalar, utilizando las subrutinas *EDLECE* y *EDLECT* originales de *PEFiCA*, y se fijan los valores las variables enteras *NGLN*, *NNUE* y *NGLE*.

```
'-----------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------
'Lectura y definición de parámetros generales

'lectura de parámetros generales de la estructura
EDLECE "TB_GEN", 5, 2, NNUD     'número de nudos (NNUD)
EDLECE "TB_GEN", 6, 2, NELE     'número de elementos (NELE)
EDLECE "TB_GEN", 7, 2, NDIM     'número de dimensiones (NDIM)
EDLECE "TB_GEN", 8, 2, NPAU     'número de puntos auxiliares (NPAU)
EDLECE "TB_GEN", 9, 2, NMAT     'número de materiales (NMAT)
EDLECE "TB_GEN", 10, 2, NSEC    'número de secciones (NSEC)
EDLECE "TB_GEN", 11, 2, NROT    'número de rótulas plásticas tipo (NROT)
EDLECE "TB_GEN", 12, 2, NCAR    'número de cargas distribuidas (NCAR)
EDLECE "TB_GEN", 13, 2, NDIR    'número de diafragmas rígidos (NDIR)
EDLECE "TB_GEN", 14, 2, NCOL    'número de ejes de columnas (NCOL)
EDLECE "TB_GEN", 15, 2, NVIG    'número de ejes de vigas (NVIG)

'lectura de parámetros de amortiguamiento natural
EDLECE "TB_GEN", 18, 2, AMVI    'coeficiente de amortiguamiento viscoso equivalente (AMVI)
EDLECT "TB_GEN", 19, 2, IDAR    'forma de amortiguamiento de Rayleigh (IDAR)
EDLECE "TB_GEN", 20, 2, FRNI    'indicador frecuencia N° 2 (FRNI)
EDLECE "TB_GEN", 21, 2, FRNJ    'indicador frecuencia N° 2 (FRNJ)

'lectura de parámetros de movimientos sísmicos
EDLECE "TB_GEN", 24, 2, NACE    'número de datos de cada acelerograma (ó número de intervalos) (NACE)
EDLECE "TB_GEN", 25, 2, DTIM    'intervalo de tiempo entre datos (DTIM)
EDLECE "TB_GEN", 26, 2, NPES    'número de datos del espectro de pseudo-aceleraciones (NPES)
EDLECE "TB_GEN", 27, 2, FESX    'factor de escala del movimiento sísmico aplicado en dirección X (FESX)
EDLECE "TB_GEN", 28, 2, FESY    'factor de escala del movimiento sísmico aplicado en dirección Y (FESY)

'lectura de parámetros para métodos de solución
EDLECT "TB_GEN", 31, 2, IDMA     'método de análisis dinámico (IDMA)
EDLECT "TB_GEN", 32, 2, IDMC    'método de cálculo de las coordenadas generalizadas en el tiempo(IDMC)
EDLECT "TB_GEN", 33, 2, IDCM    'métodos de combinación modal (IDCM)
EDLECT "TB_GEN", 34, 2, IDCD    'métodos de combinación direccional (IDCD)
EDLECT "TB_GEN", 35, 2, IDSD    'método de solución directa de las ecuaciones de movimiento (IDSD)
EDLECE "TB_GEN", 36, 2, NMIT    'número máximo de iteraciones (NMIT)
EDLECE "TB_GEN", 37, 2, TOLD    'tolerancia o error máximo de desplazamientos (TOLD)
```

Figura 6.16. Lectura de parámetros generales desde la hoja "*TB_GEN*"

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'número de grados de libertad por nudo
NGLN = 6

'número de nudos por elemento
NNUE = 2

'número de grados de libertad por elemento
NGLE = NNUE * NGLN

Debug.Print "Lectura y definición de parámetros generales", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------------
```

Figura 6.17. Parámetros generales fijados

6.3.2. Lectura de tablas de datos

UNDIN requiere 22 tablas que se leen desde las 10 hojas de datos de entrada. Algunas tablas se requieren independientemente del método de análisis dinámico que se utilice para resolver la estructura, mientras que otras tablas se requieren solo si se utiliza un método de análisis dinámico específico. A continuación se describe cada una de las tablas de datos que requiere el programa.

- *Tabla de coordenadas de nudos y de puntos auxiliares*

Esta tabla contiene las coordenadas x, y, z de cada nudo de la estructura y de cada punto auxiliar, utilizado para definir los ejes locales de los elementos por el primer procedimiento descrito en el Capítulo 3. El número de filas de la columna coincide con el número de nudos más el número de puntos auxiliares especificados en la hoja "*TB_GEN*", mientras que el número de columnas es tres.

Esta tabla se lee de la hoja "*TB_XYZ*" y se almacena en la matriz real *XYZ()*. A modo de ejemplo la Figura 6.18 muestra, para una estructura de seis nudos y sin puntos auxiliares, la parte de la tabla que se lee desde el código principal.

TABLA DE COORDENADAS DE LOS NUDOS
(los puntos auxiliares se numeran al final)

NUDO	X (m)	Y (m)	Z (m)
1	10.000	0.000	0.000
2	25.000	0.000	0.000
3	0.000	0.000	7.000
4	10.000	0.000	7.000
5	25.000	0.000	7.000
6	35.000	0.000	7.000

XYZ ()

TB_GEN TB_XYZ TB_PRO TB_ELE

Figura 6.18. Ejemplo de tabla *XYZ()*

- *Tablas de definición de materiales y de secciones tipo*

La tabla de definición de materiales tipo contiene el módulo de Young y la relación de Poisson de cada material tipo de la estructura. El número de filas coincide con el número de materiales tipo especificados en "*TB_GEN*" y el número de columnas es dos.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La tabla de propiedades de secciones contiene las propiedades de cada sección tipo que compone la estructura. Las propiedades de cada sección requeridas son el área A, los momentos de inercia respecto a los ejes principales $\bar{y}$ y $\bar{z}$, la constante de torsión J y el factor de forma f_s para considerar deformaciones por cortante. El número de filas de la tabla coincide con el número de secciones tipo especificadas en la hoja "*TB_GEN*" y el número de columnas es cinco.

Estas tablas se leen de la hoja "*TB_PRO*" y se almacenan en la matrices reales *MAT()* y *SEC()* respectivamente. La Figura 6.19 muestra, para una estructura con todos los elementos del mismo matrial pero con dos secciones transversales diferentes, la parte de las tablas que se leen desde el código principal.

TABLA DE MATERIALES

Material N°	E (kN/m2)	ν (adim)
1	2.20E+07	0.25

MAT ()

TABLA DE PROPIEDADES DE SECCIONES

Sección N°	Tipo	b (m)	h (m)	D (m)	a	b	A (m^2)	I_z (m^4)	I_y (m^4)	J (m^4)	fs (adim)
1	Rectangular	0.40	0.60	0.00	0.40	0.60	0.7100	0.0519	0.2202	0.077	1.20
2	Rectangular	1.00	0.60	0.00	0.60	1.00	0.6000	0.0180	0.0500	0.045	1.20

SEC ()

Figura 6.19. Ejemplo de tablas *MAT()* y *SEC()*

- *Tablas de conectividades, de puntos auxiliares y de asignación de materiales, secciones, tipo de rigidez y número de cortes por elemento*

Las tablas que se describen a continuación contienen un número de filas que coincide con el número de elementos especificados en la hoja "*TB_GEN*". El número de columnas varía entre una y dos dependiendo de la tabla.

La tabla de conectividades posee dos columnas y contiene la numeración de los nudos inicial y final asociados a cada elemento de la estructura. Dependiendo del orden en el que se introduzcan estos valores se definen los ejes locales del elemento.

La tabla de asignación de puntos auxiliares posee una sola columna, y contiene el indicador del punto asociado a cada elemento con el que se define la orientación de los ejes locales por el criterio del punto auxiliar. El número del punto asociado a cada elemento debe coincidir con el número de la fila de la tabla *XYZ()* en las que se almacenaron sus coordenadas. Si en algún caso el valor asociado a cada elemento es cero, el programa utiliza el criterio por defecto para definir la orientación de los ejes locales del elemento.

La tabla de asignación de materiales y secciones a elementos posee dos columnas, la primera con el tipo de material y la segunda con el tipo de sección asignados a cada elemento de la estructura. Los números del material y de la sección asociados a cada elemento deben coincidir con los números de las filas de las tablas *MAT()* y *SEC()* en los que se almacenaron sus propiedades básicas. Por ejemplo, el material N° 1 coincide con la primera fila de la matriz *MAT()*.

La tabla de asignación de rigidez tipo a elementos es de una sola columna y contiene los coeficientes que indican el tipo de rigidez a considerar en cada elemento. Si se introduce el valor *000* la matriz de rigidez y el vector de acciones fijas del elemento no considera de-

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

formaciones por cortante ni conexiones parcialmente rígidas, si se introduce el valor *001* la matriz $[\bar{k}_e]$ y el vector $\{\bar{f}_e^F\}$ considera deformaciones por cortante más no conexiones parcialmente rígidas, y si se introduce el valor *002* se consideran simultáneamente deformaciones por cortante y conexiones parcialmente rígidas en la construcción de $[\bar{k}_e]$ y $\{\bar{f}_e^F\}$.

Finalmente la tabla de coeficientes de rigidez inicial de conexiones elásticas contiene las rigideces iniciales de los extremos de los elementos para los cuales el tipo de rigidez considerada es *002*, de acuerdo con lo especificado en la tabla de asignación de rigidez tipo a elementos. El programa permite considerar elementos con conexiones parcialmente rígidas a flexión en los dos extremos y en los dos planos de flexión principales, por lo que esta tabla posee 4 columnas, las dos primeras para almacenar los coeficientes de rigidez del nudo inicial $J_{r\bar{y}}^{(i)}$ y $J_{r\bar{z}}^{(i)}$, y las dos siguientes para almacenar la rigidez de las conexiones del nudo final $J_{r\bar{y}}^{(j)}$ y $J_{r\bar{y}}^{(j)}$ (*i y j indican nudos inicial y final del elemento*).

Las 6 tablas descritas previamente se leen de la hoja "*TB_ELE*" y se almacenan en las matrices *ELE(), PAU(), AMS(), RIG,CTS()* y *JRS()* respectivamente. La Figura 6.20 y la Figura 6.21 muestran para una estructura cualquiera la parte de las tablas que se leen desde el código principal.

TABLA DE CONECTIVIDADES

Elemento N°	Nudo inicial	Nudo final	IPAU
1	1	4	0
2	2	5	0
3	3	4	0
4	4	5	0
5	5	6	0

ELE () PAU ()

ASIGNACION DE MATERIALES Y SECCIONES

Elemento N°	Material Tipo	Sección Tipo	Rigidez Tipo	N° de cortes
1	001	002	002	008
2	001	002	002	008
3	001	001	001	011
4	001	001	001	016
5	001	001	001	011

AMS () RIG () CTS ()

Figura 6.20. Ejemplo de tablas *ELE(), PAU(), AMS(), RIG()* y *CTS()*

RIGIDEZ INICIAL DE CONEXIONES ELÁSTICAS

Elemento N°	Jry	Jrz	Jsy	Jsz
1	5.00E+04	5.00E+04	5.00E+04	1.00E+10
2	5.00E+04	5.00E+04	5.00E+04	1.00E+10
3	5.00E+04	5.00E+04	5.00E+04	5.00E+04
4	5.00E+04	5.00E+04	5.00E+04	5.00E+04
5	5.00E+04	5.00E+04	5.00E+04	5.00E+04

JRS ()

Figura 6.21. Ejemplo de tabla *JRS()*

- *Tablas de clasificación de grados de libertad y de desplazamientos conocidos*

Las tablas que se describen a continuación contienen un número de filas que coincide con el número de nudos de la estructura, mientras que el número de columnas coincide con el número de grados de libertad por nudo, es decir 6. Los datos introducidos en estas tablas quedan organizados por columnas de acuerdo a la numeración de los grados de libertad de

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

cada nudo, comenzando por los tres desplazamientos u_x, u_y y u_z y terminando con las tres rotaciones θ_x, θ_y y θ_z.

La tabla de clasificación de grados de libertad asocia un indicador a cada uno de los grados de libertad estáticos de los nudos. Si el indicador es *001* el grado de libertad es conocido, si el indicador es *002* el grado de libertad es desconocido condensado y si el indicador es *003* el grado de libertad es desconocido de interés.

La tabla de desplazamientos conocidos almacena los valores numéricos de los desplazamientos iniciales impuestos en los apoyos. En este caso solo es importante introducir como datos de entrada los valores correspondientes a los grados de libertad conocidos puesto que los demás posteriormente se calculan en función de las cargas externas o de los movimientos sísmicos aplicados a la estructura. Generalmente estos desplazamientos se toman generalmente como cero.

Las dos tablas anteriores se leen de la hoja "*TB_RES*" y se almacenan en las matrices *MRE()* y *MDC()* respectivamente. La Figura 6.22 muestra para una estructura cualquiera la parte de las tablas que se leen desde el código principal.

TABLA DE CLASIFICACIÓN DE GRADOS DE LIBERTAD

Interés = 003 / Condensado (002) / Conocido=001

NUDO	u_x	u_y	u_z	θ_x	θ_y	θ_z
1	001	001	001	001	001	001
2	001	001	001	001	001	001
3	003	001	001	001	002	001
4	003	001	002	001	002	001
5	003	001	002	001	002	001
6	003	001	001	001	002	001

MRE ()

TABLAS DE DESPLAZAMIENTOS CONOCIDOS POR NUDO

NUDO	u_x	u_y	u_z	θ_x	θ_y	θ_z
1	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00

MDC ()

Figura 6.22. Ejemplo de las tablas *MRE()* y *MDC()*

- *Tablas de coordenadas de centros de masa y de masas asociadas a diafragmas*

Las tablas que se describen a continuación contienen un número de filas que coincide con el número de diafragmas rígidos de la estructura y el número de columnas varía entre 2 y 3.

La tabla de coordenadas de centros de masa almacena las coordenadas x, y, z globales del centro de masa de cada diafragma rígido, por lo que su número de columnas es tres.

La tabla de masas asociadas a diafragmas posee dos columnas: la primera contiene la masa traslacional en direcciones x e y global asociada a cada diafragma, y la segunda contiene la masa rotacional del diafragma respecto al eje z global, esta última coincide con el momento de inercia de masa, calculado respecto a un eje que pase por el centro de masa del diafragma y que sea paralelo al eje z global.

Respecto a estas dos tablas, en caso que se analice una estructura de varios niveles es importante que los la información asociada a cada diafragma se introduzca de forma ordenada y descendente, es decir, comenzando por el diafragma más elevado (*coordenada z mayor*) y terminando con la información del diafragma más bajo (*coordenada z menor).*

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

Las dos tablas anteriores se leen de la hoja "*TB_DIR*" y se almacenan en las matrices *DRX()* y *MAS()* respectivamente. La Figura 6.23 muestra para una estructura cualquiera la parte de las tablas que se leen desde el código principal.

	A	B	C	D	E	F	G
1	TABLA DE DIAFRAGMAS RIGIDOS						
2							
3	Diafragma	Coordenadas Centros de Masa			Mx=My	Mrz	
4		X_{cm} (m)	Y_{cm} (m)	Z_{cm} (m)	kkg	kkg.m²	
5	1	6.000	0.000	12.000	72.0	0.0	
6	2	6.000	0.000	9.000	72.0	0.0	
7	3	6.000	0.000	6.000	72.0	0.0	
8	4	6.000	0.000	3.000	72.0	0.0	
9							
10			DRX ()		MAS ()		

Figura 6.23. Ejemplo de las tablas *DRX()* y *MAS()*

- *Tablas de definición y asignación de rótulas plásticas tipo (resortes rotacionales)*

La tabla de definición de rótulas plásticas o resortes rotacionales tipo contiene 7 columnas y tantas filas como rótulas plásticas se especifiquen en la hoja "*TB_GEN*". En esta tabla deben especificarse los parámetros de cada resorte rotacional no lineal tipo que posea la estructura, de acuerdo con la formulación del modelo de Bouc – Wen para resortes asimétricos o con propiedades diferentes en cada dirección, descrita en el capítulo de análisis dinámico inelástico.

Específicamente deben introducirse las rigideces elásticas y de fluencia de cada resorte, los momentos y las rotaciones de fluencia positivos y negativos y el coeficiente n que gobierna la variación de la relación $M - \theta$ del rango elástico al inelástico en la etapa de carga.

La tabla de asignación de rótulas plásticas o resortes rotacionales a elementos posee un número de filas igual al número de elementos y 4 columnas, similar a la tabla de rigidez inicial de conexiones parcialmente rígidas. La tabla contiene los indicadores de los resortes rotacionales tipo asignados en los extremos de cada elemento. Un valor igual a cero indica que en el correspondiente grado de libertad del elemento no se ha asignado ningún resorte rotacional, mientras que un valor mayor o igual a uno indica que en ese grado de libertad se encuentra asociado el resorte rotacional tipo indicado en la tabla.

Las dos anteriores tablas se requieren cuando se utiliza el método de análisis dinámico inelástico. En este caso es importante indicar que las rigideces iniciales de las conexiones en los extremos de los elementos deben coincidir con las rigideces elásticas de los resortes rotacionales.

Finalmente estas tablas se leen de la hoja "*TB_ROT*" y se almacenan en las matrices *ROT()* y *ARE()* respectivamente. La Figura 6.24 muestra ara una estructura cualquiera la parte de las tablas que se leen desde el código principal.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

DEFINICIÓN DE RÓTULAS PLÁSTICAS TIPO

Rótula tipo	K_θ	K_y	$F_y^{(+)}$	$F_y^{(-)}$	$u_y^{(+)}$	$u_y^{(-)}$	n
1	5.00E+04	1.00E+02	500	500	1.00E-02	1.00E-02	5

ROT ()

ASIGNACION RÓTULAS PLÁSTICAS A ELEMENTOS

Elemento N°	$\theta_y^{(r)}$	$\theta_z^{(r)}$	$\theta_y^{(s)}$	$\theta_z^{(s)}$
1	000	001	000	000
2	000	001	000	000
3	000	000	000	000
4	000	000	000	000
5	000	000	000	000
6	000	000	000	000

ARE ()

Figura 6.24. Ejemplo de las tablas *ROT()* y *ARE()*

- *Tablas de cargas distribuidas en la luz de los elementos*

La tabla de tipos de carga distribuida almacena los elementos a los que se aplican cargas iniciales distribuidas en la luz y el tipo de carga asociada a cada elemento cargado. La tabla consta de dos columnas, en la primera se almacena el indicador de cada elemento y en la segunda se almacena el tipo de carga aplicado. La cantidad de filas coincide con la cantidad de cargas distribuidas especificadas en la hoja "*TB_GEN*".

El programa dispone de seis tipos de carga distribuida (*uniforme y triangular*), las tres primeras actúan en dirección paralela al eje $\bar{y}$ local y las tres restantes actúan en dirección paralela al eje $\bar{z}$ local del elemento. La formulación correspondiente se describió en el capítulo de análisis dinámico inelástico.

La tabla de valores de carga es de una columna y tiene el mismo número de filas que la tabla anterior, aunque esta almacena el valor numérico (*con signo*) de la carga distribuida que se aplica a cada elemento. Para los tipos de carga disponible, el signo de cada carga depende de la dirección en que se aplique respecto a los ejes locales del elemento.

Estas dos tablas se leen de la hoja "*TB_WEL*" y se almacenan en las matrices *TWE()* y *WEL()* respectivamente. La Figura 6.25 muestra para una estructura cualquiera la parte de las tablas que se leen desde el código principal.

CARGAS DISTRIBUIDAS SOBRE ELEMENTOS

Carga N°	Elemento	Tipo de carga	Valor de la carga
1	3	001	-36.5
2	4	001	-36.5
3	5	001	-36.5

TWE () WEL ()

Figura 6.25. Ejemplo de las tablas *TWE()* y *WEL()*

- *Tablas de ejes de vigas y columnas*

Las dos tablas que se describen a continuación permiten agrupar los elementos de la estructura en conjuntos de elementos conectados secuencialmente, sobre los cuales es posible visualizar más claramente cantidades como por ejemplo diagramas de acciones internas (*cortante, momento, etc*) o derivas.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La tabla de coordenadas de ejes de columnas contiene las coordenadas x, y de los nudos que pertenecen a un mismo eje de columnas. La tabla contiene dos columnas, una para cada coordenada, y tantas filas como ejes de columnas se especifican en la hoja "*TB_GEN*".

La tabla de elementos asociados a ejes de vigas almacena la cantidad y la numeración de los elementos que están asociados a un mismo eje de vigas. Los elementos deben están conectados de forma secuencial y la orientación del eje $\bar{x}$ local debe ser la misma. La tabla posee tantas filas como ejes de viga se hayan especificado y tantas columnas como elementos tenga la estructura. En la primera columna se debe introducir la cantidad de elementos que componen el eje y en las columnas restantes los indicadores de los elementos que pertenecen a cada eje, dispuestos en forma secuencial y ordenada.

Estas dos tablas se leen de la hoja "*TB_EJE*" y se almacenan en las matrices *CXY()* y *VIG()* respectivamente. La Figura 6.26 muestra para una estructura cualquiera la parte de las tablas que se leen desde el código principal.

COORDENADAS DE EJES DE COLUMNAS

EJE	X m	Y m
1	0.000	0.000
2	6.000	0.000
3	12.000	0.000

CXY ()

TABLA DE ELEMENTOS ASOCIADOS A EJES DE VIGAS

Viga	N° de elementos	Elementos					
1	6	13	14	15	16	17	18
2	6	19	20	21	22	23	24
3	6	25	26	27	28	29	30
4	6	31	32	33	34	35	36

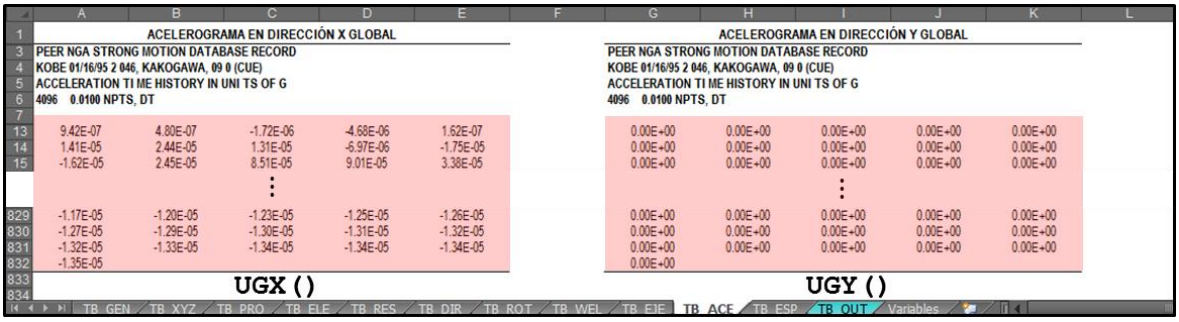
VIG ()

Figura 6.26. Ejemplo de las tablas *CXY()* y *VIG()*

- *Tablas de aceleración de la base*

Las tablas de aceleración de la base en direcciones x e y global se toman de un registro acelerográfico y deben introducirse en la hoja "*TB_ACE*" en dos tablas de 5 columnas y tantas filas como requiera la cantidad de datos de cada acelerograma.

Los datos deben quedar de forma secuencial y por filas, de forma que cuando se completen los cinco datos de una fila se continúe con la siguiente, formato que por lo general manejan las bases de datos consultadas (PEER). Estas tablas se almacenan en las matrices *UGX()* y *UGY()* respectivamente, como se muestra en la Figura 6.27.



ACELEROGRAMA EN DIRECCIÓN X GLOBAL

PEER NGA STRONG MOTION DATABASE RECORD
KOBE 01/16/95 2 046, KAKOGAWA, 09 0 (CUE)
ACCELERATION TI ME HISTORY IN UNI TS OF G
4096 0.0100 NPTS, DT

9.42E-07	4.80E-07	-1.72E-06	-4.68E-06	1.62E-07
1.41E-05	2.44E-05	1.31E-05	-6.97E-06	-1.75E-05
-1.62E-05	2.45E-05	8.51E-05	9.01E-05	3.38E-05
⋮				
-1.17E-05	-1.20E-05	-1.23E-05	-1.25E-05	-1.26E-05
-1.27E-05	-1.29E-05	-1.30E-05	-1.31E-05	-1.32E-05
-1.32E-05	-1.33E-05	-1.34E-05	-1.34E-05	-1.34E-05
-1.35E-05				

UGX ()

ACELEROGRAMA EN DIRECCIÓN Y GLOBAL

PEER NGA STRONG MOTION DATABASE RECORD
KOBE 01/16/95 2 046, KAKOGAWA, 09 0 (CUE)
ACCELERATION TI ME HISTORY IN UNI TS OF G
4096 0.0100 NPTS, DT

0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
⋮				
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00				

UGY ()

Figura 6.27. Ejemplo de las tablas *UGX()* y *UGY()*

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

- *Tabla de espectro de pseudo – aceleraciones*

La ultima tabla de datos corresponde al espectro de pseudo – aceleraciones. Esta tabla esta compuesta de dos columnas, en la primera se almacena el rango de periodos de vibración considerado para su construcción y en la segunda las pseudo – aceleraciones correspondientes, expresadas como fracción de la aceleración de la gravedad. El número de filas de la tabla coincide con el número de datos especificado en la hoja "*TB_GEN*".

Esta tabla se lee de la hoja "*TB_ESP*" y se almacena en la matriz *ESP()*. La Figura 6.28 muestra la parte de la tabla que se lee desde el código principal.

	A	B	C
1	ESPECTRO DE PSEUDO-ACELERACIONES		
2			
3	Dato N°	t (seg)	S_a / g
4	1	0.000	0.731
5	2	0.560	0.731
6	3	0.804	0.507
7	4	1.048	0.389
8	5	1.292	0.315
9	6	1.536	0.265
10	7	1.780	0.229
11	8	2.024	0.201
12	9	2.268	0.179
13	10	2.512	0.162
14	11	2.756	0.148
15	12	3.000	0.136
16			
17		ESP ()	

Figura 6.28. Ejemplo de las tabla *ESP()*

La Tabla 6.5 resume las características de las tablas de datos leídas desde las diferentes hojas de datos, como se describió previamente.

N°	Variable	Contenido	Tipo	Hoja de datos	Tipo de análisis en el que se requiere la tabla
1	XYZ ()	Tabla de coordenadas de nudos y de puntos auxiliares	Real	TB_XYZ	Cualquiera
2	MAT ()	Tabla de definición de materiales tipo	Real	TB_PRO	Cualquiera
3	SEC ()	Tabla de definición de secciones transversales tipo	Real	TB_PRO	Cualquiera
4	ELE ()	Tabla de conectividades entre elementos	Entero	TB_ELE	Cualquiera
5	PAU ()	Tabla de asignación de puntos auxiliares a elementos	Entero	TB_ELE	Cualquiera
6	AMS ()	Tabla de asignación de materiales y secciones tipo a elementos	Entero	TB_ELE	Cualquiera
7	RIG ()	Tabla de asignación de rigidez tipos a elementos	Entero	TB_ELE	Cualquiera
8	CTS ()	Tabla de número de cortes por elemento	Entero	TB_ELE	Cualquiera
9	JRS ()	Tabla de rigideces de conexiones elásticas entre elementos	Real	TB_ELE	Cualquiera
10	MRE ()	Tabla de clasificación de grados de libertad	Entero	TB_RES	Cualquiera
11	MDC ()	Tabla de desplazamientos conocidos	Real	TB_RES	Cualquiera
12	DRX ()	Tabla de coordenadas de centros de masa	Real	TB_DIR	Cualquiera
13	MAS ()	Tabla de masas traslacional y rotacional por diafragma	Real	TB_DIR	Cualquiera
14	ROT ()	Tabla de definición de rótulas plásticas	Real	TB_ROT	Análisis dinámico inelástico
15	ARE ()	Tabla de asignación de rótulas plásticas a elementos	Entero	TB_ROT	Análisis dinámico inelástico
16	TWE ()	Tabla de tipos de carga distribuida aplicada a elementos	Entero	TB_WEL	Análisis dinámico inelástico

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

N°	Variable	Contenido	Tipo	Hoja de datos	Tipo de análisis en el que se requiere la tabla

Tabla 6.5. Tablas leídas de las diferentes hojas de datos

17	WEL()	Tabla de valores numéricos de cargas distribuidas	Real	TB_WEL	Análisis dinámico inelástico
18	CXY()	Tabla de coordenadas X,Y de ejes de columnas	Real	TB_EJE	Cualquiera
19	VIG()	Tabla de elementos asociados a ejes de vigas	Entero	TB_EJE	Cualquiera
20	UGX()	Tabla de aceleración de la base en dirección X global	Real	TB_ACE	Análisis modal cronológico y análisis dinámico inelástico
21	UGY()	Tabla de aceleración de la base en dirección Y global	Real	TB_ACE	Análisis modal cronológico y análisis dinámico inelástico
22	ESP()	Tabla de espectro de pseudo-aceleraciones	Real	TB_ESP	Análisis modal espectral

Tabla 6.5. Tablas leídas de las diferentes hojas de datos (*cont.*)

En la Figura 6.29 se muestran las líneas de la subrutina principal con las que se leen las diferentes tablas de datos previamente descritas. Se utilizan las subrutinas *EDLECI*, *EDLECR*, *EDTABI* y *EDTABR* originales de *PEFiCA*.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Lectura de tablas de datos

'tabla de coordenadas de nudos y de puntos auxiliares (XYZ)
EDLECR "TB_XYZ", 5, 2, XYZ(), NNUD + NPAU, 3

'tablas de definición de materiales y secciones tipo (XYZ)
EDLECR "TB_PRO", 5, 2, MAT(), NMAT, 2              'definición de materiales tipo (MAT)
EDLECR "TB_PRO", 5, 12, SEC(), NSEC, 5             'definición de secciones tipo (SEC)

'tablas de geometría (ELE, PAU, AMS, RIG, CTS, JRS)
EDLECI "TB_ELE", 5, 2, ELE(), NELE, NNUE           'conectividades (ELE)
EDLECI "TB_ELE", 5, 4, PAU(), NELE, 1              'puntos auxiliares asignados a elementos (PAU)
EDLECI "TB_ELE", 5, 7, AMS(), NELE, 2              'indicadores de material y sección asignados a elementos (AMS)
EDLECI "TB_ELE", 5, 9, RIG(), NELE, 1              'indicadores de rigidez por elemento (RIG)
EDLECI "TB_ELE", 5, 10, CTS(), NELE, 1             'número de cortes por elemento (CTS)
EDLECR "TB_ELE", 5, 13, JRS(), NELE, 4             'rigideces de conexiones elásticas (JRS)

'tablas de clasificación de grados de libertad y desplazamientos conocidos (MRE, MDC)
EDTABI "TB_RES", 5, 2, MRE(), NNUD, NGLN           'clasificación de grados de libertad (MRE)
EDTABR "TB_RES", 5, 10, MDC(), NNUD, NGLN          'desplazamientos conocidos (MDC)

'tablas de masas traslacionales y rotacionales de diafragmas (DRX, MAS)
EDLECR "TB_DIR", 5, 2, DRX(), NDIR, 3              'coordenadas de centros de masa (DRX)
EDLECR "TB_DIR", 5, 5, MAS(), NDIR, 2              'masas traslacionales y rotacionales por diafragma (MAS)

'tablas de rótulas plásticas (ROT, ARE)
EDLECR "TB_ROT", 5, 2, ROT(), NROT, 7              'definición de rótulas plásticas (ROT)
EDLECI "TB_ROT", 5, 11, ARE(), NELE, 4             'asignación de rótulas plásticas a elementos (ARE)

'tablas de cargas iniciales distribuidas (TWE, WEL)
EDLECI "TB_WEL", 5, 2, TWE(), NCAR, 2              'tabla de tipo de carga distribuida aplicada a elementos (TWE)
EDLECR "TB_WEL", 5, 4, WEL(), NCAR, 1              'valores de carga distribuida aplicadas a elementos (WEL)

'tablas de ejes de vigas y columnas (CXY, VIG)
EDLECR "TB_EJE", 5, 2, CXY(), NCOL, 2              'coordenadas XY de ejes de columnas (CXY)
EDLECI "TB_EJE", 5, 6, VIG(), NVIG, NELE           'elementos asociados a ejes de vigas (VIG)

'tablas de aceleración de la base (UGX, UGY)
EDLECR "TB_ACE", 13, 1, UGX(), (NACE / 5 + 1), 5   'aceleración de la base en dirección X en formato base (UGX)
EDLECR "TB_ACE", 13, 7, UGY(), (NACE / 5 + 1), 5   'aceleración de la base en dirección Y en formato base (UGY)

'tabla de espectro de pseudo-aceleraciones (ESP)
EDLECR "TB_ESP", 4, 2, ESP(), NPES, 2              'espectro de pseudo-aceleraciones (ESP)

Debug.Print "Lectura de tablas de datos", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.29. Lectura de tablas de datos

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

6.3.3. Cantidad de grados de libertad por tipo

En esta etapa se calculan unos parámetros escalares de tipo entero relacionados con las cantidades de grados de libertad estáticos y las cantidades de grados de libertad dinámicos que posee la estructura. Específicamente se obtienen las siguientes cantidades:

- Número de grados de libertad total.
- Número de grados de libertad conocidos
- Número de grados de libertad desconocidos condensados
- Número de grados de libertad desconocidos de interés
- Número de grados de libertad desconocidos
- Número de grados de libertad considerados por diafragma
- Número de grados de libertad dinámicos de la estructura

Para calcular las cantidades de grados de libertad conocidos, condensados y de interés se utiliza la subrutina *MTREPI()* original de *PEFiCA* con la que se calcula el número de veces que aparecen los indicadores *001, 002* y *003* en la matriz *MRE()*, dando como resultado las cantidades requeridas. La Figura 6.30 muestra las líneas del código principal con las que se realizan los cálculos indicados previamente.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Cantidad de grados de libertad por tipo

'número de grados de libertad total (NGLT)
NGLT = NNUD * NGLN

'número de grados de libertad conocidos (NGLC)
NGLC = MTREPI(MRE(), 1)

'número de grados de libertad desconocidos condensados (NGLO)
NGLO = MTREPI(MRE(), 2)

'número de grados de libertad desconocidos de interés (NGLI)
NGLI = MTREPI(MRE(), 3)

'número de grados de libertad desconocidos (NGLD)
NGLD = NGLI + NGLO

'número de grados de libertad considerados por diafragma (NGDR)
NGDR = 2 * NDIM - 3

'número de grados de libertad dinámicos de la estructura (NDIN)
NDIN = NDIR * NGDR

Debug.Print "Cantidad de grados de libertad por tipo", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.30. Cantidad de grados de libertad por tipo

6.3.4. Creación de matrices auxiliares

En esta etapa se calcula un parámetro escalar adicional y se crean 13 tablas auxiliares y necesarias para el funcionamiento de las subrutinas con las que posteriormente se plantean y resuelven las ecuaciones de movimiento y se obtiene la respuesta de la estructura, utilizando alguno de los métodos disponibles de análisis dinámico.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

Para su creación se utilizan las subrutinas *NGLNCD*, *NGLELE*, *ORNDIR*, *TRDIRZ* y *MTTRAN* existentes en *PEFiCA*, y las subrutinas nuevas *AEMPAE*, *AEDCON*, *NGLROT* y *AEMPRO*. A continuación se describen las características más relevantes de cada una de las tablas auxiliares.

- *Tabla de numeración de grados de libertad por nudo*

Esta tabla contiene la numeración que el programa asigna a los seis grados de libertad de cada nudo de la estructura mediante la subrutina *NGLNCD*. Se almacena en la variable entera *MGL()* de seis columnas (*grados de libertad por nudo*) y tantas filas como nudos posea la estructura, de forma que en cada fila se almacena la numeración de los grados de libertad de un mismo nudo.

- *Tabla de incidencias*

Esta tabla se construye con la subrutina *NGLELE* y contiene la numeración de los doce grados de libertad de cada elemento. Se almacena en la variable entera *INC()* de 12 columnas y tantas filas como elementos posea la estructura, de forma que en cada fila se almacena la numeración de los grados de libertad de un mismo elemento en el formato general, es decir, comenzando por los seis grados del nudo inicial $u_x^{(i)}, u_y^{(i)}, u_z^{(i)}, \theta_x^{(i)}, \theta_y^{(i)}, \theta_z^{(i)}$ y terminando con los seis grados de libertad del nudo final $u_x^{(j)}, u_y^{(j)}, u_z^{(j)}, \theta_x^{(j)}, \theta_y^{(j)}, \theta_z^{(j)}$.

- *Tablas de nudos y elementos asociados a cada eje de columnas*

Con la subrutina *ORECOL* se construyen tres tablas auxiliares de tipo entero en las que se almacenan las cantidades y la numeración de los nudos y los elementos asociados a cada eje de columnas. En las tres tablas el número de filas es igual al número de ejes de columnas especificado en la hoja "*TB_GEN*", de forma que a cada fila le corresponde la información asociada a un mismo eje.

La primera tabla almacenada en la matriz *NEC()*, contiene en una sola columna la cantidad de nudos asociados a cada eje de columnas. La segunda tabla se almacena en la matriz *NNE()*, de tantas columnas como nudos posea la estructura, y contiene la numeración de los nudos de cada eje en forma secuencial, comenzando por el más elevado (*coordenada z mayor*) y terminando por el más bajo (*coordenada z menor*). La tercera tabla se almacena en la matriz *COL()* y contiene simultáneamente la cantidad y la numeración de los elementos asociados a cada eje de columnas, ordenados de menor a mayor, similar a como contiene la información la tabla *VIG()*. El número de columnas es igual al número de elementos más uno, dado que en la primera columna se almacenan las cantidades de elementos de cada eje.

- *Tablas de propiedades de elementos*

La tabla de propiedades de elementos organiza las propiedades del material, de la sección y de la rigidez de las conexiones asociada a cada elemento. Se crea con la subrutina *AEMPAE* y se almacena en la matriz *PAE()*, cuyo tamaño es de 11 columnas y tantas filas como

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

elementos tenga la estructura. En las dos primeras columnas se almacenan el módulo de Young E y la relación de Poisson v del material, en las siguientes 5 columnas se almacenan el área A, los momentos de inercia $I_{\bar{z}}$ e $I_{\bar{y}}$, la constante de torsión J y el factor de forma f_s de la sección, y en las cuatro últimas columnas se almacenan las rigideces iniciales de las conexiones a momento de cada extremo $J_{\bar{y}i}, J_{\bar{z}i}, J_{\bar{y}j}$ y $J_{\bar{z}j}$.

- *Subvector de desplazamientos conocidos*

El subvector de desplazamientos conocidos se crea con la subrutina *AEDCON* y se almacena en la matriz real *DBB()*. Este vector posee una sola columna y un número de filas igual al número de grados de libertad conocidos, previamente calculado. Los coeficientes de este vector son constantes, no cambian en ningún instante de tiempo.

- *Tablas de nudos asociados a cada diafragma rígido*

Con la subrutina *ORNDIR* se construyen dos tablas enteras en las que se almacenan las cantidades y la numeración de los nudos asociados a cada diafragma rígido. El número de filas de cada tabla es igual al número de diafragmas especificado en la hoja "*TB_GEN*", de forma que a cada fila le corresponde la información asociada a un mismo diafragma.

La primera tabla almacenada en la matriz *NDR()*, contiene en una sola columna la cantidad de nudos asociados a cada diafragma. La segunda tabla se almacena en la matriz *CND()*, de tantas columnas como nudos posea la estructura, y contiene la numeración de los nudos asociados a cada diafragma. La identificación de los nudos que pertenecen a cada diafragma se realiza comparando las coordenadas z de los nudos con las coordenadas en z de los centros de masa.

- *Matriz de asociación de desplazamientos*

La matriz de asociación de desplazamientos $[A]$ requerida para la asociación de grados de libertad se construye con la subrutina *TRDIRZ*. Esta se almacena en la matriz real *TDR()*, el número de filas coincide con el número de grados de libertad de interés y el número de columnas coincide con el número de grados de libertad dinámicos de la estructura.

La matriz transpuesta de asociación de desplazamientos $[B] = [A]^T$ se construye con la subrutina *MTTRAN* y se almacena en la matriz real *TDT()*.

- *Tablas de numeración y de propiedades de rótulas plásticas y cantidad de rótulas plásticas de la estructura*

La tabla de numeración de rótulas o resortes rotacionales contiene la numeración que UNDIN asigna a cada rótula plástica o resorte rotacional no lineal de la estructura. Se crea con la subrutina *NGLROT* y se almacena en la matriz entera *NRO()*, cuyo número de filas coincide con el número de elementos y el número de columnas es 4, teniendo en cuenta que se consideran rótulas plásticas a flexión en los dos extremos y en cada plano del elemento.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

Esta misma subrutina calcula la cantidad total de rótulas plásticas (*o resortes rotacionales*) asignadas a la estructura y almacena el resultado en la variable *ROTE*.

La tabla de propiedades de rótulas de la estructura se crea con la subrutina *AEMPRO* y se almacena en la matriz real *PRO()*. Esta tabla organiza la misma información que contiene la tabla *ROT()* para las rótulas tipo, pero ahora para cada una de las rótulas plásticas asignadas a la estructura. Por lo tanto el número de columnas de *PRO()* es 7, igual que en *ROT()*, y el número de filas es igual al número total de rótulas plásticas de la estructura.

Las dos tablas y la variable *ROTE* que se describieron previamente se crean solo cuando se utiliza el método de análisis dinámico inelástico.

La Tabla 6.6 resume las características de las tablas auxiliares creadas en esta etapa. Se indica el nombre de la variable, el tipo, el contenido y la subrutina con la que se crea.

N°	Variable	Contenido	Tipo	Subrutina
1	`MGL()`	Tabla de numeración de grados de libertad por nudo	Entero	`NGLNCD`
2	`INC()`	Tabla de numeración de grados de libertad por elemento (*tabla de incidencias*)	Entero	`NGLELE`
3	`NEC()`	Tabla de cantidad de nudos asociados a cada eje de columnas	Entero	`ORECOL`
4	`NNE()`	Tabla de nudos asociados a cada eje de columnas	Entero	`ORECOL`
5	`COL()`	Tabla de elementos asociados a cada eje de columnas	Entero	`ORECOL`
6	`PAE()`	Tabla de propiedades de elementos	Real	`AEMPAE`
7	`DBB()`	Subvector de desplazamientos conocidos (*apoyos*)	Real	`AEDCON`
8	`NDR()`	Tabla de nudos asociados a cada diafragma rígido	Entero	`ORNDIR`
9	`CND()`	Tabla de cantidad de nudos asociados a cada diafragma rígido	Entero	`ORNDIR`
10	`TDR()`	Matriz de asociación de desplazamientos (*matriz geométrica*)	Real	`TRDIRZ`
11	`TDT()`	Matriz de asociación de desplazamientos transpuesta	Real	`MTTRAN`
12	`NRO()`	Tabla de numeración de rótulas plásticas	Entero	`NGLROT`
13	`ROTE`	Cantidad de rótulas plásticas de la estructura	Entero	`NGLROT`
14	`PRO()`	Tabla de propiedades de rótulas plásticas de la estructura	Real	`AEMPRO`

Tabla 6.6. Matrices auxiliares

La Figura 6.31 muestra las líneas del código principal con las que se construyen las tablas previamente descritas.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Creación de matrices auxiliares

'matriz de numeración de grados de libertad por nudo (MGL)
NGLNCD MGL(), MRE(), NGLI, DRX(), XYZ()

'matriz de grados de libertad por elemento o matriz de incidencias (INC)
NGLELE INC(), MGL(), ELE()

'tablas de nudos y de elementos asociados a ejes de columnas (NEC, NNE)
ORECOL NEC(), NNE(), COL(), CXY(), XYZ(), ELE(), NELE

'tabla de propiedades por elemento (PAE)
AEMPAE MAT(), SEC(), AMS(), JRS(), PAE()

'vector columna de desplazamientos conocidos (DBB)
AEDCON NGLC, MGL(), MDC(), DBB()

'tablas de nudos asociados a cada diafragma rígido (NDR, CND)
ORNDIR DRX(), XYZ(), NDR(), CND()

'matriz de asociación de desplazamientos o matriz geométrica [A] (TDR)
TRDIRZ TDR(), XYZ(), DRX(), NDR(), CND(), NGDR

'matriz geométrica transpuesta (TDT)
MTTRAN TDR(), TDT()

'matriz de numeración de rótulas y matriz de propiedades de rótulas (NRO, PRO)
If IDMA = "DINÁMICO INELÁSTICO" Then
    'matriz de numeración de rótulas (NRO)
    NGLROT ARE(), NRO(), ROTE

    'matriz de propiedades de rótulas (PRO)
    AEMPRO ROTE, ARE(), NRO(), ROT(), PRO()
End If

Debug.Print "Creación de matrices auxiliares", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.31. Creación de matrices auxiliares

6.4. Parte III: Matrices de masa, rigidez y amortiguamiento. Análisis modal

A continuación se describen las líneas de la subrutina principal con las que UNDIN construye las matrices de masa y rigidez lateral inicial (*es decir, en* $t = 0$), calcula los periodos, las frecuencias y los modos de vibración natural, las matrices modales de masa, rigidez y amortiguamiento, la matriz de amortiguamiento por superposición de matrices modales, y finalmente, calcula los factores de participación y la masa modal efectiva asociada a cada modo de vibración de la estructura.

Estas cantidades posteriormente se utilizan en los métodos de análisis modal cronológico y análisis modal espectral, aunque algunas de ellas también se requieren en el método de análisis dinámico inelástico.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

6.4.1. Matriz de masa concentrada

La matriz de masa concentrada se construye con la subrutina *ADMASA* y se almacena en la matriz real *MDR()*. Esta matriz es cuadrada y su tamaño coincide con el número total de grados de libertad dinámicos de la estructura. La matriz creada es diagonal, puesto que se basa en la formulación descrita en los Capítulos 2 y 3, en la que se plantean las ecuaciones de movimiento de cada diafragma en el respectivo centro de masa. La Figura 6.32 muestra la línea del código principal con las que se construye *MDR()*.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Matriz de masa

'matriz de masa concentrada en centros de masa (MDR)
ADMASA NDIR, NGDR, MAS(), MDR()

Debug.Print "Matriz de masa", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.32. Matriz de masa concentrada

6.4.2. Matriz de rigidez lateral

De acuerdo con el Capítulo 3 la matriz de rigidez lateral $[k_\phi]$ necesaria para el análisis dinámico, se crea a partir de la matriz de rigidez condensada a los grados de libertad de interés $[\tilde{k}_{\alpha\alpha}]$. Esta última se crea a partir de la matriz de rigidez general $[k]$ que considera los grados de libertad estáticos de la estructura. En esta etapa se indica cómo se construyen las tres matrices $[k]$, $[\tilde{k}_{\alpha\alpha}]$ e $[k_\phi]$ indicadas previamente.

La matriz de rigidez general $[k]$ se construye con la subrutina *AEKGLT*, en la que se crean y ensamblan las matrices de rigidez en coordenadas globales de los elementos estructurales. El resultado se almacena en la matriz real y cuadrada *KGL()* de tamaño igual al número de grados de libertad estáticos.

La matriz de rigidez condensada a los grados de libertad de interés $[\tilde{k}_{\alpha\alpha}]$ se construye con la subrutina *AEKGLI* siguiendo la formulación matemática descrita en el Capítulo 3. Como resultado se crea la matriz *KDI()*, real, cuadrada y de tamaño igual al número de grados de libertad de interés para el análisis dinámico. Asi mismo la subrutina *AEKGLI* entrega las submatrices $[k_{\alpha\alpha}], [k_{\alpha\gamma}], [k_{\gamma\alpha}]$ y $[k_{\gamma\gamma}]^{-1}$ de $[k]$ que se requieren posteriormente para otras operaciones ($[k_{\gamma\gamma}]$ *queda invertida*).

La matriz $[k_\phi]$ que corresponde al triple producto $[A]^T[\tilde{k}_{\alpha\alpha}][A]$ se calcula con la subrutina *MTMUL3* original de PEFiCA.

En esta etapa también se calculan las coordenadas de los centros de rigidez de cada diafragma rígido de una estructura espacial, siguiendo la formulación descrita en el mismo capítulo. Para tal fin se utiliza la subrutina *AECRIG*, con la que se calculan y almacenan en la tabla *DRX()* las coordenadas de los centros de rigidez y las excentricidades entre los cen-

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

tros de masa y de rigidez. A la matriz *DRX()* se le adicionan 4 columnas al final con los resultados obtenidos.

La Figura 6.33 muestra las líneas del código principal con las que se construyen las matrices *KGL(), KDI()* y *KDP(),* y las líneas con las que se modifica la matriz *DRX().*

```
'-----------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------
'Matriz de rigidez lateral

'matriz de rigidez considerando todos los grados de libertad (KGL)
AEKGLT NELE, NGLT, ELE(), INC(), PAU(), RIG(), XYZ(), PAE(), KGL()

'matriz de rigidez condensada a grados de libertad de interés (KKA, KAG, KGA, KGG, KDI)
AEKGLI NGLI, NGLO, KGL(), KAA(), KAG(), KGA(), KGG(), KDI()

'matriz de rigidez condensada a los grados de libertad de diafragmas (KDP)
MTMUL3 TDR(), KDI(), KDP()

'tabla de coordenadas de centros de masa y de rigidez para pórticos espaciales (DRX)
If NDIM = 3 Then
    AECRIG KDP(), DRX()
End If

Debug.Print "Matriz de rigidez considerando todos los grados de libertad", (Timer - TINI)
Debug.Print "Matriz de rigidez condensada a grados de libertad de interés", (Timer - TINI)
Debug.Print "Matriz de rigidez lateral", (Timer - TINI)
Debug.Print "Coordenadas de centros de rigidez", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------
```

Figura 6.33. Matriz de rigidez lateral

6.4.3. Frecuencias, periodos y modos de vibración natural

Las frecuencias de vibración al cuadrado y los modos de vibración natural ortonormalizados de la estructura se calculan con la subrutina *VPJACO*, la cual utiliza el método de Jacobi generalizado para resolver el problema de valores propios.

Esta subrutina es original de PEFiCA, pero se ha complementado para que en la matriz $[\Phi]$ y en el vector $\{\Omega^2\}$ los modos y las frecuencias de vibración queden en el orden ascendente, dado que el método de Jacobi generalizado no lo garantiza.

La subrutina mencionada crea la matriz cuadrada *MOV()* de tamaño igual al número de grados de libertad dinámicos, el cual coincide con el número de modos de vibración. A cada columna de *MOV()* le corresponde un modo de vibración. Por otro lado la misma subrutina crea el vector columna *VAP()* en el que el número de filas coincide con el número de modos de vibración, y que almacena las frecuencias de vibración natural al cuadrado.

Con la subrutina *MTTRAN*, original de PEFiCA, se crea la matriz $[\Phi]^T$ requerida en etapas de cálculo posteriores.

Finalmente con la subrutina *ADPEFR* se crean los vectores columna *FRN()* y *PER(),* en los que se almacenan las frecuencias y los periodos de vibración natural de la estructura asociados a cada modo de vibración. Su tamaño es idéntico al del vector *VAP().*

La Figura 6.34 muestra las líneas del código principal con las que se construyen las matrices *MOV()* y *MOT()* y los vectores *VAP(), FRN()* y *PER()* previamente descritos.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'-----------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------
'Frecuencias, periodos y modos de vibración ortonormalizados

'número de modos de vibración (NMOD)
NMOD = NDIR * NGDR

'frecuencias al cuadrado y modos de vibración ortonormalizados obtenidos resolviendo el problema de valores propios
'por el método de Jacobi Generalizado (MOV, VAP)
VPJACO KDP(), MDR(), MOV(), VAP(), ITER

'matriz transpuesta de modos de vibración (MOT)
MTTRAN MOV(), MOT()

'frecuencias y periodos de vibración natural (FRN, PER)
ADPEFR VAP(), FRN(), PER()

Debug.Print "Frecuencias, periodos y modos de vibración ortonormalizados", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------
```

Figura 6.34. Frecuencias, periodos y modos de vibración natural

6.4.4. Matrices modales de masa, rigidez y amortiguamiento

Las matrices modales de masa y rigidez se construyen con la subrutina *MTMUL3* con las que se realizan los triples productos $[M] = [\Phi]^T[m][\Phi]$ y $[K] = [\Phi]^T\left[k_\phi\right][\Phi]$. El único uso de estas matrices es verificar que la primera de ellas corresponde a la matriz identidad, y que la segunda es diagonal con cada coeficiente igual a la frecuencia de vibración al cuadrado, dado que los modos de vibración obtenidos con *VPJACO* son ortonormalizados.

La matriz modal de amortiguamiento $[C]$ se construye con la subrutina *ADMMAM* siguiendo la formulación del Capítulo 3. Las matrices $[M], [C]\ y\ [K]$ se almacenan en las matrices cuadradas *MMM()*, *MMA()* y *MMK()* respectivamente. La Figura 6.35 muestra las líneas del código principal con las que se construyen las matrices indicadas.

```
'-----------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------
'Matrices modales de masa, rigidez y amortiguamiento

'matrices modal de masa y matriz modal de rigidez (MMM, MMK)
MTMUL3 MOV(), MDR(), MMM()
MTMUL3 MOV(), KDP(), MMK()

'matriz modal de amortiguamiento (MMA)
ADMMAM AMVI, FRN(), MMA()

Debug.Print "Matrices modales de masa, rigidez y amortiguamiento", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------
```

Figura 6.35. Matrices modales de masa, rigidez y amortiguamiento

6.4.5. Matriz de amortiguamiento por superposición de matrices modales

La matriz de amortiguamiento que se calcula en esta etapa, se utiliza solamente para la verificación de las ecuaciones de movimiento y de balance de energía, cuando la respuesta de la estructura se obtiene en cada instante de tiempo por análisis modal cronológico. En este

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

caso la matriz de amortiguamiento se construye con el criterio de superposición de matrices modales descrito en el Capítulo 3.

Para tal fin se utiliza la subrutina *ADAMRS*, que crea la matriz cuadrada *AMR()* de tamaño igual al número de modos de vibración natural. La Figura 6.36 muestra las líneas del código principal con las que se construye *AMR()*.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Matriz de amortiguamiento por superposición de matrices modales (solo para el caso de análisis modal cronológico)
If IDMA = "MODAL CRONOLÓGICO" Then
    ADAMRS MMM(), MMA(), MOV(), MOT(), MDR(), AMR()
    Debug.Print "Matriz de amortiguamiento por superposición de matrices modales", (Timer - TINI)
End If
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.36. Matriz de amortiguamiento por superposición de matrices modales

6.4.6. Factores de participación y masa modal efectiva

La matriz de influencia $[\gamma]$ se crea con la subrutina *ADGAMA* y se almacena en la matriz real *GAM()*, con un numero de filas igual al número de grados de libertad dinámicos de la estructura, y con un número de columnas igual al número de grados de libertad por diafragma, el cual varía entre uno para pórticos planos y tres para pórticos espaciales.

La matriz de factores de participación $[\alpha]$ se calcula con la subrutina *ADALFA* realizando el triple producto $[\alpha] = [\Phi]^T[m][\gamma]$. Se almacena en la matriz real *ALF()*, cuyo tamaño coincide con el de la matriz de influencia.

Con la subrutina *ADMASM* se crean tres matrices, las dos primeras de tamaño igual al de la matriz de factores de participación, en las que se almacenan las masas modales efectivas asociadas a cada modo de vibración y a cada componente de aceleración de la base, y una tercera que sirve para verificar que la suma de las masas modales efectivas corresponde a la masa total de la estructura.

Las dos primeras se almacenan en las matrices reales *MME()* y *MMP()*, la primera contiene los valores absolutos de las masas modales y la segunda las muestra como fracción de la masa traslacional o rotacional total de la estructura. La tercera se almacena en la matriz real *MTE()*. Esta matriz es diagonal y su tamaño es igual al número de grados de libertad por diafragma (*1 o 3 dependiendo la estructura*). Cada coeficiente de la diagonal debe almacenar la masa traslacional o rotacional total de la estructura en cada dirección.

La Figura 6.37 muestra las líneas del código principal con las que se construyen las matrices *GAM()*, *ALF()*, *MME()*, *MMP()* y *MTE()*.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'-----------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------
'Factores de participación y masa modal efectiva

'matriz de influencia (GAM)
ADGAMA NDIR, NGDR, GAM()

'matriz de coeficientes de participación (ALF)
ADALFA MOV(), MDR(), GAM(), ALF()

'matrices de masa modal efectiva y de masa total de la estructura (MME, MMP, MTE)
ADMASM ALF(), MME(), MMP(), MTE()

Debug.Print "Factores de participación y masa modal efectiva", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------
```

Figura 6.37. Factores de participación y masa modal efectiva

6.5. Parte IV: Análisis modal cronológico

En esta etapa UNDIN calcula la variación en el tiempo de la respuesta de una estructura en el rango elástico y con amortiguamiento clásico, sometida a una o varias componentes de aceleración en la base.

A continuación se describe cómo se definen las acciones modales externas equivalentes al movimiento de la base, que constituyen los términos independientes de las ecuaciones modales desacopladas. Se describe como se calculan los desplazamientos, las velocidades y las aceleraciones en coordenadas generalizadas y a partir de estas, como se calculan los desplazamientos, las velocidades y las aceleraciones relativas a la base de los centros de masa. Se describe la forma en que a partir de los desplazamientos de los centros de masa se calculan con el programa los demás parámetros que representan la respuesta de la estructura en cada instante de tiempo (*desplazamientos de nudos, derivas, acciones equivalentes, acciones internas, etc.*).

Los vectores variables en el tiempo, cuya formulación se describió en el Capítulo 4, se tratan en el código como matrices de datos, en los que a cada columna le corresponde un instante de tiempo. Por ejemplo el vector de aceleración de la base $\{\ddot{u}_g(t)\}$ en el código corresponde a la matriz $[\ddot{u}_g]$ en la que cada fila corresponde a una componente de aceleración de la base y cada columna corresponde a un instante de tiempo.

El programa incluye el instante inicial ($t = 0$) en cada tabla de resultados dado que este no se incluye en los datos de los acelerogramas, por lo que el número de instantes de tiempo considerados en el análisis se aumenta en uno respecto al número de datos de cada acelerograma.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

6.5.1. Acciones modales externas equivalentes a la aceleración de la base

El vector de acciones modales externas equivalentes a la aceleración de la base en cada instante de tiempo $\{\bar{F}_{ext}(t)\}$, corresponde en el código a la matriz $[\bar{F}_{ext}] = [\Phi]^T[F_{ext}]$ donde $[F_{ext}] = -[m][\gamma]\left[\ddot{u}_g\right]$. La forma de obtener las matrices $[\Phi]^T$, $[m]$ y $[\gamma]$ ya mencionó, por lo tanto resta describir cómo construir la matriz $[\ddot{u}_g]$. A continuación se indica la forma como se obtienen las matrices $\left[\ddot{u}_g\right]$, $[F_{ext}]$ y $[\bar{F}_{ext}]$.

Con la subrutina *ADUGXY* se crea la matriz de aceleración de la base $\left[\ddot{u}_g\right]$ y se almacena en matriz real *MUO()*, cuyo número de filas coincide con el número de componentes de aceleración en la base consideradas en el análisis, dependiendo si el pórtico estudiado es plano o espacial. Cada columna de *MUO()* corresponde a un instante de tiempo.

La misma subrutina crea dos matrices reales denominadas *TME()* y *ACE()*, la primera es un vector fila que contiene los instantes de tiempo considerados en el análisis, mientras que la segunda es una tabla que almacena en la primera columna los datos de *TME()* (*es decir*, *TME() transpuesta*) y en las columnas restantes se almacenan los datos de cada acelerograma a partir de los cuales se crea $[\ddot{u}_g]$. El vector *TME()* se usa posteriormente para crear tablas de resultados que permiten graficar la variación de cada parámetro en función del tiempo, mientras que *ACE()* se utiliza para graficar los datos de los acelerogramas.

La matriz de acciones externas equivalentes a la aceleración de la base $[F_{ext}]$ se crea utilizando las subrutinas *MTPORE* y *MTMULT*, con las que se realiza el triple producto $-[m][\gamma][\ddot{u}_g]$. El resultado se almacena en la matriz real *FEX()* cuyo tamaño es similar al de la matriz *MUO()*.

Finalmente se utiliza la subrutina *MTMULT* para calcular la matriz de acciones modales externas equivalentes al movimiento de la base $[\bar{F}_{ext}] = [\Phi]^T[F_{ext}]$. El resultado se almacena en la matriz real *TEI()* de tamaño similar a *MUO()*.

La Figura 6.38 muestra las líneas del código principal con las que se construyen las matrices *ACE()*, *TME()*, *MUO()*, *FEX()* y *TEI()*.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Acciones modales externas equivalentes a la aceleración de la base

'tabla de acelerogramas, vector de instantes de tiempo y matriz de aceleración de la base (ACE, TME, MUO)
ADUGXY NACE, DTIM, FESX, FESY, NGDR, UGX(), UGY(), ACE(), TME(), MUO()

'matriz de acciones externas equivalentes a la aceleración de la base (FEX)
MTPORE MDR(), -1, TM1()
MTMULT TM1(), GAM(), TM2()
MTMULT TM2(), MUO(), FEX()

'matriz de acciones modales externas (TEI)
MTMULT MOT(), FEX(), TEI()

Debug.Print "Acciones modales externas equivalentes a la aceleración de la base", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.38. Acciones modales externas equivalentes a la aceleración de la base

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

6.5.2. Desplazamientos, velocidades y aceleraciones generalizadas

Las matrices de desplazamientos, velocidades y aceleraciones generalizadas $[\eta]$, $[\dot{\eta}]$ y $[\ddot{\eta}]$ respectivamente, corresponden a la solución del sistema de ecuaciones modales desacopladas $[M][\ddot{\eta}] + [C][\dot{\eta}] + [K][\eta] = [\bar{F}_{ext}]$ donde $[M]$, $[C]$ y $[K]$ son respectivamente las matrices modales de masa, amortiguamiento y rigidez.

Para el cálculo de las matrices $[\eta]$, $[\dot{\eta}]$ y $[\ddot{\eta}]$ se dispone de las subrutinas *ADACEL*, *ADWILS* y *ADNEWM* con las se resuelven las ecuaciones modales desacopladas utilizando uno de los tres métodos descritos en el Capítulo 4 (*aceleración lineal, excitación lineal y Newmark en forma iterativa*). Como resultado se crean las matrices reales *DCG()*, *VCG()* y *ACG()* en las que cada fila corresponde la solución de cada ecuación modal y cada columna corresponde a un instante de tiempo.

La subrutina utilizada depende del método seleccionado en la hoja "*TB_GEN*" a través de la variable *IDMC*. La Figura 6.39 muestra las líneas del código principal con las que se crean las matrices *DCG()*, *VCG()* y *ACG()*.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Desplazamientos, velocidades y aceleraciones generalizadas

'Desplazamientos, velocidades y aceleraciones generalizadas en cada instante de tiempo (DCG, VCG, ACG)
Select Case IDMC

    'método de la aceleración lineal
    Case "ACELERACIÓN LINEAL"
    ADACEL AMVI, FRN(), DTIM, TEI(), DCG(), VCG(), ACG()

    'método basado en la linealización de la excitación
    Case "EXCITACIÓN LINEAL"
    ADWILS AMVI, FRN(), DTIM, TEI(), DCG(), VCG(), ACG()

    'método de Newmark - forma iterativa
    Case "NEWMARK"
    ADNEWM AMVI, FRN(), DTIM, TEI(), DCG(), VCG(), ACG()
```

Figura 6.39. Desplazamientos, velocidades y aceleraciones generalizadas

```
    'método desconocido
    Case Else
    MsgBox "Error. Método de cálculo desconocido."
    Stop
    End

End Select

Debug.Print "Desplazamientos, velocidades y aceleraciones generalizadas", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.39. Desplazamientos, velocidades y aceleraciones generalizadas (*cont.*)

6.5.3. Desplazamientos, velocidades y aceleraciones de centros de masa

Las matrices de desplazamientos, velocidades y aceleraciones relativas a la base y en coordenadas globales de los centros de masa $[u_\phi]$, $[\dot{u}_\phi]$ y $[\ddot{u}_\phi]$ corresponden a los productos

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

$[\Phi][\eta]$, $[\Phi][\dot{\eta}]$y $[\Phi][\ddot{\eta}]$ respectivamente. En el código principal se utiliza la subrutina *MTMULT* de PEFiCA para realizar cada producto matricial. Los resultados se almacenan en las matrices reales *DCM()*, *VCM()* y *ACM()* en las que cada fila corresponde a un grado de libertad dinámico o posible desplazamiento (*o rotación*) de cada centro de masa, y cada columna corresponde a un instante de tiempo.

La Figura 6.40 muestra las líneas del código principal con las que se crean las matrices *DCM()*, *VCM()* y *ACM()*.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Desplazamientos, velocidades y aceleraciones relativas a la base de centros de masa

'desplazamientos relativos a la base (DCM)
MTMULT MOV(), DCG(), DCM()

'velocidades relativas a la base (VCM)
MTMULT MOV(), VCG(), VCM()

'aceleraciones relativas a la base (ACM)
MTMULT MOV(), ACG(), ACM()

Debug.Print "Desplazamientos, velocidades y aceleraciones relativas a la base de centros de masa", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.40. Desplazamientos, velocidades y aceleraciones de centros de masa

6.5.4. Verificación de la ecuación de movimiento

La verificación de la ecuación de movimiento se realiza calculando las acciones inerciales, de amortiguamiento y elásticas (*o equivalentes a las acciones internas*) que aparecen al lado izquierdo de la igualdad en la ecuación $[F_{I\phi}] + [F_{a\phi}] + [F_{s\phi}] = [F_{ext}]$, y comparándolas con las acciones externas equivalentes a la aceleración de la base $[F_{ext}]$. Esta comparación sirve de verificación de la solución obtenida y da una idea del error en el que se incurre al calcular los desplazamientos, las velocidades y las aceleraciones generalizadas utilizando métodos numéricos.

En el código se utiliza la subrutina *MTMULT* para calcular las matrices de acciones inerciales, de amortiguamiento y elásticas, que corresponden respectivamente a los productos $[F_{I\phi}] = [m][\ddot{u}_\phi]$, $[F_{a\phi}] = [c][\dot{u}_\phi]$ y $[F_{s\phi}] = [k_\phi][u_\phi]$. Los resultados se almacenan en las matrices reales *FIN()*, *FAM()* y *FEQ()*, cuyo tamaño coincide con el de la matriz $[u_\phi]$, y en las que cada fila corresponde a una acción (*fuerza o momento*) y cada columna a un instante de tiempo.

Con la subrutina *MTSUMA* se crea la matriz real *VF1()* sumando las matrices *FIN()*, *FAM()* y *FEQ()*. Los coeficientes de *VF1()* y los coeficientes de la matriz de acciones externas equivalentes a la aceleración de la base *FEX()* deben ser similares para que la solución obtenida de las ecuaciones modales desacopladas pueda considerarse correcta.

La Figura 6.41muestra las líneas del código principal con las que se crean las matrices *FIN()*, *FAM()*, *FEQ()* y *VF1()*.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Verificación del equilibrio dinámico

'acciones inerciales trasladadas a centros de masa (FIN)
MTMULT MDR(), ACM(), FIN()

'acciones de amortiguamiento trasladadas a centros de masa (FAM)
MTMULT AMR(), VCM(), FAM()

'acciones elásticas trasladadas a centros de masa (FEQ)
MTMULT KDP(), DCM(), FEQ()

'matriz de verificación del equilibrio dinámico (VF1)
MTSUMA FIN(), FAM(), TM1()
MTSUMA TM1(), FEQ(), VF1()

Debug.Print "Verificación del equilibrio dinámico", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.41. Verificación de la ecuación de movimiento

6.5.5. Balance de energía

La verificación de la ecuación de balance de energía se realiza calculando cada componente de energía involucrada en el sistema, específicamente la energía cinética $[E_c]$, la energía disipada por amortiguamiento natural $[E_D]$, la energía de deformación elástica $[E_{ss}]$ y la energía introducida por las acciones externas $[E_e]$, de forma que se pueda verificar que la igualdad $[E_c] + [E_D] + [E_{ss}] = [E_e]$ se cumple de forma aproximada, teniendo en cuenta que se están utilizando métodos numéricos para la solución de la ecuación de movimiento.

Como se supone que la estructura se mantiene en el rango elástico de deformaciones, la energía de deformación inelástica o disipada por histéresis $[E_{sy}]$ vale cero en cada instante de tiempo.

Para el cálculo de las energías cinética y de deformación elástica se utiliza la subrutina *ADENED*, la cual utiliza las expresiones directas descritas en el Capítulo 4, que son función de las matrices de masa y rigidez y, en este caso, de las matrices de velocidades y desplazamientos relativos a la base de los centros de masa. Como resultado se crean dos vectores fila reales denominados *ECI()* y *ESE()* en los que cada coeficiente corresponde a la energía cinética o de deformación elástica de la estructura en un instante de tiempo.

La energía disipada por amortiguamiento natural y la energía de entrada desde $t = 0$ hasta cualquier otro instante de tiempo no se pueden evaluar de forma directa. Estas se obtienen con la subrutina *ADENIN*, que calcula la energía al final de un intervalo de tiempo a partir de la energía al comienzo del intervalo, y aproxima el incremento de cada tipo de energía asumiendo que este equivale a una sumatoria de áreas de trapecios. Los resultados se almacenan en los vectores fila reales denominados *EAM()* y *EEX()*, los cuales tienen una forma similar a los vectores *ECI()* y *ESE()*.

Con la subrutina *MTSUMA* se crea el vector fila *VF2()* que calcula la energía de entrada sumando las energías cinética, amortiguada y de deformación. La comparación de los vec-

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

tores *EEX()* y *VF2()* sirve de verificación a la respuesta estructural calculada y da otra idea del error, similar a como lo hace *VF1()*.

Finalmente con la subrutina *ADBAEN* se almacenan en forma ordenada los vectores fila *ECI(), EAM(), ESE(), ESY(), EEX() y VF2()*. En cuanto a *ESY()*, que representa la energía disipada por histéresis, se crea un vector lleno de ceros con la subrutina *MTCONS* de PEFiCA.

La Figura 6.42 muestra las líneas del código principal con las que se crean las matrices *ECI(), EAM(), ESE(), EEX(), ESY(), VF2()* y *ENE()*.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Balance de energía

'energía cinética (ECI)
ADENED MDR(), VCM(), ECI()

'energía de amortiguamiento (EAM)
ADENIN DTIM, FAM(), VCM(), EAM()

'energía de deformación elástica (ESE)
ADENED KDP(), DCM(), ESE()

'energía de deformación inelástica (ESY=0)
MTCONS ESY(), 0, 1, NACE + 1

'energía de entrada (EEX)
ADENIN DTIM, FEX(), VCM(), EEX()

'vector fila de verificación de balance de energía (VF2)
MTSUMA ECI(), EAM(), TM1()
MTSUMA TM1(), ESE(), TM2()
MTSUMA TM2(), ESY(), VF2()

'matriz resumen de balance de energía (ENE)
ADBAEN ECI(), EAM(), ESE(), ESY(), EEX(), VF2(), ENE()

Debug.Print "Balance de energía", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.42. Balance de energía

6.5.6. Desplazamientos de los nudos

La matriz $[u]$ de desplazamientos relativos a la base y en coordenadas globales de los nudos de la estructura, se calcula a partir de los desplazamientos de los centros de masa $[u_\phi]$ y de la matriz de rigidez general $[k]$, siguiendo la formulación presentada en el Capítulo 4.

La matriz $[u]$ se crea con la subrutina *AEDESP* y se almacena en la matriz real *DNG()* en la que a cada fila le corresponde un grado de libertad estático y a cada columna le corresponde un instante de tiempo. En esta etapa también se crea una tabla resumen con los desplazamientos máximos de cada nudo y los instantes de tiempo en estos que se presentan. Se crea la matriz *DEM()* con la subrutina *AETRDE*.

La Figura 6.43 muestra las líneas del código principal con las que se crean las matrices *DNG()* y *DEM()*.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Desplazamientos de los nudos, relativos a la base y en coordenadas globales

'matriz completa de desplazamientos de los nudos en coordenadas globales (DNG)
AEDESP NGLD, NGLC, TDR(), DCM(), KGG(), KGA(), DBB(), DNG()

'tabla resumen de desplazamientos máximos probables y tiempos correspondientes ordenados por nudo (DEM)
AETRDE IDMA, MGL(), DNG(), TME(), DEM()

Debug.Print "Desplazamientos de los nudos, relativos a la base y en coordenadas globales", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.43. Desplazamientos relativos a la base y en coordenadas globales de los nudos

6.5.7. Derivas de centros de masa

Las derivas de los centros de masa en direcciones x e y se calculan a partir de los desplazamientos de los centros de masa con la subrutina *AEDECM*. Esta subrutina crea dos matrices reales denominadas *EXX()* y *EYY()* en la que cada fila corresponde a cada centro de masa y cada columna corresponde a un instante de tiempo. Las derivas calculadas aparecen como fracción de la altura de piso y en el orden en que se numeraron los diafragmas, de forma que se pueden comparar directamente con los límites especificados en las normas.

La Figura 6.44 muestra las líneas del código principal con las que se crean las matrices *EXX()* y *EYY()*.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Derivas de centros de masa

'matrices de derivas de centros de masa en direcciones X e Y
AEDECM NDIR, DCM(), DRX(), EXX(), EYY()

Debug.Print "Derivas de centros de masa", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.44. Derivas de centros de masa

6.5.8. Derivas en ejes de columnas

Para cada eje de columnas especificado en la hoja "*TB_GEN*" y siempre que se trate de un pórtico espacial, UNDIN calcula las derivas totales de cada eje en función de los desplazamientos y de las coordenadas de los nudos que están asociados a un mismo eje. Esto se realiza con la subrutina *AEDERI*, con la que se crea una matriz temporal denominada *DER()* en la que cada fila corresponde a un nudo o nivel del eje y cada columna corresponde a un instante de tiempo. Al igual que con las derivas de los centros de masa, las derivas calculadas aparecen como fracción de la altura de piso y ordenadas de acuerdo a los niveles de la estructura, comenzando por el más elevado y terminando por el más bajo.

Para cada eje también se crea, con la subrutina *ADVMAX*, una tabla temporal denominada *TM1()* con las derivas máximas del eje y los instantes de tiempo donde estas se pre-

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

sentan, la cual se escribe en la hoja de resultados "*TB_OUT*" con la subrutina *EDIMPR* de PEFiCA. Finalmente la matriz temporal *DER()* se exporta a un archivo de texto.

La Figura 6.45 muestra las líneas del código principal con las que se crean las matrices *DER()* y *TM1()* y con las que se exporta cada matriz *DER()* a un archivo de texto.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Derivas en ejes de columnas (si el pórtico es espacial)

If NDIM = 3 Then

    'para cada eje de columnas:
    For ICOL = 1 To NCOL

        'derivas en el eje de columnas actual como fracción de la altura de piso (DER)
        AEDERI ICOL, DNG(), XYZ(), MGL(), NEC(), NNE(), DER()

        'tabla de derivas máximas del eje actual (TM1)
        ADVMAX DER(), TME(), TM1()

        'escribir tabla de derivas máximas del eje actual (TM1)
        EDIMPR "TB_OUT", "Derivas máximas en eje de columna N° " & ICOL, FILA, COLM, TM1()

        'crear archivo de texto con la matriz DEV()
        IOEXMR DER(), TME(), "DER-" & ICOL

    Next ICOL

    Debug.Print "Derivas en ejes de columnas", (Timer - TINI)

End If
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.45. Derivas en ejes de columnas

6.5.9. Cortantes de piso

Las matrices de cortantes de piso en direcciones x e y global se calculan con la subrutina *AECORP* a partir de las acciones elásticas o equivalentes, y se almacenan en las matrices denominadas *VPX()* y *VPY()* respectivamente. En estas tablas cada fila corresponde a un piso y cada columna corresponde a un instante de tiempo.

Adicionalmente con la subrutina *AEDVPI* se crean dos tablas de datos que permiten graficar la variación en altura de los cortantes de piso en cada instante de tiempo. Estas tablas se almacenan en las matrices reales *DVX()* y *DVY()* en las que nuevamente, cada columna contiene los datos para graficar los cortantes de piso en un instante de tiempo.

Finalmente con la subrutina *AENVOL* se crean y se adicionan dos columnas al final de la tabla que contienen las envolventes de los diagramas de cortante de piso. La Figura 6.46 muestra las líneas del código principal con las que se crean las matrices *VPX(), VPY(), DVX()* y *DVY()*.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'-----------------------------------------------------------------------------------------------------------
'Cortantes de piso

'tablas de cortantes de piso en direcciones X e Y (VPX, VPY)
AECORP FEQ(), NDIM, VPX(), VPY()

'tablas resumen para graficar diagramas de cortantes de piso (DVX, DVY)
AEDVPI DRX(), VPX(), DVX()
AEDVPI DRX(), VPY(), DVY()

'crear envolventes y adjuntarlas al final de las tablas para graficar diagramas
AENVOL DVX()
AENVOL DVY()

Debug.Print "Cortantes de piso", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.46. Cortantes de piso

6.5.10. Acciones en los extremos de los elementos

La matriz de acciones en coordenadas locales en los extremos de cada elemento estructural se crea con la subrutina *AEFELE* a partir de los desplazamientos de los nudos y de la matriz de rigidez del elemento, siguiendo la formulación matemática descrita en el Capítulo 4. Como resultado se crea la matriz real *FEL()* de 12 filas, en la que cada fila corresponde a un grado de libertad del elemento y cada columna corresponde a un instante de tiempo. Esta matriz es temporal, se sobrescribe cuando se trabaja sobre otro elemento.

Para que la subrutina *AEFELE* funcione adecuadamente se requiere de la matriz *RFF()*, que almacena las acciones fijas producidas por las cargas distribuidas en la luz de los elementos. En este caso de análisis aún no se consideran cargas de este tipo, como si se hace en el método de análisis dinámico inelástico, y por lo tanto los coeficientes de esta matriz valen cero. Esta matriz se crea fuera del ciclo con la subrutina *MTCONS*.

Con la subrutina *AETRFE* se crea la tabla resumen *FEM()* que de forma similar a *DEM()*, almacena las acciones máximas en los extremos y los instantes de tiempo donde se presentan.

Dado que la matriz *FEL()* es temporal, esta se exporta con la subrutina *IOEXMR* a un archivo de texto para no perder la información del elemento. En la tabla exportada la matriz *FEL()* queda transpuesta y con una primera columna adicional que contiene los instantes de tiempo considerados en el análisis.

La Figura 6.47 muestra las líneas del código principal con las que se crean las matrices *FEL()* y *FEM()* y con las que se exporta cada matriz *FEL()* a un archivo de texto.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Acciones en los extremos de los elementos en coordenadas locales

'inicializar matrices FEM() y RFF() llenas de ceros
MTCONS FEM(), 0, NELE, 2 * NGLE
MTCONS RFF(), 0, NGLE, NELE

'para cada elemento de la estructura:
For IELE = 1 To NELE

    'matriz de acciones en los extremos del elemento actual (FEL)
    AEFELE IELE, INC(), ELE(), PAU(), RIG(), XYZ(), PAE(), DNG(), RFF(), FEL()

    'calcular y almacenar acciones maximas del elemento y tiempos correspondientes en matriz resumen (FEM)
    AETRFE IDMA, IELE, FEL(), TME(), FEM()

    'crear archivo de texto con matriz FEL()
    IOEXMR FEL(), TME(), "FEL-" & IELE

Next IELE

Debug.Print "Acciones en los extremos de los elementos en coordenadas locales", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.47. Acciones en coordenadas locales en los extremos de los elementos

6.5.11. Diagramas de acciones internas

En esta etapa se crean unas tablas de datos que permiten graficar los diagramas de acciones internas de cada eje de vigas y de columnas especificado en la hoja "*TB_GEN*".

La subrutina *AEDIVC* utiliza, entre otras matrices, la información de las tablas auxiliares de ejes de vigas y de columnas *VIG()* y *COL()* creadas previamente, para construir seis matrices reales denominadas *NXX(), VYY(), VZZ(), TXX(), MYY()* y *MZZ()* que contienen la información necesaria para graficar los seis diagramas de acciones internas del eje. En estas tablas cada fila contiene la variación de cada acción interna en una sección de uno de los elementos que pertenecen al eje, y cada columna, exceptuando la primera, corresponde a un instante de tiempo. La primera columna contiene las distancias entre el extremo inicial del primer elemento del eje hasta cada sección.

Con la subrutina *AEDIEN* se calculan las envolventes máximas y mínimas de cada acción interna y se adjuntan en dos columnas adicionales al final de las matrices *NXX(), VYY(), VZZ(), TXX(), MYY()* y *MZZ()*, similar a como se indicó previamente para las matrices de graficar los diagramas de cortantes de piso.

Como las seis matrices de acciones internas anteriores son temporales, es decir, se sobrescriben cuando se calcula otro eje diferente, estas se exportan a archivos de texto con la subrutina *IOEXDI*. En este caso la subrutina solo exporta las matrices en las que las acciones internas sean diferentes a cero en todos los instantes de tiempo, dado que en el caso de un pórtico plano, tres de las seis acciones internas valen cero, dependiendo de la orientación escogida de los ejes locales.

Similar al cálculo de las acciones en los extremos de los elementos, para que la subrutina *AEDIVC* funcione adecuadamente se requiere de las tablas *TWE()* y *WEL()* relacionadas

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

con las cargas distribuidas en la luz de los elementos. Estas matrices se inicializan llenas de ceros con las instrucciones *MTCONI* y *MTCONS* de PEFiCA.

La Figura 6.48 muestra las líneas del código principal con las que se crean y exportan a archivos de texto las matrices *NXX()*, *VYY()*, *VZZ()*, *TXX()*, *MYY()* y *MZZ()*.

```
'-----------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------
'Diagramas de acciones internas en vigas y columnas

'inicializar matrices TWE() y WEL() (no se utilizan pero las requiere la subrutina AEDIVC)
MTCONI TWE(), 0, 1, 1
MTCONS WEL(), 0, 1, 1

'diagramas de acciones internas de vigas
For IVIG = 1 To NVIG

    'tablas de acciones internas del eje de vigas actual (NXX, VYY, VZZ, TXX, MYY, MZZ)
    AEDIVC IDMA, IVIG, NACE + 1, VIG(), ELE(), CTS(), XYZ(), TWE(), WEL(), NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

    'crear envolventes y adjuntarlas al final de la tablas de acciones internas del eje actual
    AEDIEN NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

    'crear archivos de texto con tablas de acciones internas
    IOEXDI IVIG, NDIM, "V", NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

Next IVIG

'diagramas de acciones internas de columnas
For ICOL = 1 To NCOL

    'tablas de acciones internas del eje de columnas actual (NXX, VYY, VZZ, TXX, MYY, MZZ)
    AEDIVC IDMA, ICOL, NACE + 1, COL(), ELE(), CTS(), XYZ(), TWE(), WEL(), NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

    'crear envolventes y adjuntarlas al final de la tablas de acciones internas del eje actual
    AEDIEN NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

    'crear archivos de texto con tablas de acciones internas
    IOEXDI ICOL, NDIM, "C", NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

Next ICOL

Debug.Print "Diagramas de acciones internas en vigas y columnas", (Timer - TINI)
-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
```

Figura 6.48. Diagramas de acciones internas

6.6. Parte V: Análisis modal espectral

En esta etapa UNDIN calcula la respuesta máxima probable de una estructura en el rango elástico y con amortiguamiento clásico, sometida a una o varias componentes de aceleración en la base. La respuesta estructural se obtiene a partir de un espectro de pseudo – aceleraciones.

Inicialmente se describe como se calculan los desplazamientos generalizados máximos a partir del espectro de pseudo – aceleraciones y, a partir de estas, como se estiman los desplazamientos (*relativos a la base*) máximos probables de los centros de masa. A partir de los desplazamientos máximos por modo de los centros de masa se estima el valor máximo probable de los demás parámetros que representan la respuesta de la estructura en cada ins-

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

tante de tiempo (*desplazamientos de nudos, derivas, acciones equivalentes, acciones internas, etc.*) cuando actúan de forma simultánea todas las componentes de aceleración de la base consideradas en el análisis.

Se describe la forma en que en cada etapa se crean matrices resumen para cada cantidad de interés, en las que se almacena la respuesta máxima por modo, la respuesta máxima probable por cada componente de aceleración de la base y finalmente, la respuesta máxima probable debida a la acción simultánea de todas las componentes de aceleración de la base.

6.6.1. Desplazamientos generalizados máximos

La matriz de desplazamientos generalizados máximos $[\eta_o]$ es función de las frecuencias de vibración natural, los factores de participación y de la pseudo – aceleración espectral asociada a cada modo. A continuación se describe como se obtiene esta matriz.

Con la subrutina *ADESPT* se crea una matriz de tipo real denominada *SAT()* que en la primera columna almacena los periodos de vibración de la estructura, y en la segunda columna almacena el valor de S_a correspondiente a cada periodo de vibración. Los valores de S_a son interpolados del espectro de pseudo – aceleraciones almacenado en la matriz *ESP()*.

Con la subrutina *ADCGEN* se crea la matriz de desplazamientos generalizados máximos por modo y por componente de aceleración de la base $[\eta_o]$, siguiendo la formulación matemática descrita en el Capítulo 4 para calcular cada coeficiente $(\eta_{ij})_o$. Como resultado se crea la matriz *GEN()* en la que cada fila corresponde a un modo y cada columna corresponde a cada componente de aceleración de la base considerada, dependiendo del tipo de pórtico (*plano o espacial*).

La Figura 6.49 muestra las líneas del código principal con las que se crean las matrices *SAT()* y *GEN()*.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Desplazamientos generalizados máximos

'calcular pseudo-aceleraciones espectrales correspondientes a cada periodo de vibración (SAT)
ADESPT PER(), ESP(), SAT()

'calcular desplazamientos generalizadas máximos (GEN)
ADCGEN FESX, FESY, SAT(), ALF(), FRN(), GEN()

Debug.Print "Desplazamientos generalizadas máximos", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.49. Desplazamientos generalizados máximos

6.6.2. Matriz de coeficientes de correlación

En esta etapa se construye la matriz de coeficientes de correlación $[\rho]$, requerida cuando se utiliza el método de la *Combinación Cuadrática Completa CCC* para combinar la respuesta

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

máxima por modo de la estructura, producida cuando actúa cada componente de la aceleración de la base de forma individual.

Para ello se utiliza la subrutina *ADMCOR* la cual construye la matriz $[\rho]$ siguiendo la formulación matemática mostrada en el Capítulo 4. El resultado se almacena en la matriz real *COR()*, la cual es cuadrada y su tamaño coincide con el número de modos de vibración natural de la estructura.

En caso que no se utilice el método *CCC* la subrutina *ADMCOR* crea una matriz cuadrada del mismo tamaño pero llena de ceros. En este caso esta matriz no se utiliza, aunque se requiere en otras subrutinas que realizan la combinación modal de la respuesta.

La Figura 6.50 muestra la línea del código principal con la que se crea la matriz *COR()*.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Matriz de coeficientes de correlación utilizada en el método CCC (COR)
ADMCOR IDMA, IDCM, AMVI, FRN(), COR()
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.50. Matriz de coeficientes de correlación de método *CCC*

6.6.3. Desplazamientos máximos probables de centros de masa

El vector de desplazamientos máximos probables (*relativos a la base*) de los centros de masa $\{(u_\phi)_o\}$ se calcula a partir de los vectores $\{(u_\phi)_o^x\}$ y $\{(u_\phi)_o^y\}$ como se indicó en la Tabla 4.2. A su vez los vectores $\{(u_\phi)_o^x\}$ y $\{(u_\phi)_o^y\}$ son función de las matrices $[(u_\phi)^x]$ y $[(u_\phi)^y]$ como se indicó en la misma tabla. A continuación se describe como se calcula cada uno de los anteriores términos en el programa.

Con la subrutina *ADDESP* se crean $[(u_\phi)^x]$ y $[(u_\phi)^y]$ las cuales se almacenan en las matrices reales *DCX()* y *DCY()* respectivamente. En estas matrices cada columna corresponde a un modo de vibración y cada fila corresponde a un posible desplazamiento o rotación de cada centro de masa.

Con la subrutina *ADTRAE* se crea la matriz resumen *DCR()* que almacena las matrices $[(u_\phi)^x]$ y $[(u_\phi)^y]$ y los vectores $\{(u_\phi)_o^x\}$, $\{(u_\phi)_o^y\}$ y $\{(u_\phi)_o\}$, estos tres últimos calculados utilizando los métodos de combinación modal y direccional con la misma subrutina.

Finalmente con la subrutina *ORDIRE* se crean las matrices *XCR()*, *YCR()* y *ZCR()* que almacenan de forma separada la información de la matriz *DCR()*,con la diferencia que las dos primeras almacenan los desplazamientos en direcciones x e y la tercera almacena las rotaciones en dirección z. En cada una de las anteriores tablas cada fila corresponde a un nivel de la estructura.

La Figura 6.51 muestra las líneas del código principal con las que se crean las matrices *DCX()*, *DCY()*, *DCR()*, *XCR()*, *YCR()* y *ZCR()*.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Desplazamientos máximos probables de centros de masa

'desplazamientos máximos por modo al aplicar los movimientos sísmicos en direcciones X e Y individualmente
'(DCX, DCY)
ADDESP GEN(), MOV(), DCX(), DCY()

'tabla resumen de desplazamientos máximos probables de centros de masa (DCR)
ADTRAE IDCM, IDCD, COR(), DCX(), DCY(), DCR()

'tablas resumen de desplazamientos de CM's separados en direcciones Ux,Uy,Rz (XCR, YCR, ZCR)
ORDIRE NDIM, DCR(), DRX(), XCR(), YCR(), ZCR()

Debug.Print "Desplazamientos máximos probables de centros de masa", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.51. Desplazamientos máximos probables de centros de masa

6.6.4. Desplazamientos máximos probables de los nudos

El vector $\{(u)_o\}$ de desplazamientos máximos probables (*relativos a la base y en coordenadas globales*) de los nudos, se calcula a partir de los vectores $\{(u)_o^x\}$ y $\{(u)_o^y\}$, que a su vez son función de las matrices $[(u)^x]$ y $[(u)^y]$ como se indicó en la Tabla 4.2. A continuación se describe como se calcula cada término en UNDIN.

Con la subrutina *AEDESP* se crean las matrices $[(u)^x]$ y $[(u)^y]$ las cuales se almacenan en las matrices reales *DNX()* y *DNY()* respectivamente. En estas matrices cada columna corresponde a un modo de vibración y cada fila corresponde a un posible desplazamiento o rotación de cada nudo. Esta subrutina se utilizó en el método de análisis modal cronológico para crear la matriz de desplazamientos.

Con la subrutina *ADTRAE* se crea la matriz resumen denominada *DNR()* que almacena las matrices $[(u)^x]$ y $[(u)^y]$ y los vectores $\{(u)_o^x\}$, $\{(u)_o^y\}$ y $\{(u)_o\}$, similar a como se construye la matriz *DCR()*. Los tres últimos vectores también se crean con *ADTRAE*.

Finalmente con la subrutina *AETRDE* se crea la tabla resumen *DEM()* con los desplazamientos máximos probables $\{(u)_o\}$ organizados de forma que en cada fila se almacenan los seis desplazamientos de un mismo nudo en el formato general $(u_x, u_y, u_z, \theta_x, \theta_y, \theta_z)$.

La Figura 6.52 muestra las líneas del código principal con las que se crean las matrices *DNX(), DNY(), DNR()* y *DEM()*.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Desplazamientos máximos probables de los nudos en coordenadas globales

'desplazamientos de los nudos por modo al aplicar los movimientos sísmicos en X e Y individualmente (DNX, DNY)
AEDESP NGLD, NGLC, TDR(), DCX(), KGG(), KGA(), DBB(), DNX()
AEDESP NGLD, NGLC, TDR(), DCY(), KGG(), KGA(), DBB(), DNY()

'tabla resumen de desplazamientos máximos probables de nudos (DNR)
ADTRAE IDCM, IDCD, COR(), DNX(), DNY(), DNR()
```

Figura 6.52. Desplazamientos máximos probables en coordenadas globales de los nudos

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'tabla resumen de desplazamientos máximos probables ordenados por nudo (DEM)
MTCONS TME(), 0, 1, 1   'no se utiliza pero hay que definirlo
AETRDE IDMA, MGL(), DNR(), TME(), DEM()

Debug.Print "Desplazamientos máximos probables de los nudos en coordenadas globales", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.52. Desplazamientos máximos probables en coordenadas globales de los nudos (*cont.*)

6.6.5. Derivas máximas probables de centros de masa

El cálculo de las derivas máximas probables en direcciones x e y, $\{(\Delta_x)_o\}$ y $\{(\Delta_y)_o\}$ respectivamente, se describe a continuación.

Se utiliza la subrutina *AEDECM* para calcular las derivas máximas por modo. Como resultado se crean cuatro matrices reales denominadas *EXX(), EYX(), EXY()* y *EYY()*, las dos primeras almacenan las derivas producidas solamente por la aceleración de la base en dirección x, y las dos últimas almacenan las derivas producidas por la aceleración de la base en dirección y.

Posteriormente con la subrutina *ADTRAE* se crean dos tablas resumen denominadas *EXR()* y *EYR()*. La primera almacena las matrices $[(\Delta_x)^x]$ y $[(\Delta_x)^y]$ y los vectores $\{(\Delta_x)_o^x\}$, $\{(\Delta_x)_o^y\}$ y $\{(\Delta_x)_o\}$ y la segunda almacena las matrices $[(\Delta_y)^x]$ y $[(\Delta_y)^y]$ y los vectores $\{(\Delta_y)_o^x\}$, $\{(\Delta_y)_o^y\}$ y $\{(\Delta_y)_o\}$, resumiendo el cálculo de las derivas de los centros de masa en direcciones x e y respectivamente. Los seis vectores previamente mencionados también se crean con la misma subrutina.

La Figura 6.53 muestra las líneas del código principal con las que se crean las matrices *EXX(), EYX(), EXY(), EYY(), EXR()* y *EYR()*.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Derivas máximas probables de centros de masa

'derivas máximas de centros de masa por modo al aplicar los movimientos sísmicos en X e Y individualmente
'(EX1, EX2, EY1, EY2)
AEDECM NDIR, DCX(), DRX(), EXX(), EYX()
AEDECM NDIR, DCY(), DRX(), EXY(), EYY()

'tablas resumen de derivas máximas probables de centros de masa (EXR, EYR)
ADTRAE IDCM, IDCD, COR(), EXX(), EXY(), EXR()
ADTRAE IDCM, IDCD, COR(), EYX(), EYY(), EYR()

Debug.Print "Derivas máximas probables de centros de masa", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.53. Derivas máximas probables de centros de masa

6.6.6. Derivas máximas probables en ejes de columnas

A continuación se describe el cálculo del vector $\{(\Delta_i)_o\}$ para cada eje de columnas de un pórtico espacial.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

Con la subrutina *AEDERI* se calculan las matrices $[(\Delta_i)^x]$ y $[(\Delta_i)^y]$, y se almacenan en las matrices *DEX()* y *DEY()*. Posteriormente con la subrutina *ADTRAE* se calculan los vectores $\{(\Delta_i)_o^x\}$, $\{(\Delta_i)_o^y\}$ y $\{(\Delta_i)_o\}$ y se crea una matriz resumen denominada *DER()*, para guardar en una sola tabla las dos matrices y los tres vectores previamente creados.

Finalmente dado que las matrices *EX()*, *DEY()* y *DER()* son temporales, con las subrutinas *EDIMPR* y *IOEXPR* de PEFiCA se escribe y se exporta la matriz *DER()* para no perder los resultados.

La Figura 6.54 muestra las líneas del código principal con las que se crean las matrices *DEX()*, *DEY()* y *DER()* y con las que se exporta la matriz *DER()*.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Derivas máximas probables en ejes de columnas

If NDIM = 3 Then
    'para cada eje de columna:
    For ICOL = 1 To NCOL

        'derivas máximas por modo al aplicar los movimientos sísmicos en X e Y individualmente (DEX, DEY)
        AEDERI ICOL, DNX(), XYZ(), MGL(), NEC(), NNE(), DEX()
        AEDERI ICOL, DNY(), XYZ(), MGL(), NEC(), NNE(), DEY()

        'tabla resumen de derivas máximas probables del eje actual (DER)
        ADTRAE IDCM, IDCD, COR(), DEX(), DEY(), DER()

        'escribir y crear archivo de texto con matriz DER()
        EDIMPR "TB_OUT", "Tabla resumen de derivas máximas en eje de columna " & ICOL, FILA, COLM, DER()
        IOEXPR DER(), "DER-" & ICOL

    Next ICOL
    Debug.Print "Derivas máximas por ejes de columnas", (Timer - TINI)

End If
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.54. Derivas máximas probables en ejes de columnas

6.6.7. Acciones equivalentes máximas probables trasladadas a los centros de masa

El vector de acciones equivalentes máximas probables $\{(F_{s\phi})_o\}$ se obtiene a partir de los vectores $\{(F_{s\phi})_o^x\}$ y $\{(F_{s\phi})_o^x\}$, los cuales a su vez se obtienen de las matrices $[(F_{s\phi})^x]$ y $[(F_{s\phi})^y]$ como se indicó en el Capítulo 4. A continuación se describe la forma en que se calcula el vector $\{(F_{s\phi})_o\}$.

Con la subrutina *MTMULT* se crean dos matrices reales denominadas *FCX()* y *FCY()* que corresponden a las matrices $[(F_{s\phi})^x]$ y $[(F_{s\phi})^y]$ respectivamente. En estas matrices cada columna corresponde a un modo y cada fila corresponde a una fuerza o momento equivalente.

Con la subrutina *ADTRAE* se calculan los vectores $\{(F_{s\phi})_o^x\}$, $\{(F_{s\phi})_o^x\}$ y $\{(F_{s\phi})_o\}$ y se crea la matriz resumen denominada *FCR()*, que almacena de forma ordenada las dos matrices anteriores y los tres vectores que se acaban de calcular.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

Con la subrutina *ORDIRE* se crean las matrices *FXR()*, *FYR()* y *MZR()* en las que se almacena la misma información contenida en la matriz *FCR()*, pero separada por componentes. La primera matriz almacena las fuerzas en dirección x, la segunda almacena las fuerzas equivalentes en dirección y y la tercera los momentos equivalentes en dirección z.

La Figura 6.55 muestra las líneas del código principal con las que se crean las matrices *FCX()*, *FCX()*, *FCR()*, *FXR()*, *FYR()* y *MZR()*.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Acciones equivalentes máximas probables trasladadas a centros de masa

'acciones equivalentes máximas por modo al aplicar los movimientos sísmicos en X e Y individualmente (FCX, FCY)
MTMULT KDP(), DCX(), FCX()
MTMULT KDP(), DCY(), FCY()

'tabla resumen de acciones ewuivalentes máximas probables (FCR)
ADTRAE IDCM, IDCD, COR(), FCX(), FCY(), FCR()

'tablas resumen de acciones equivalentes en direcciones Ux,Uy,Rz (FXR, FYR, MZR)
ORDIRE NDIM, FCR(), DRX(), FXR(), FYR(), MZR()

Debug.Print "Acciones equivalentes máximas probables trasladadas a centros de masa", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.55. Acciones equivalentes máximas probables trasladadas a centros de masa

6.6.8. Cortantes de piso máximos probables

Con la subrutina *AECORP* se crean las matrices *PXX()*, *PYX()*, *PXY()* y *PYY()* que almacenan los cortantes de piso máximos por modo. Las dos primeras matrices almacenan los cortantes debidos a la aceleración de la base en dirección x y las dos matrices restantes almacenan los cortantes debidos a la aceleración de la base en dirección y. Estas cuatro tablas corresponden a las matrices $[(V_x)^x]$, $[(V_y)^x]$, $[(V_x)^y]$ y $[(V_y)^y]$ respectivamente.

Con *ADTRAE* se crean dos matrices resumen denominadas *VXR()* y *VYR()*. La primera almacena las matrices $[(V_x)^x]$ y $[(V_x)^y]$ y los vectores $\{(V_x)_o^x\}$, $\{(V_x)_o^y\}$ y $\{(V_x)_o\}$ que se calculan internamente con la misma subrutina. De forma similar *VYR()* almacena en forma secuencial las matrices $[(V_y)^x]$ y $[(V_y)^y]$ y los vectores $\{(V_y)_o^x\}$, $\{(V_y)_o^y\}$ y $\{(V_y)_o\}$.

Se utiliza nuevamente la subrutina *AEDVPI* para construir las matrices *DVX() y DVY()* como las descritas en la parte de análisis modal cronológico. En estas matrices ahora cada columna corresponde a un modo o a un valor máximo probable, no a un instante de tiempo.

La Figura 6.56 muestra las líneas del código principal con las que se crean las matrices *PXX()*, *PYX()*, *PXY()*, *PYY()*, *VXR()*, *VYR()*, *DVX()* y *DVY()*.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Cortantes de piso máximos probables

'cortantes de piso máximos por modo al aplicar los movimientos sísmicos en X e Y individualmente
AECORP FCX(), NDIM, PXX(), PYX()
AECORP FCY(), NDIM, PXY(), PYY()
```

Figura 6.56. Cortantes de piso máximos probables

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'tablas resumen de cortantes de piso máximos probables (VXR, VYR)
ADTRAE IDCM, IDCD, COR(), PXX(), PXY(), VXR()
ADTRAE IDCM, IDCD, COR(), PYX(), PYY(), VYR()

'tablas resumen para graficar diagramas de cortantes de piso (DVX, DVY)
AEDVPI DRX(), VXR(), DVX()
AEDVPI DRX(), VYR(), DVY()

Debug.Print "Cortantes de piso máximos probables", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.56. Cortantes de piso máximos probables (*cont.*)

6.6.9. Acciones máximas en los extremos de los elementos

Las acciones máximas probables en coordenadas locales en los extremos, se calculan para cada elemento de la estructura de forma similar a como se escribió en la parte de análisis modal cronológico. Se utiliza dos veces la subrutina *AEFELE* para construir las matrices *FBX()* y *FBY()* que coinciden con las matrices $[(\bar{f}_e)^x]$ y $[(\bar{f}_e)^y]$ respectivamente.

La subrutina *ADTRAE* crea la matriz temporal *FEL()*, que almacena los vectores $\{(\bar{f}_e)_o^x\}$, $\{(\bar{f}_e)_o^y\}$ y $\{(\bar{f}_e)_o\}$ en forma secuencial junto con las matrices $[(\bar{f}_e)^x]$ y $[(\bar{f}_e)^y]$.

Con la subrutina *AETRFE* se crea una tabla resumen denominada *FEM()*, que almacena las acciones máximas probables en los extremos de cada elemento, tomadas del vector $\{(\bar{f}_e)_o\}$ y organizadas de forma que en cada fila se almacenen las 12 acciones de un mismo elemento. Nuevamente dado que la matriz *FEL()* es temporal, esta se exporta a un archivo de texto utilizando la subrutina *IOEXPR*.

La Figura 6.57 muestra las líneas del código principal con las que se crean las matrices *FBX()*, *FBY()*, *FEL()* y *FEM()* y con la que se exporta la matriz *FEM()*.

```
'-----------------------------------------------------------------------------------------------------------
'Acciones máximas probables en coordenadas locales en los extremos de los elementos

'inicializar matrices FEM() y RFF() llenas de ceros
MTCONS FEM(), 0, NELE, NGLE
MTCONS RFF(), 0, NGLE, NELE

'para cada elemento estructural:
For IELE = 1 To NELE

    'acciones máximas en los extremos del elemento actual por modo (FBX, FBY)
    AEFELE IELE, INC(), ELE(), PAU(), RIG(), XYZ(), PAE(), DNX(), RFF(), FBX()
    AEFELE IELE, INC(), ELE(), PAU(), RIG(), XYZ(), PAE(), DNY(), RFF(), FBY()

    'tabla resumen de acciones máximas probables en los extremos del elemento actual (FEL)
    ADTRAE IDCM, IDCD, COR(), FBX(), FBY(), FEL()

    'crear archivo de texto con tabla resumen FEL()
    IOEXPR FEL(), "FEL-" & IELE

    'tabla resumen de acciones maximas en extremos de elementos (FEM)
    AETRFE IDMA, IELE, FEL(), TME(), FEM()

Next IELE

Debug.Print "Acciones máximas probables en los extremos de los elementos", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.57. Acciones máximas probables en los extremos de los elementos

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

6.6.10. Diagramas de acciones internas

En esta etapa se crean las tablas de datos que permiten graficar los diagramas de acciones internas de cada eje de vigas y de columnas, de forma similar a como se describió en la parte de análisis modal cronológico.

Se utiliza la subrutina *AEDIVC* para construir seis matrices reales denominadas *NXX(), VYY(), VZZ(), TXX(), MYY()* y *MZZ()* que contienen la información necesaria para graficar los seis diagramas de acciones internas del eje. En estas matrices cada columna, exceptuando la primera, corresponde a un modo de vibración natural. En cada matriz se han calculado de forma simultánea las acciones internas debidas a las dos componentes de aceleración de la base actuando de forma individual, por lo que el número de columnas es dos veces el número de modos.

Con la subrutina *ADTRDI* se adicionan cuatro columnas a las matrices *NXX(), VYY(), VZZ(), TXX(), MYY()* y *MZZ()*. Las tres primeras columnas contienen los valores máximos probables cuando el sismo actúa individualmente en cada dirección, la tercera contiene las acciones máximas probables por el sismo actuando simultáneamente en cada dirección considerada, siendo esta la envolvente de acciones internas positiva, y la cuarta contiene el negativo de la tercera columna, que correspondería a la envolvente máxima negativa.

Como las seis matrices de acciones internas son temporales, es decir, se sobrescriben cuando se pasa a otro eje, estas se exportan a archivos de texto con la subrutina *IOEXDI*. Se tiene en cuenta nuevamente el tipo de pórtico para escoger que matrices exportar.

La Figura 6.58 muestra las líneas del código principal con las que se crean y exportan a archivos de texto las matrices *NXX(), VYY(), VZZ(), TXX(), MYY()* y *MZZ()*.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Diagramas de acciones internas en vigas y columnas

'inicializar matrices TWE() y WEL() (no se utilizan pero las requiere la subrutina AEDIVC)
MTCONI TWE(), 0, 1, 1
MTCONS WEL(), 0, 1, 1

'diagramas de acciones internas de vigas
For IVIG = 1 To NVIG

    'tablas de acciones internas máximas por modo, al aplicar los movimientos sísmicos en direcciones X e Y
    'individualmente (NXX, VYY, VZZ, TXX, MYY, MZZ)
    AEDIVC IDMA, IVIG, NMOD, VIG(), ELE(), CTS(), XYZ(), TWE(), WEL(), NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

    'crear tablas resumen de acciones internas máximas probables en el eje actual
    ADTRDI IDCM, IDCD, NMOD, COR(), NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

    'crear archivos de texto con tablas de acciones internas
    IOEXDI IVIG, NDIM, "V", NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

Next IVIG

'diagramas de acciones internas de columnas
For ICOL = 1 To NCOL

    'tablas de acciones internas máximas por modo, al aplicar los movimientos sísmicos en direcciones X e Y
    'individualmente (NXX, VYY, VZZ, TXX, MYY, MZZ)
    AEDIVC IDMA, ICOL, NMOD, COL(), ELE(), CTS(), XYZ(), TWE(), WEL(), NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()
```

Figura 6.58. Diagramas de acciones internas

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
    'crear tablas resumen de acciones internas máximas probables en el eje actual
    ADTRDI IDCM, IDCD, NMOD, COR(), NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

    'crear archivos de texto con tablas de acciones internas
    IOEXDI ICOL, NDIM, "C", NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

Next ICOL

Debug.Print "Diagramas de acciones internas en vigas y columnas", (Timer - TINI)
'''''''
```

Con la subrutina *AEFFES* se crea el vector de acciones fijas en coordenadas globales producidas por las cargas iniciales, el cual se almacena en el vector columna *FFE()*. También se crea la matriz *RFF()* que almacena en cada columna el vector de acciones fijas en coordenadas locales de cada elemento de la estructura. El vector columna *FNE()*, que almacena las acciones iniciales externas en coordenadas globales aplicadas en los nudos y que se requiere en esta subrutina, se crea con la subrutina *MTCONS* lleno de ceros, dado que no se consideran acciones iniciales de este tipo.

Con la subrutina *AEDESI* se calculan los desplazamientos en coordenadas globales de los nudos y de los centros de masa, y las acciones equivalentes producidas por las cargas iniciales. Los resultados se almacenan en los vectores columna denominados *DNG()*, *DCM()* y *FEQ()* respectivamente.

Con la subrutina *ORGLFU* se crea la tabla *RDI()* de desplazamientos iniciales de nudos, en la que en cada fila se almacenan los seis desplazamientos en $t = 0$ de cada nudo.

A continuación se calculan las acciones en coordenadas locales en los extremos de cada elemento producidas por las cargas iniciales. Para ello se utiliza la subrutina *AEFELE*, utilizada ya en los métodos descritos anteriormente, y que entrega como resultado el vector columna *FEL()*.

El vector *FEL()* es temporal, por lo tanto con la subrutina *MTREMV* se almacena en la tabla resumen *RFI()*, en la que cada columna corresponde a un elemento. La Figura 6.59 muestra las líneas del código principal con las que se crean las matrices descritas.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Análisis elástico de la estructura para cargas iniciales

'vector de acciones aplicadas en los nudos (FNE=0)
MTCONS FNE(), 0, NGLT, 1

'vector y tabla resumen de acciones fijas de la estructura en coordenadas locales (FFE, RFF)
AEFFES NELE, NGLT, INC(), ELE(), XYZ(), PAE(), TWE(), WEL(), PAU(), RIG(), FFE(), RFF()

'desplazamientos de los nudos y de los centros de masa debidos a cargas iniciales y acciones equivalentes
'iniciales (DCM, DNG, FEQ)
AEDESI NGLI, NGLO, KDP(), FNE(), FFE(), KAG(), KGA(), KGG(), TDR(), TDT(), DBB(), DCM(), DNG(), FEQ()

'crear tabla de desplazamientos iniciales organizada por nudo (RDI)
ORGLFU RDI(), DNG(), MGL()

'inicializar tabla resumen de acciones iniciales en los extremos de los elementos (RFI)
MTCONS RFI(), 0, NGLE, NELE

'acciones iniciales en los extremos de los elementos en coordenadas locales (RFI)
For IELE = 1 To NELE

    'calcular acciones en los extremos del elemento actual (FEL)
    AEFELE IELE, INC(), ELE(), PAU(), RIG(), XYZ(), PAE(), DNG(), RFF(), FEL()

    'almacenar acciones iniciales del elemento en tabla resumen (RFI)
    MTREMV RFI(), FEL(), "COLUMNA", IELE

Next IELE

Debug.Print "Análisis de la estructura para cargas iniciales", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.59. Análisis elástico de la estructura para cargas iniciales

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

6.7.2. Matriz de acciones externas equivalentes a la aceleración de la base

En esta parte se repiten las líneas que se utilizaron en el método de análisis modal cronológico para crear la matriz $[F_{ext}]$, con las que se crean las matrices *MUO(), TME(), ACE() y FEX()* como se muestra en la Figura 6.60.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Matriz de acciones externas equivalentes a la aceleración de la base

'tabla de acelerogramas, vector de instantes de tiempo y matriz de aceleración de la base (ACE, TME, MUO)
ADUGXY NACE, DTIM, FESX, FESY, NGDR, UGX(), UGY(), ACE(), TME(), MUO()

'matriz de acciones externas equivalentes a la aceleración de la base (FEX)
MTPORE MDR(), -1, TM1()
MTMULT TM1(), GAM(), TM2()
MTMULT TM2(), MUO(), FEX()

Debug.Print "Matriz de acciones externas equivalentes a la aceleración de la base", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.60. Acciones externas equivalentes a la aceleración de la base

6.7.3. Inicialización de matrices de resultados

A continuación se describe la forma en que se crean e inicializan las matrices de resultados que almacenarán la respuesta principal de la estructura en cada instante de tiempo.

Se utiliza la instrucción *Redim Preserve* para inicializar las matrices $[u_\phi]$, $[u]$ y $[F_{s\phi}]$ de desplazamientos de centros de masa, desplazamientos de nudos y acciones equivalentes trasladadas a centros de masa respectivamente, aumentando el tamaño de los vectores columna *DCM(), DNG()* y *FEQ()* obtenidos del análisis para cargas iniciales. Los anteriores vectores se convierten en matrices con tantas columnas como instantes de tiempo se consideren en el análisis, almacenando en la primera columna de cada matriz los resultados del análisis para cargas iniciales.

Con la subrutina *MTCONS* se inicializan las matrices *VCM(), ACM() y FAM()* llenas de ceros, en las que más adelante se almacenan las velocidades y las aceleraciones de los centros de masa y las acciones resultantes de amortiguamiento en cada instante de tiempo, matrices $[\dot{u}_\phi]$, $[\ddot{u}_\phi]$ y $[F_{a\phi}]$ respectivamente.

Con la subrutina *MTCONS* también se inicializan las matrices *ERO(), DRO(), FRO(), KRO(), VF3() y FBL()*, llenas de ceros y con tantas columnas como instantes de tiempo se consideren en el análisis. En las 5 primeras matrices posteriormente se almacenarán, para cada resorte rotacional, el parámetro de deformación elástica, la rotación, el momento, la rigidez tangente al comienzo del intervalo y el momento en el extremo del elemento correspondiente a cada resorte, este último para verificar las acciones en las rótulas plásticas. Cada fila de estas matrices corresponde a una rótula plástica asignada a la estructura. La matriz *FBL()* almacenará las acciones en coordenadas locales en los extremos de todos los elementos de la estructura, su número de filas es igual a 12 veces el número de elementos.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

Con la subrutina *MTCONS* se inicializan las matrices *ITE()* y *ERR(),* en las que más adelante se almacena la cantidad de iteraciones y el error final alcanzados en los ciclos iterativos principal y secundario, una vez los resultados convergen. Cada matriz tiene dos filas y a cada columna le corresponde a un instante de tiempo. La primera fila de cada matriz almacenará los resultados del ciclo principal o externo, mientras que la segunda fila almacenará los resultados del ciclo secundario.

Con la subrutina *AEINIR* se llena la primera columna de las matrices *ERO(), DRO(), FRO()* y *KRO()* con los momentos, las rotaciones, los parámetros de rotación elástica y la rigidez tangente de las rótulas plásticas (*resortes rotacionales*), producidas por las cargas iniciales. Los momentos se toman de las acciones en los extremos de los elementos donde hay rótulas asociadas, la rigidez tangente se supone igual a la rigidez inicial, y a partir de estos se calculan las rotaciones y los parámetros de rotación elástica de las rótulas, utilizando la formulación del modelo de Bouc – Wen.

Con la subrutina *AEINIF* se llena la primera columna de la matriz *FBL(),* con las acciones en coordenadas locales en los extremos de los elementos producidas por las cargas iniciales, almacenadas en la matriz *RFI().*

Con la subrutina *ADACIN* se calculan y se almacenan, en la primera columna, las aceleraciones iniciales de los centros de masa, despejándolas de la ecuación de movimiento planteada en $t = 0$. Para ello se conocen las acciones externas, las acciones equivalentes y como se suponen las velocidades iniciales $\{\dot{u}_\phi(0)\} = \{0\}$, las acciones iniciales de amortiguamiento valen cero.

Finalmente con la subrutina *ADCONS* se crea un vector que almacena las constantes c_1 a c_9 y los coeficientes γ, β y Δt de la forma del método de Newmark (*aceleración promedio o lineal*) que se utilizará posteriormente para solucionar las ecuaciones de movimiento en forma incremental. La Figura 6.61 muestra las líneas del código principal con las que se realizan las operaciones anteriormente descritas.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Inicialización de matrices de resultados

'redimensionar matrices de desplazamientos de centros de masa, de desplazamientos de nudos y de acciones
'elásticas, conservando las condiciones iniciales (primera columna) (DCM, DNG, FEQ)
ReDim Preserve DCM(NDIN, NACE + 1)
ReDim Preserve DNG(NGLT, NACE + 1)
ReDim Preserve FEQ(NDIN, NACE + 1)

'inicializar matrices de velocidades y aceleraciones de centros de masa (VCM, ACM)
MTCONS VCM(), 0, NDIN, (NACE + 1)
MTCONS ACM(), 0, NDIN, (NACE + 1)

'inicializar matriz de acciones de amortiguamiento (FAM)
MTCONS FAM(), 0, NDIN, (NACE + 1)
```

Figura 6.61. Inicialización de matrices de resultados

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'inicializar matrices de resultados de rótulas plásticas:
MTCONS ERO(), 0, ROTE, NACE + 1                 'parámetros de deformación elástica (ERO)
MTCONS DRO(), 0, ROTE, NACE + 1                 'desplazamientos (o rotaciones) (DRO)
MTCONS FRO(), 0, ROTE, NACE + 1                 'fuerzas (o momentos) (FRO)
MTCONS KRO(), 0, ROTE, NACE + 1                 'rigidez tangente al comienzo de cada intervalo (KRO)
MTCONS FBL(), 0, (NELE * NGLE), (NACE + 1)      'acciones en los extremos de los elementos (FBL)
MTCOPI FRO(), VF3()                             'matriz de verificación de fuerzas en rótulas (VF3)

'llenar primera columna de matrices de resultados de rótulas plásticas (KRO, DRO, FRO, ERO)
AEINIR NRO(), PRO(), RFI(), KRO(), DRO(), FRO(), ERO()

'llenar primera columna de matriz de acciones en los extremos de los elementos (FBL)
AEINIF NELE, NGLE, RFI(), FBL()

'inicializar matrices resumen de iteraciones y errores (ITE, ERR)
MTCONS ITE(), 0, 2, NACE + 1
MTCONS ERR(), 0, 2, NACE + 1

'aceleraciones iniciales de centros de masa (ACM)
ADACIN MDR(), FEQ(), FEX(), ACM()

'coeficientes para métodos de solución directa de las ecuaciones de movimiento (CSD)
ADCONS IDSD, DTIM, CSD()

Debug.Print "Cálculos previos", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.61. Inicialización de matrices de resultados (*cont.*)

6.7.4. Respuesta de la estructura en cada instante de tiempo. Procedimiento principal

En esta etapa se obtiene la respuesta principal de la estructura, sometida a las acciones externas equivalentes $[F_{ext}]$ y considerando la no linealidad debida a la formación de rótulas plásticas. Se utiliza el método de Newmark, combinado con el método de Newton – Raphson modificado, para el cálculo del incremento en los desplazamientos de los centros de masa y de los nudos en un proceso iterativo principal (*externo*), en el que en cada iteración también requiere de un proceso iterativo secundario (*interno*) para calcular el incremento en las acciones equivalentes trasladadas a los centros de masa, producido por el incremento en los desplazamientos de los centros de masa (*Capítulo 5*).

Como resultados de esta etapa se obtienen los desplazamientos, las velocidades y las aceleraciones de los centros de masa al final del intervalo $\{u_\phi(t+\Delta t)\}$, $\{\dot{u}_\phi(t+\Delta t)\}$ y $\{\ddot{u}_\phi(t+\Delta t)\}$, los desplazamientos de los nudos al final del intervalo $\{u(t+\Delta t)\}$, las acciones equivalentes y las acciones de amortiguamiento al final del intervalo $\{F_{s\phi}(t+\Delta t)\}$ y $\{F_{a\phi}(t+\Delta t)\}$ y las acciones en coordenadas locales en los extremos de cada elemento estructural al final del intervalo $\{\bar{f}_e(t+\Delta t)\}$. A continuación se describen las operaciones que se realizan para cada intervalo de tiempo y antes de comenzar el ciclo iterativo.

Con la subrutina *PBRKRO* se revisa si se han producido cambios significativos en las rigideces tangentes de las rótulas plásticas o resortes rotacionales de un intervalo de tiempo al siguiente. Si los cambios no son importantes, se asume que la matriz de rigidez lateral de la estructura, parámetro cuya construcción demanda más esfuerzo computacional, prácticamente no cambia, y por lo tanto no se calcula nuevamente para el intervalo actual.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

Si las rigideces tangentes de los resortes rotacionales cambian apreciablemente respecto al intervalo anterior, con la subrutina *ORMPRO* se actualiza la matriz de propiedades de elementos *PAE()* con las rigideces tangente almacenadas en la matriz *KRO()*. Posteriormente con las subrutinas *AEKGLT*, *AEKGLI* y *MTMUL3* se crean las matrices de rigidez tangente general $[k(t)]$, condensada a los grados de libertad de interés $[\tilde{k}_{\alpha\alpha}(t)]$ y lateral $[k_{\phi}(t)]$. respectivamente Los resultados se almacenan en las matrices reales *KGL(), KDI()* y *KDP()*. La subrutina *AEKGLI* también entrega las submatrices *KAA(), KAG(), KGA()* y *KGG()* ($[k_{\alpha\alpha}(t)]$, $[k_{\alpha\gamma}(t)]$, $[k_{\gamma\alpha}(t)]$ y $[k_{\gamma\gamma}(t)]^{-1}$).

Con la subrutina *ADAMRY* se crea la matriz de amortiguamiento usando el criterio de Rayleigh. Dependiendo de la forma de amortiguamiento especificada en la hoja "*TB_GEN*", esta se calcula una sola vez con la rigidez inicial o al comienzo de cada intervalo con la rigidez tangente. El resultado se almacena en la matriz *AMR()*.

Las matrices auxiliares $[a(t)]$ y $[b(t)]$ se crean con la subrutina *ADMAAB* en función de las matrices de masa y amortiguamiento y de las constantes almacenadas en *CSD()*. Los resultados se almacenan en las matrices *AAA()* y *BBB()*. Similar a como funciona *ADAMRY*, las matrices se calculan una vez o al comienzo de cada intervalo, dependiendo de la forma de amortiguamiento especificada.

La subrutina *ADMARA* calcula la matriz de rigidez tangente efectiva $[\hat{k}_{\phi}(t)]$, la inversa de la matriz de rigidez tangente efectiva $[\hat{k}_{\phi}(t)]^{-1}$ y la matriz $([\hat{k}_{\phi}(t)] - [k_{\phi}(t)])$, cantidades que se utilizan en el ciclo iterativo principal. Los resultados se almacenan respectivamente en la matrices *KTE(), KEI() y KTR()*.

La subrutina *ADIEFE* calcula el vector de incremento efectivo en el intervalo de las acciones externas equivalentes al movimiento de la base $\{\Delta\hat{F}_{ext}\}$. Se crea como resultado el vector columna *FEI()*.

Con la subrutina *ADINVA* se inicializan en cero las variables escalares relacionadas con el número de iteraciones y el error de convergencia de los ciclos interno y externo denominadas *ITED*, *ITEF* y *ERRF*. También se inicializa lleno de ceros el vector *DDN()* en el que se acumularán los incrementos en los desplazamientos de los nudos calculados en cada iteración. Se inicializa el vector *DRJ()* con los valores del vector *FEI()* ($\{\Delta R^{(1)}\} = \{\Delta\hat{F}_{ext}\}$) y en las matrices *ERO(), DRO() y FRO()* de rótulas plásticas, en la matriz *FEQ()* de acciones equivalentes y en la matriz *FBL()* de acciones en los extremos de los elementos, se inicializa la columna de datos correspondiente al instante $t + \Delta t$ (*al final del intervalo*) con los valores correspondientes al instante t (*al comienzo del intervalo*), teniendo en cuenta la forma como funcionan las subrutinas del ciclo principal.

Finalmente con la subrutina *MTMULT* se calcula un primer incremento en los desplazamientos de los centros de masa $\{\Delta u_{\phi}^{(1)}\} = [\hat{k}_{\phi(t)}]^{-1}\{\Delta R^{(1)}\}$ el cual se almacena en el vector *DCJ()*, y se inicializa el vector *DDC()* que almacena el vector $\{\Delta u_{\phi}\}$, para lo cual se copia el vector *DCJ()* con la subrutina *MTCOPI*.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La Figura 6.62 muestra las líneas del código principal con las que se realizan las operaciones anteriormente descritas.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Respuesta de la estructura en cada instante de tiempo - procedimiento principal:
'Respuesta de la estructura en cada instante de tiempo - procedimiento principal:
'Desplazamientos, velocidades y aceleraciones de centros de masa, desplazamientos de nudos en coordenadas
'globales, acciones equivalentes trasladadas a centros de masa, fuerzas y desplazamientos de rótulas plásticas
'y acciones en los extremos de los elementos en coordenadas locales

For ITIM = 1 To NACE

    'revisar si hay cambios importantes en la rigidez tangente de las rótulas plásticas (KRO)
    PBRKRO KRO(), ITIM, CROT

    If (CROT = "SI") Then
        'actualizar matriz de propiedades con rigidez tangente de rótulas al comienzo del intervalo (PAE)
        ORMPRO ITIM, NRO(), KRO(), PAE()

        'matriz de rigidez tangente al comienzo del intervalo (KDP)
        AEKGLT NELE, NGLT, ELE(), INC(), PAU(), RIG(), XYZ(), PAE(), KGL()
        AEKGLI NGLI, NGLO, KGL(), KAA(), KAG(), KGA(), KGG(), KDI()
        MTMUL3 TDR(), KDI(), KDP()
    End If

    'matriz de amortiguamiento de Rayleigh (AMR)
    ADAMRY IDAR, ITIM, FRN(FRNI, 1), FRN(FRNJ, 1), AMVI, MDR(), KDP(), AMR()

    'matrices auxiliares [a] y [b] (AAA, BBB)
    ADMAAB IDAR, ITIM, MDR(), AMR(), CSD(), AAA(), BBB()

    'matrices de rigidez auxiliares (KTE, KEI, KTR)
    ADMARA CSD(), MDR(), AMR(), KDP(), KTE(), KEI(), KTR()

    'incremento efectivo en acciones externas (FEI)
    ADIEFE ITIM, CSD(), FEX(), AAA(), BBB(), VCM(), ACM(), FEI()

    'inicializar variables fuera del ciclo (ITED, ITEF, ERRF, DDN, DRJ, ERO, DRO, FRO, FEQ, FBL)
    ADINVA ITIM, NGLT, ITED, ITEF, ERRF, FEI(), DDN(), DRJ(), ERO(), DRO(), FRO(), FEQ(), FBL()

    'primer incremento de desplazamientos de centros de masa en la iteración actual (DCJ, DDC)
    MTMULT KEI(), DRJ(), DCJ()
    MTCOPI DCJ(), DDC()

    '_____________________________________________________________________________________________
```

Figura 6.62. Respuesta principal de la estructura – instrucciones antes del ciclo iterativo principal

Posteriormente dentro del ciclo principal se realizan las siguientes operaciones:

Con la subrutina *AEDESP* se calcula el vector $\{\Delta u'^{(j)}\}$de incremento supuesto de los desplazamientos en coordenadas globales de los nudos en la iteración actual, el cual por lo general no es correcto dado que no siempre hay linealidad entre acciones y desplazamientos. El resultado se almacena en el vector real *DNJ()*.

Posteriormente con la subrutina *AEDFSJ* se calcula el incremento en las acciones equivalentes trasladadas a los centros de masa $\{\Delta F_{s\phi}^{(j)}\}$ producido por el incremento en los desplazamientos de los centros de masa $\{\Delta u_{\phi}^{(j)}\}$, el cual se almacena en el vector columna *DFS()*. Adicionalmente con esta subrutina se obtienen como resultado el incremento real de los desplazamientos de los nudos $\{\Delta u^{(j)}\}$, el cual se sobrescribe en el vector *DNJ()*, asi como los parámetros de rotación elástica, las rotaciones, los momentos y la rigidez tangente de las rótulas plásticas al final del intervalo que se almacenan en las matrices *ERO()*,

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

DRO(), FRO() y KRO() respectivamente, y las acciones en coordenadas locales en los extremos de los elementos estructurales, que se almacenan en la matriz *FBL()*.

Con las subrutinas *MTCOPI* y *MTSUMA* se calcula el incremento de desplazamientos de nudos acumulado hasta la iteración *j* en el vector *DDN()*.

Con la subrutina *ADFEQF* se calculan las acciones equivalentes al final del intervalo $\{F^{(j)}_{s\phi(t+\Delta t)}\}$, las cuales se almacenan en la columna correspondiente de la matriz *FEQ()*, y se calcula el incremento efectivo de las acciones equivalentes en la iteración actual $\{\Delta F^{(j)}\}$, el cual se almacena en el vector columna *DFJ()*.

Las acciones residuales para la siguiente iteración $\{\Delta R^{(j+1)}\}$ se calculan con las subrutinas *MTCOPI y MTREST* y se sobrescriben en el vector columna *DRJ()*. A partir de estas y con la inversa de la matriz de rigidez tangente efectiva $[\hat{k}_{\phi(t)}]^{-1}$, se calcula un incremento de desplazamientos de centros de masa para la siguiente iteración $\{\Delta u^{(j+1)}_{\phi}\}$ utilizando la subrutina *MTMULT*, el cual se sobrescribe en el vector *DCJ()*. Este se acumula en el vector *DDC()* con las subrutinas *MTCOPI* y *MTSUMA* como se indicó previamente con los desplazamientos de los nudos.

Con la subrutina *ADERRD* se calcula el error ϵ utilizando el criterio de comparación de los trabajos realizados por las acciones residuales y por las acciones externas efectivas. El resultado se almacena en la variable real *ERRD*. Finalmente se cuenta la iteración actual sumando uno al contador *ITED*.

Las anteriores operaciones se realizan mientras el error calculado ϵ sea superior a la tolerancia especificada y mientras no se supere el número máximo de iteraciones especificado, como lo muestra la Figura 6.63.

```
'respuesta de la estructura al final del intervalo actual obtenida de forma iterativa:
Do

    'incremento supuesto de desplazamientos de nudos al final de la iteración actual (DNJ)
    AEDESP NGLD, NGLC, TDR(), DCJ(), KGG(), KGA(), DBB(), DNJ()

    'incremento real de desplazamientos de nudos, acciones equivalentes reales y parámetros de rótulas
    'plásticas al final del intervalo y de la iteración actual (DNJ, FEQ, ERO, DRO, FRO, KRO, DFS, FBL)
    AEDFSJ ITIM, TOLD, ITEF, ERRF, NMIT, INC(), ELE(), PAU(), RIG(), TDT(), XYZ(), PAE(), NRO(), PRO(), _
    KGG(), DNJ(), ERO(), DRO(), FRO(), KRO(), DFS(), FBL()

    'incremento de desplazamientos de nudos acumulados hasta la iteración actual (DDN)
    MTCOPI DDN(), TM1()
    MTSUMA TM1(), DNJ(), DDN()

    'acciones equivalentes al final del intervalo e incremento efectivo de acciones equivalentes en la
    'iteración actual (FEQ, DFJ)
    ADFEQF ITIM, KTR(), DCJ(), DFS(), FEQ(), DFJ()

    'residuo al final de la iteración (DRJ)
    MTCOPI DRJ(), TM1()
    MTREST TM1(), DFJ(), DRJ()

    'incremento de desplazamientos de centros de masa para la siguiente iteración (DCJ)
    MTMULT KEI(), DRJ(), DCJ()

    'incremento acumulado de desplazamientos de centros de masa hasta la siguiente iteración (DDC)
    MTCOPI DDC(), TM1()
    MTSUMA TM1(), DCJ(), DDC()
```

Figura 6.63. Respuesta principal de la estructura – instrucciones del ciclo iterativo principal

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
    'error en la siguiente iteración (ERRD)
    ADERRD DRJ(), DCJ(), FEI(), DDC(), ERRD

    'contar iteración (ITED)
    ITED = ITED + 1

Loop While ((ERRD > TOLD) And (ITED < NMIT))
'-----------------------------------------------------------------------------------------------------
```

Figura 6.63. Respuesta principal de la estructura – instrucciones del ciclo iterativo principal (*cont.*)

Terminado el ciclo iterativo principal con la subrutina *ADREFI* se calculan los desplazamientos, las velocidades y las aceleraciones de los centros de masa al final del intervalo, almacenando los resultados en las columnas correspondientes de las matrices *DCM(), VCM() y ACM()* respectivamente.

Con las subrutinas *MTSUBM*, *MTSUMA* y *MTREMV* se calculan y almacenan en la matriz *DNG()* los desplazamientos de los nudos al final del intervalo. Las acciones de amortiguamiento al final del intervalo se calculan con la subrutina *ADAMFI* y se almacenan en la matriz *FAM()*.

Finamente el numero de iteraciones y los errores finales al momento en que convergen los ciclos interno y externo se almacenan en las matrices *ITE()* y *ERR()*. La Figura 6.64 muestra las líneas del código principal que se ejecutan después del ciclo iterativo.

```
    '-----------------------------------------------------------------------------------------------------

    'desplazamientos, velocidades y aceleraciones de centros de masa al final del intervalo (DCM, VCM, ACM)
    ADREFI ITIM, NDIN, CSD(), DDC(), DCM(), VCM(), ACM()

    'desplazamientos de nudos al final del itervalo almacenados en tabla resumen (DNG)
    MTSUBM DNG(), TM1(), 1, ITIM, (NGLD + NGLC), 1
    MTSUMA TM1(), DDN(), TM2()
    MTREMV DNG(), TM2(), "COLUMNA", (ITIM + 1)

    'acciones de amortiguamiento al final del intervalo (FAM)
    ADAMFI ITIM, VCM(), AMR(), FAM()

    'número de iteraciones y error máximo almacenados en tablas resumen (ITE,ERR)
    ITE(1, ITIM) = ITED
    ITE(2, ITIM) = ITEF
    ERR(1, ITIM) = ERRD
    ERR(2, ITIM) = ERRF

Next ITIM

Debug.Print "Desplazamientos, velocidades y aceleraciones de centros de masa", (Timer - TINI)
Debug.Print "Desplazamientos de nudos", (Timer - TINI)
Debug.Print "Acciones elásticas y de amortiguamiento", (Timer - TINI)
Debug.Print "Fuerzas y desplazamientos de rótulas plásticas", (Timer - TINI)
Debug.Print "Acciones en los extremos de los elementos en coordenadas locales", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------
```

Figura 6.64. Respuesta principal de la estructura – instrucciones posteriores al ciclo iterativo principal

6.7.5. Verificación de la ecuación de movimiento

La verificación de la ecuación de movimiento se realiza de la misma forma en que se describió en la parte de análisis modal cronológico, con la única diferencia que las matrices de acciones equivalentes $[F_{s\phi}]$ y de acciones de amortiguamiento $[F_{a\phi}]$ se han calculado en la

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

etapa anterior. La Figura 6.65 muestra la parte del código principal en la que realiza la verificación de la ecuación de movimiento.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Verificación del equilibrio dinámico

'acciones inerciales trasladadas a centros de masa (FIN)
MTMULT MDR(), ACM(), FIN()

'matriz de verificación del equilibrio dinámico (VF1)
MTSUMA FIN(), FAM(), TM1()
MTSUMA TM1(), FEQ(), VF1()

Debug.Print "Verificación del equilibrio dinámico", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.65. Verificación de la ecuación de movimiento

6.7.6. Balance de energía

La verificación de la ecuación de balance de energía se realiza de forma similar a como se describió para el análisis modal cronológico. En este caso la forma como se determina la energía cinética $[E_c]$, la energía disipada por amortiguamiento natural $[E_a]$, la energía de entrada $[E_e]$ y la forma como se construye la matriz de balance de energía *ENE()* no cambian. La diferencia radica en la forma en que se evalúa la energía de deformación $[E_s]$ que corresponde a la suma de las energías de deformación elástica $[E_{ss}]$ e inelástica $[E_{sy}]$.

La energía de deformación total, $[E_s] = [E_{ss}] + [E_{sy}]$, se calcula con la subrutina *ADENIN* y se almacena en el vector fila *ESS()*, de la forma como se evalúan las energías de amortiguamiento natural o de entrada. No se usa la subrutina *ADENED* ya que la matriz de rigidez no es constante y la expresión directa usada para el caso elástico no es aplicable.

La energía de deformación inelástica $[E_{sy}]$ se calcula en función de los momentos y las rotaciones de las rótulas plásticas o resortes rotacionales no lineales, utilizando el procedimiento descrito en el Capítulo 5. Se utiliza la subrutina *ADENSY* y el resultado obtenido se almacena en el vector fila *ESY()*. Se espera que este vector tenga valores diferentes a cero puesto que se espera que haya disipación de energía por histéresis.

Finalmente la energía de deformación elástica $[E_{ss}]$ equivale a la diferencia entre las energías $[E_s]$ y $[E_{sy}]$. Se utiliza la subrutina *MTREST* y el resultado se almacena en el vector fila *ESE()*. La Figura 6.66 muestra la parte del código principal en la que realiza la verificación de la ecuación de balance de energía.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Balance de energía

'energía cinética (ECI)
ADENED MDR(), VCM(), ECI()

'energía disipada por amortiguamiento natural (EAM)
ADENIN DTIM, FAM(), VCM(), EAM()
```

Figura 6.66. Balance de energía

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'energía de deformación inelástica (ESY)
ADENSY PRO(), DRO(), FRO(), ESY()

'energía de deformación elástica (ESE)
MTREST ESS(), ESY(), ESE()

'energía de entrada (EEX)
ADENIN DTIM, FEX(), VCM(), EEX()

'vector fila de verificación de balance de energía (VF2)
MTSUMA ECI(), EAM(), TM1()
MTSUMA TM1(), ESE(), TM2()
MTSUMA TM2(), ESY(), VF2()

'matriz resumen de balance de energía (ENE)
ADBAEN ECI(), EAM(), ESE(), ESY(), EEX(), VF2(), ENE()

Debug.Print "Balance de energía", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.66. Balance de energía (*cont.*)

6.7.7. Desplazamientos de los nudos

Los desplazamientos relativos a la base y en coordenadas globales de los nudos de la estructura se calcularon previamente en el procedimiento principal, por lo que en esta etapa simplemente se crea la matriz resumen *DEM()* con los desplazamientos máximos y los instantes de tiempo donde se presentan, como lo muestra la Figura 6.67.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Desplazamientos de los nudos relativos a la base y en coordenadas globales

'tabla resumen de desplazamientos máximos probables y tiempos correspondientes ordenados por nudo (DEM)
AETRDE IDMA, MGL(), DNG(), TME(), DEM()

Debug.Print "Desplazamientos de los nudos relativos a la base y en coordenadas globales", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.67. Desplazamientos relativos a la base y en coordenadas globales de los nudos

6.7.8. Derivas de centros de masa y de ejes de columnas

Las derivas de los centros de masa y las derivas de los ejes de columnas se calculan a partir de los desplazamientos almacenados en las matrices *DCM()* y *DNG()* de la misma forma en que se hizo en la parte de análisis modal cronológico, como lo muestra la Figura 6.68 y la Figura 6.69.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Derivas de centros de masa

'matrices de derivas de centros de masa en direcciones X e Y
AEDECM NDIR, DCM(), DRX(), EXX(), EYY()

Debug.Print "Derivas de centros de masa", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.68. Derivas de centros de masa

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Derivas en ejes de columnas (si el pórtico es espacial)

If NDIM = 3 Then

    'para cada eje de columnas:
    For ICOL = 1 To NCOL

        'derivas en el eje de columnas actual como fracción de la altura de piso (DER)
        AEDERI ICOL, DNG(), XYZ(), MGL(), NEC(), NNE(), DER()

        'tabla de derivas máximas del eje actual (TM1)
        ADVMAX DER(), TME(), TM1()

        'escribir tabla de derivas máximas del eje actual (TM1)
        EDIMPR "TB_OUT", "Derivas máximas en eje de columna N° " & ICOL, FILA, COLM, TM1()

        'crear archivo de texto con la matriz DEV()
        IOEXMR DER(), TME(), "DER-" & ICOL

    Next ICOL

    Debug.Print "Derivas en ejes de columnas", (Timer - TINI)

 End If
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.69. Derivas en ejes de columnas

6.7.9. Cortantes de piso

Los cortantes de piso se calculan a partir de las acciones equivalentes trasladadas a los centros de masa de la misma forma como se describió en la parte de análisis modal cronológico, como se muestra en la Figura 6.70.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Cortantes de piso

'tablas de cortantes de piso en direcciones X e Y (VPX, VPY)
AECORP FEQ(), NDIM, VPX(), VPY()

'tablas resumen para graficar diagramas de cortantes de piso (DVX, DVY)
AEDVPI DRX(), VPX(), DVX()
AEDVPI DRX(), VPY(), DVY()

'crear envolventes y adjuntarlas al final de las tablas para graficar diagramas
AENVOL DVX()
AENVOL DVY()

Debug.Print "Cortantes de piso", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.70. Cortantes de piso

6.7.10. Acciones en los extremos de los elementos

Las acciones en coordenadas locales en los extremos de los elementos se calcularon previamente en la etapa donde se obtiene la respuesta principal de la estructura, y se encuentran almacenadas en la matriz *FBL()*.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

Para cada elemento se crea la matriz $[\bar{f}_e]$ utilizando la subrutina *MTSUBM*, con la que se extraen de la matriz *FBL()* las acciones del elemento en todos los instantes de tiempo, y se almacenan en la matriz temporal *FEL()*. Nuevamente con la subrutina *IOEXMR* se exporta la matriz *FEL()* a un archivo de texto y con la subrutina *AETRFE* se crea la tabla resumen *FEM()* que tiene la misma forma a la descrita en la parte de análisis modal cronológico.

Posteriormente se identifica si el elemento actual posee rótulas plásticas asociadas, lo cual se hace con la subrutina *PBELRP*, con el fin de crear una matriz similar a *FRO()* con las acciones en los extremos de los elementos que tengan asociadas rótulas plásticas. Esta matriz se denomina *VF3()*, se crea con la subrutina *AEVFRO* y sirve para verificar que las acciones en los extremos donde hay rótulas coinciden con las acciones calculadas en las rótulas plásticas o resortes rotacionales.

La Figura 6.71 muestra las líneas del código principal con las que se crea un archivo de texto con la matriz *FEL()* de cada elemento, y con las que se crean las tablas resumen *FEM()* y *VF3()* previamente mencionadas.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Acciones en los extremos de los elementos en coordenadas locales

'inicializar matriz FEM() llena de ceros
MTCONS FEM(), 0, NELE, 2 * NGLE

'para cada elemento de la estructura:
For IELE = 1 To NELE

    'vector de acciones en los extremos del elemento en coordenadas locales (FEL)
    FILE = 1 + NGLE * (IELE - 1)
    MTSUBM FBL(), FEL(), FILE, 1, NGLE, NACE + 1

    'calcular y almacenar acciones maximas del elemento y tiempos correspondientes en matriz resumen (FEM)
    AETRFE IDMA, IELE, FEL(), TME(), FEM()

    'crear archivo de texto con matriz FEL()
    IOEXMR FEL(), TME(), "FEL-" & IELE

    'identificar si el elemento posee rótulas plásticas (ROTU)
    PBELRP IELE, NRO(), ROTU

    'almacenar acciones asociadas a rótulas en matriz de verificación (VF3)
    AEVFRO IELE, (NACE + 1), ROTU, NRO(), FEL(), VF3()

Next IELE

Debug.Print "Acciones en los extremos de los elementos en coordenadas locales", (Timer - TINI)
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.71. Acciones en coordenadas locales en los extremos de los elementos

6.7.11. Diagramas de acciones internas

Las tablas de acciones internas *NXX(), VYY(), VZZ(), TXX(), MYY()* y *MZZ()* de cada eje de vigas y columnas, se calculan nuevamente con la subrutina *AEDIVC* a partir de las acciones en coordenadas locales en los extremos de los elementos, con la diferencia que ahora se incluye en cada sección el efecto de las cargas iniciales, para lo cual simplemente se deben

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

especificar las características de estas cargas, las cuales están almacenadas en las tablas de datos *TWE()* y *WEL()*. A diferencia del caso de análisis modal cronológico o espectral, la subrutina ahora si utiliza estas tablas.

La creación de las envolventes y de los archivos de texto se realiza de la misma forma como se describió en la parte de análisis modal cronológico. La Figura 6.72 muestra las líneas del código principal con las que se crean y exportan las matrices *NXX(), VYY(), VZZ(), TXX(), MYY()* y *MZZ()* de cada eje de vigas y columnas.

```
'-----------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------
'Diagramas de acciones internas en vigas y columnas

'diagramas de acciones internas de vigas
For IVIG = 1 To NVIG

    'tablas de acciones internas del eje de vigas actual (NXX, VYY, VZZ, TXX, MYY, MZZ)
    AEDIVC IDMA, IVIG, NACE + 1, VIG(), ELE(), CTS(), XYZ(), TWE(), WEL(), NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

    'crear envolventes y adjuntarlas al final de la tablas de acciones internas del eje actual
    AEDIEN NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

    'crear archivos de texto con tablas de acciones internas
    IOEXDI IVIG, NDIM, "V", NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

Next IVIG

'diagramas de acciones internas de columnas
For ICOL = 1 To NCOL

    'tablas de acciones internas del eje de columnas actual (NXX, VYY, VZZ, TXX, MYY, MZZ)
    AEDIVC IDMA, ICOL, NACE + 1, COL(), ELE(), CTS(), XYZ(), TWE(), WEL(), NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

    'crear envolventes y adjuntarlas al final de la tablas de acciones internas del eje actual
    AEDIEN NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

    'crear archivos de texto con tablas de acciones internas
    IOEXDI ICOL, NDIM, "C", NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

Next ICOL

Debug.Print "Diagramas de acciones internas en vigas y columnas", (Timer - TINI)
-----------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------
```

Figura 6.72. Diagramas de acciones internas

6.8. Parte VII: Escritura y almacenamiento de resultados

La escritura de las matrices de resultados en la hoja "*TB_OUT*" se realiza con las subrutinas *EDIMPI* y *EDIMPR*, la primera para matrices enteras y la segunda para matrices reales.

En las etapas de cálculo implementadas para los métodos de análisis modal cronológico y análisis dinámico inelástico, en las que se obtiene la respuesta de la estructura en cada instante de tiempo, se utiliza la subrutina *ADVMAX* con la que se obtienen los valores máximos absolutos y los instantes de tiempo en los que tales valores se presentan. Como resultado se crea una tabla temporal que se escribe en la hoja "*TB_OUT*".

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La creación de archivos de texto se realiza con las subrutinas *IOEXPI, IOEXPR y IOEXMR*. Las dos primeras exportan la matriz de datos sin ninguna modificación, mientras que la tercera, que se utiliza para tablas de parámetros que varían en el tiempo, transpone los datos y adjunta una primera columna con los instantes de tiempo considerados en el análisis, facilitando la construcción de gráficas de respuesta en función del tiempo.

A continuación se muestran las líneas del código principal con las que se escriben y se exportan las tablas de datos de entrada y las tablas de resultados obtenidas por alguno de los métodos de análisis dinámico. Algunas líneas se encuentran inactivas (*en color verde*), sin embargo se activan fácilmente entrando al código principal y borrando la comilla al comienzo de cada línea.

6.8.1. Tablas auxiliares

La Figura 6.73 muestra las líneas del código principal con las que se escriben y exportan las tablas auxiliares que crea el programa para el funcionamiento de otras subrutinas. Tales tablas son las matrices de numeración de grados de libertad *MGL()* e *INC()*, las tablas de nudos y elementos asociados a ejes de vigas y columnas *NEC(), NNE(), COL(), VIG()*, la tabla de propiedades de elementos *PAE()*, las tablas de nudos asociados a diafragmas rígidos *NDR()* y *CND()*, la matriz de asociación de desplazamientos *TDR()* y la matriz de coeficientes de correlación *COR()*.

```
'-----------------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------------
'TABLAS AUXILIARES
'-----------------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------------
'escribir y exportar tablas de numeración de grados de libertad (MGL, INC)
EDIMPI "TB_OUT", "Matriz de numeración de GL por nudo - MGL()", FILA, COLM, MGL()
EDIMPI "TB_OUT", "Matriz de incidencias - INC()", FILA, COLM, INC()
IOEXPI MGL(), "MGL"
IOEXPI INC(), "INC"
'-----------------------------------------------------------------------------------------------------------------------
'escribir y exportar tablas de nudos y elementos asociados a ejes de vigas y columnas (NEC, NNE, COL, VIG)
'EDIMPI "TB_OUT", "Nudos asociados a cada eje de columna - NEC()", FILA, COLM, NEC()
'EDIMPI "TB_OUT", "Cantidad de nudos asociados a cada eje de columna - NNE()", FILA, COLM, NNE()
'EDIMPI "TB_OUT", "Elementos asociados a cada eje de columna - COL()", FILA, COLM, COL()
'EDIMPI "TB_OUT", "Elementos asociados a cada eje de vigas - VIG()", FILA, COLM, VIG()
IOEXPI NEC(), "NEC"
IOEXPI NNE(), "NNE"
IOEXPI COL(), "COL"
IOEXPI VIG(), "VIG"
'-----------------------------------------------------------------------------------------------------------------------
'escribir y exportar tabla de propiedades de elementos (PAE)
'EDIMPR "TB_OUT", "Tabla de propiedades de elementos - PAE()", FILA, COLM, PAE()
IOEXPR PAE(), "PAE"
'-----------------------------------------------------------------------------------------------------------------------
'escribir y exportar tablas de nudos asociados a diafragmas rígidos (NDR, CND)
'EDIMPI "TB_OUT", "Tabla de nudos asociados a cada diafragma rígido - NDR()", FILA, COLM, NDR()
'EDIMPI "TB_OUT", "Tabla de cantidad de nudos asociados a cada diafragma rígido - CND()", FILA, COLM, CND()
IOEXPI NDR(), "NDR"
IOEXPI CND(), "CND"
'-----------------------------------------------------------------------------------------------------------------------
'escribir y exportar matriz geométrica [A] (TDR)
'EDIMPR "TB_OUT", "Matriz de asociación de desplazamientos (matriz geométrica)-TDR()", FILA, COLM, TDR(), 0
IOEXPR TDR(), "TDR"
'-----------------------------------------------------------------------------------------------------------------------
'escribir y exportar matriz de coeficientes de correlación para el método CQC (COR)
'EDIMPR "TB_OUT", "Matriz de coeficientes de correlación - COR()", FILA, COLM, COR(), 0
'-----------------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------------
```

Figura 6.73. Tablas auxiliares

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

6.8.2. Matrices de masa, rigidez y amortiguamiento

La Figura 6.74 muestra las líneas del código principal con las que se escriben y exportan a archivos de texto las matrices de rigidez inicial *KGL()*, *KDI()* y *KDP()*, la tabla de coordenadas de centros de masa y de rigidez *DRX()*, la matriz de masa concentrada *MDR()* y la matriz de amortiguamiento *AMR()*, la cual se imprime y exporta solo si se utiliza el método de análisis modal cronológico.

```
'-----------------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------------
'MATRICES DE MASA, RIGIDEZ Y AMORTIGUAMIENTO
'-----------------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------------
'escribir y exportar matriz de rigidez de la estructura considerando todos los grados de libertad (KGL)
'EDIMPR "TB_OUT", "Matriz de rigidez de la estructura - KGL()", FILA, COLM, KGL(), 0
IOEXPR KGL(), "KGL"
'-----------------------------------------------------------------------------------------------------------------------
'escribir y exportar matriz de rigidez condensada a grados de libertad de interés (KDI)
'EDIMPR "TB_OUT", "Matriz de rigidez condensada a GL's de interés - KDI()", FILA, COLM, KDI(), 0
IOEXPR KDI(), "KDI"
'-----------------------------------------------------------------------------------------------------------------------
'escribir y exportar matriz de rigidez lateral (KDP)
EDIMPR "TB_OUT", "Matriz de rigidez lateral - KDP()", FILA, COLM, KDP()
IOEXPR KDP(), "KDP"
'-----------------------------------------------------------------------------------------------------------------------
'escribir y exportar tabla de coordenadas de centros de masa y centros de rigidez (DRX)
If NDIM = 3 Then
    EDIMPR "TB_OUT", "Coordenadas de centros de masa y de rigidez - DRX()", FILA, COLM, DRX()
    IOEXPR DRX(), "DRX"
End If
'-----------------------------------------------------------------------------------------------------------------------
'escribir y exportar matriz de masa (MDR)
EDIMPR "TB_OUT", "Matriz de masa - MDR()", FILA, COLM, MDR()
IOEXPR MDR(), "MDR"
'-----------------------------------------------------------------------------------------------------------------------
'escribir y exportar matriz de amortiguamiento (AMR)
If IDMA = "MODAL CRONOLÓGICO" Then
    EDIMPR "TB_OUT", "Matriz de amortiguamiento - AMR()", FILA, COLM, AMR()
    IOEXPR AMR(), "AMR"
End If
'-----------------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------------
```

Figura 6.74. Matrices de masa, rigidez y amortiguamiento

6.8.3. Análisis modal

La Figura 6.75 muestra las líneas del código principal con las que se escriben y exportan a archivos de texto las matrices que almacenan las frecuencias, los periodos y los modos de vibración natural de la estructura *FRN()*, *PER()* y *MOV()*, las matrices modales de masa, rigidez y amortiguamiento *MMM()*, *MMK()* y *MMA()*, la matriz de influencia *GAM()* y las matrices de coeficientes de participación y de masa modal efectiva *ALF()*, *MME()*, *MMP()* y *MTE()*.

```
'-----------------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------------
'ANÁLISIS MODAL
'-----------------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------------
'escribir y exportar modos, frecuencias y periodos de vibración (MOV, FRN, PER)
EDIMPR "TB_OUT", "Modos de vibración ortonormalizados - MOV()", FILA, COLM, MOV()
EDIMPR "TB_OUT", "Frecuencias naturales de vibración - FRN()", FILA, COLM, FRN()
EDIMPR "TB_OUT", "Periodos de vibración natural - PER()", FILA, COLM, PER()
IOEXPR MOV(), "MOV"
IOEXPR FRN(), "FRN"
IOEXPR PER(), "PER"
'-----------------------------------------------------------------------------------------------------------------------
```

Figura 6.75. Matrices de análisis modal

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
'-----------------------------------------------------------------------------------------------------------------------
'escribir y exportar matrices modales de Masa, Rigidez y Amortiguamiento (MMM, MMK, MMA)
'EDIMPR "TB_OUT", "Matriz Modal de Masa - MMM()", FILA, COLM, MMM()
'EDIMPR "TB_OUT", "Matriz Modal de Rigidez - MMK()", FILA, COLM, MMK()
'EDIMPR "TB_OUT", "Matriz Modal de Amortiguamiento - MMA()", FILA, COLM, MMA()
IOEXPR MMM(), "MMM"
IOEXPR MMK(), "MMK"
IOEXPR MMA(), "MMA"
'-----------------------------------------------------------------------------------------------------------------------
'escribir y exportar matrices de coeficientes de participación y masa modal efectiva (GAM, ALF, MME, MMP, MTE)
'EDIMPR "TB_OUT", "Matriz de influencia - GAM()", FILA, COLM, GAM()
EDIMPR "TB_OUT", "Matriz de Coeficientes de Participación - ALF()", FILA, COLM, ALF()
EDIMPR "TB_OUT", "Matriz de Masa Modal Efectiva - MME()", FILA, COLM, MME()
EDIMPR "TB_OUT", "Matriz de Masa Modal Efectiva en porcentaje - MMP()", FILA, COLM, MMP()
EDIMPR "TB_OUT", "Matriz de Masa Total de la Estructura - MTE()", FILA, COLM, MTE()
IOEXPR GAM(), "GAM"
IOEXPR ALF(), "ALF"
IOEXPR MME(), "MME"
IOEXPR MMP(), "MMP"
IOEXPR MTE(), "MTE"
'-----------------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------------
```

Figura 6.75. Matrices de análisis modal (*cont.*)

6.8.4. Análisis modal cronológico y análisis dinámico inelástico

A continuación se muestran las líneas del código principal con las que se escriben y exportan a archivos de texto las matrices que almacenan los resultados obtenidos por los métodos de análisis modal cronológico y análisis dinámico inelástico.

La Figura 6.78 muestra la forma como se escriben y exportan las matrices de aceleración de la base *ACE()* y *MUO()*, las matrices de coordenadas generalizadas que se obtienen en el caso de análisis modal cronológico *DCG()*, *VCG()* y *ACG()*.

```
'-----------------------------------------------------------------------------------------------------------------------
'-----------------------------------------------------------------------------------------------------------------------
'ANÁLISIS MODAL CRONOLÓGICO Y ANÁLISIS DINÁMICO INELÁSTICO
'-----------------------------------------------------------------------------------------------------------------------
If (IDMA = "MODAL CRONOLÓGICO") Or (IDMA = "DINÁMICO INELÁSTICO") Then
    '-------------------------------------------------------------------------------------------------------------------
    'aceleración de la base:
    'exportar matrices ACE() y MUO()
    IOEXMR ACE(), TME(), "ACE"
    IOEXMR MUO(), TME(), "MUO"
    '-------------------------------------------------------------------------------------------------------------------
    'coordenadas generalizadas:
    If IDMA = "MODAL CRONOLÓGICO" Then
        'exportar matrices TEI(), DCG(), VCG() y ACG()
        IOEXMR TEI(), TME(), "TEI"
        IOEXMR DCG(), TME(), "DCG"
        IOEXMR VCG(), TME(), "VCG"
        IOEXMR ACG(), TME(), "ACG"

        'calcular y escribir valores máximos absolutos y tiempos correspondientes (TM1, TM2, TM3)
        ADVMAX DCG(), TME(), TM1()
        ADVMAX VCG(), TME(), TM2()
        ADVMAX ACG(), TME(), TM3()
        EDIMPR "TB_OUT", "Desplazamientos generalizados máximos", FILA, COLM, TM1()
        EDIMPR "TB_OUT", "Velocidades generalizadas máximas", FILA, COLM, TM2()
        EDIMPR "TB_OUT", "Aceleraciones generalizadas máximas", FILA, COLM, TM3()
    End If
    '-------------------------------------------------------------------------------------------------------------------
```

Figura 6.76. Tablas de aceleración de la base y coordenadas generalizadas

La Figura 6.77 muestra la forma como se escriben y exportan las matrices de desplazamientos, acciones fijas y acciones en los extremos de los elementos producidas por las car-

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

gas iniciales *RDI()*, *RFF()* y *RFI()*, las tablas de desplazamientos, velocidades y aceleraciones de los centros de masa *DCM()*, *VCM()* y *ACM()*, las matrices de resultados de rótulas plásticas *ERO()*, *DRO()*, *FRO()* y *KRO()*, las matrices de iteraciones y errores *ITE()* y *ERR()* y las matrices de verificación de la ecuación de movimiento *FIN()*, *FAM()*, *FEQ()*, *FEX()* y *VF1()*.

```
'desplazamientos y acciones internas debidas a cargas iniciales
If IDMA = "DINÁMICO INELÁSTICO" Then
    'escribir y exportar la matriz RDI()
    EDIMPR "TB_OUT", "Desplazamientos iniciales de los nudos - RDI()", FILA, COLM, RDI()
    IOEXPR RDI(), "RDI"

    'escribir y exportar matrices RFF() y RFI()
    EDIMPR "TB_OUT", "Acciones fijas por elemento debidas a cargas iniciales - RFF()", FILA, COLM, RFF()
    EDIMPR "TB_OUT", "Acciones iniciales en los extremos de los elementos - RFI()", FILA, COLM, RFI()
    IOEXPR RFF(), "RFF"
    IOEXPR RFI(), "RFI"
End If
'-----------------------------------------------------------------------------------------------------------
'Desplazamientos, velocidades y aceleraciones relativas a la base de centros de masa:
'exportar matrices DCM(), VCM() y ACM()
IOEXMR DCM(), TME(), "DCM"
IOEXMR VCM(), TME(), "VCM"
IOEXMR ACM(), TME(), "ACM"

'calcular y escribir valores máximos absolutos y tiempos correspondientes (TM1, TM2, TM3)
ADVMAX DCM(), TME(), TM1()
ADVMAX VCM(), TME(), TM2()
ADVMAX ACM(), TME(), TM3()
EDIMPR "TB_OUT", "Desplazamientos máximos de centros de masa", FILA, COLM, TM1()
EDIMPR "TB_OUT", "Velocidades máximas de centros de masa", FILA, COLM, TM2()
EDIMPR "TB_OUT", "Aceleraciones máximas de centros de masa", FILA, COLM, TM3()
'-----------------------------------------------------------------------------------------------------------
'Rótulas plásticas
If IDMA = "DINÁMICO INELÁSTICO" Then
    'exportar matrices de resultados de rótulas en función del tiempo (ERO, DRO, FRO, KRO)
    IOEXMR ERO(), TME(), "ERO"
    IOEXMR DRO(), TME(), "DRO"
    IOEXMR FRO(), TME(), "FRO"
    IOEXMR KRO(), TME(), "KRO"
    IOEXMR VF3(), TME(), "VF3"

    'crear y exportar matrices de relaciones fuerza-desplazamiento de rótulas plásticas
    For IROT = 1 To ROTE
        MTSUBM DRO(), TM1(), IROT, 1, 1, NACE + 1
        MTSUBM FRO(), TM2(), IROT, 1, 1, NACE + 1
        MTADMR TM3(), TM1(), TM2(), 1
        MTTRAN TM3(), TM4()
        IOEXPR TM4(), "ROT" & IROT
    Next IROT
End If
'-----------------------------------------------------------------------------------------------------------
'Iteraciones y errores máximos
If IDMA = "DINÁMICO INELÁSTICO" Then
    'exportar matrices resumen (ITE, ERR)
    IOEXMR ITE(), TME(), "ITE"
    IOEXMR ERR(), TME(), "ERR"
End If
'-----------------------------------------------------------------------------------------------------------
'Verificación del equilibrio dinámico:
'exportar matrices FIN(), FAM(), FEQ(), FEX() y VF1()
IOEXMR FIN(), TME(), "FIN"
IOEXMR FAM(), TME(), "FAM"
IOEXMR FEQ(), TME(), "FEQ"
IOEXMR FEX(), TME(), "FEX"
IOEXMR VF1(), TME(), "VF1"

'calcular y escribir valores máximos absolutos y tiempos correspondientes (TM1, TM2, TM3)
ADVMAX FIN(), TME(), TM1()
ADVMAX FAM(), TME(), TM2()
ADVMAX FEQ(), TME(), TM3()
EDIMPR "TB_OUT", "Acciones inerciales máximas", FILA, COLM, TM1()
EDIMPR "TB_OUT", "Acciones de amortiguamiento máximas", FILA, COLM, TM2()
EDIMPR "TB_OUT", "Acciones elásticas máximas", FILA, COLM, TM3()
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.77. Tablas de aceleración de la base y coordenadas generalizadas

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La Figura 6.78 muestra la forma como se escribe y exporta las matriz de balance de energía *ENE()*,la matriz de desplazamientos de los nudos *DNG()*, las matrices de derivas de centros de masa *EXX()* y *EYY()*, las matrices de cortantes de piso *VPX()* y *VPY()* y la tabla resumen de acciones máximas en los extremos de los elementos *FEM()*. Las matrices de derivas de los centros de masa de los cortantes de piso en dirección *y* global se imprimen y exportan solamente si el pórtico es espacial.

```
'-----------------------------------------------------------------------------------------------------------
'Balance de energía:
'exportar matriz ENE()
IOEXMR ENE(), TME(), "ENE"
'-----------------------------------------------------------------------------------------------------------
'Desplazamientos relativos a la base y en coordenadas globales de nudos:
'exportar matrices DNG() y DEM()
IOEXMR DNG(), TME(), "DNG"
IOEXPR DEM(), "DEM"

'escribir matriz DEM()
EDIMPR "TB_OUT", "Desplazamientos máximos de nudos y tiempos correspondientes-DEM()", FILA, COLM, DEM()
'-----------------------------------------------------------------------------------------------------------
'Derivas de centros de masa:

'derivas en dirección X global para pórticos planos y espaciales
If (NDIM = 2) Or (NDIM = 3) Then
    'exportar matriz EXX()
    IOEXMR EXX(), TME(), "EXX"

    'calcular y escribir derivas máximas en dirección X (TM1)
    ADVMAX EXX(), TME(), TM1()
    EDIMPR "TB_OUT", "Derivas máximas de centros de masa en dirección X", FILA, COLM, TM1()
End If

'derivas en dirección Y global para pórticos espaciales
If NDIM = 3 Then
    'exportar matriz EYY()
    IOEXMR EYY(), TME(), "EYY"

    'calcular y escribir derivas máximas en dirección Y (TM2)
    ADVMAX EYY(), TME(), TM2()
    EDIMPR "TB_OUT", "Derivas máximas de centros de masa en dirección Y", FILA, COLM, TM2()
End If
'-----------------------------------------------------------------------------------------------------------
'Cortantes de piso:

'cortantes de piso en dirección X global para pórticos planos y espaciales
If (NDIM = 2) Or (NDIM = 3) Then
    'exportar matrices VPX() y DVX()
    IOEXMR VPX(), TME(), "VPX"
    IOEXPR DVX(), "DVX"

    'calcular y escribir cortantes de piso máximos en dirección X (TM1)
    ADVMAX VPX(), TME(), TM1()
    EDIMPR "TB_OUT", "Cortantes de piso máximos en X", FILA, COLM, TM1()
End If

'cortantes de piso en dirección Y global para pórticos espaciales
If NDIM = 3 Then
    'exportar matrices VPY() y DVY()
    IOEXMR VPY(), TME(), "VPY"
    IOEXPR DVY(), "DVY"

    'calcular y escribir cortantes de piso máximos en dirección Y (TM2)
    ADVMAX VPY(), TME(), TM2()
    EDIMPR "TB_OUT", "Cortantes de piso máximos en Y", FILA, COLM, TM2()
End If
'-----------------------------------------------------------------------------------------------------------
'Acciones en los extremos de los elementos en coordenadas locales:

'escribir y exportar matriz FEM()
EDIMPR "TB_OUT", "Acciones máximas en los extremos de los elementos - FEM", FILA, COLM, FEM()
IOEXPR FEM(), "FEM"
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.78. Tablas adicionales de análisis modal cronológico y análisis dinámico inelástico

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

6.8.5. Análisis modal espectral

A continuación se muestran las líneas del código principal con las que se escriben y exportan a archivos de texto las matrices que almacenan los resultados obtenidos por el método de análisis modal espectral.

La Figura 6.79 muestra la forma como se escriben y exportan las matrices que almacenan el espectro de pseudo – aceleraciones, las pseudo – aceleraciones espectrales asociadas a cada modo y los desplazamientos generalizados *ESP(), SAT()* y *GEN()*, asi como las matrices resumen de desplazamientos máximos de centros de masa *XCR(), YCR()* y *ZCR()*, las tablas resumen de desplazamientos máximos de nudos *DNR() y DEM()* y las tablas resumen de derivas máximas de centros de masa *EXR()* y *EYR()*. Algunas tablas resumen se escriben y exportan solamente si el pórtico analizado es espacial.

```
'-----------------------------------------------------------------------------------------------------------
'ANÁLISIS MODAL ESPECTRAL
'-----------------------------------------------------------------------------------------------------------
'Desplazamientos generalizados máximos:

'escribir matrices ESP(), SAT() y GEN()
EDIMPR "TB_OUT", "Espectro de pseudo-aceleraciones - ESP()", FILA, COLM, ESP()
EDIMPR "TB_OUT", "Vector de pseudo-aceleraciones interpoladas - SAT()", FILA, COLM, SAT()
EDIMPR "TB_OUT", "Matriz de desplazamientos generalizados máximos - GEN()", FILA, COLM, GEN()

'exportar matrices ESP(), SAT() y GEN()
IOEXPR ESP(), "ESP"
IOEXPR SAT(), "SAT"
IOEXPR GEN(), "GEN"
'-----------------------------------------------------------------------------------------------------------
'Desplazamientos máximos probables de centros de masa:

'escribir y exportar matrices XCR() para pórticos planos
If (NDIM = 2) Or (NDIM = 3) Then
    EDIMPR "TB_OUT", "Desplazamientos máximos en X de CM's de diafragmas - XCR()", FILA, COLM, XCR()
    IOEXPR XCR(), "XCR"
End If

'escribir y exportar matrices YCR() Y ZCR() para pórticos espaciales
If NDIM = 3 Then
    EDIMPR "TB_OUT", "Desplazamientos máximos en Y de CM's de diafragmas - YCR()", FILA, COLM, YCR()
    EDIMPR "TB_OUT", "Rotaciones máximas en Z de CM's de diafragmas - ZCR()", FILA, COLM, ZCR()
    IOEXPR YCR(), "YCR"
    IOEXPR ZCR(), "ZCR"
End If
'-----------------------------------------------------------------------------------------------------------
'Desplazamientos máximos probables de los nudos en coordenadas globales:

'escribir y exportar matrices DNR() y DEM()
EDIMPR "TB_OUT", "Desplazamientos máximos probables de nudos - DEM()", FILA, COLM, DEM()
IOEXPR DNR(), "DNR"
IOEXPR DEM(), "DEM"
'-----------------------------------------------------------------------------------------------------------
'Derivas máximas probables de centros de masa:

'escribir y exportar matriz EXR() para pórticos planos y espaciales
If (NDIM = 2) Or (NDIM = 3) Then
    EDIMPR "TB_OUT", "Derivas máximas de CM's en dirección X - EXR()", FILA, COLM, EXR()
    IOEXPR EXR(), "EXR"
End If

'escribir y exportar matriz EYR() para pórticos espaciales
If NDIM = 3 Then
    EDIMPR "TB_OUT", "Derivas máximas de CM's en dirección Y - EYR()", FILA, COLM, EYR()
    IOEXPR EYR(), "EYR"
End If
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.79. Tablas de análisis modal espectral (*cont.*)

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La Figura 6.80 muestra la forma como se escriben y exportan las matrices resumen de acciones máximas equivalentes trasladadas a los centros de masa *FXR()*, *FYR()* y *MZR()*, las tablas resumen de cortantes de piso *VXR()*, *VYR()*, *DVX() y DVY()* y las tabla resumen de acciones máximas en los extremos de los elementos *FEM()*. Nuevamente algunas tablas resumen se escriben y exportan solamente si el pórtico analizado es espacial.

```
'-----------------------------------------------------------------------------------------------------------
'Acciones elásticas máximas probables trasladadas a centros de masa:

'escribir y exportar matriz FXR() para pórticos planos y espaciales
If (NDIM = 2) Or (NDIM = 3) Then
    EDIMPR "TB_OUT", "Fuerzas elásticas máximas en dirección X - FXR()", FILA, COLM, FXR()
    IOEXPR FXR(), "FXR"
End If

'escribir y exportar matrices FYR() y MZR() para pórticos espaciales
If NDIM = 3 Then
    EDIMPR "TB_OUT", "Fuerzas elásticas máximas en dirección Y - FYR()", FILA, COLM, FYR()
    EDIMPR "TB_OUT", "Momentos elásticos máximos en dirección Z - MZR()", FILA, COLM, MZR()
    IOEXPR FYR(), "FYR"
    IOEXPR MZR(), "MZR"
End If
'-----------------------------------------------------------------------------------------------------------
'Cortantes de piso máximos probables:

'escribir y exportar matrices VXR() y DVX() para pórticos planos y espaciales
If (NDIM = 2) Or (NDIM = 3) Then
    EDIMPR "TB_OUT", "Tabla resumen de cortantes de piso máximos en X - VXR()", FILA, COLM, VXR()
    IOEXPR VXR(), "VXR"
    IOEXPR DVX(), "DVX"
End If

'escribir y exportar matrices VYR() y DVY() para pórticos espaciales
If NDIM = 3 Then
    EDIMPR "TB_OUT", "Tabla resumen de cortantes de piso máximos en Y - VYR()", FILA, COLM, VYR()
    IOEXPR VYR(), "VYR"
    IOEXPR DVY(), "DVY"
End If
'-----------------------------------------------------------------------------------------------------------
'Acciones máximas probables en coordenadas locales en los extremos de los elementos:

'escribir y exportar matrices FEM()
EDIMPR "TB_OUT", "Acciones máximas en los extremos de los elementos - FEM()", FILA, COLM, FEM()
IOEXPR FEM(), "FEM"
'-----------------------------------------------------------------------------------------------------------
```

Figura 6.80. Tablas de análisis modal espectral

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

Capítulo 7
Descripción de las subrutinas desarrolladas

En este capítulo se describe de forma más detallada el funcionamiento de las subrutinas secundarias nuevas, mencionadas en el Capítulo 6 en el que se describió la subrutina principal *UNDIN*. L a Tabla 7.1 indica el módulo en el que se encuentra almacenado cada grupo de subrutinas desarrolladas o complementadas.

Módulo	Descripción
`MdAE`	Análisis estructural
`MdAD`	Análisis dinámico
`MdK, MdF`	Crear matrices de rigidez y vectores de acciones fijas
`MdVP`	Calcular modos de vibración
`MdMT`	Operaciones matriciales básicas
`MdNGL`	Numeración de grados de libertad
`MdOR`	Organización de matrices
`MdTR`	Cambios de base
`MdIO`	Importar o exportar matrices
`MdPB`	Uso común

Tabla 7.1. Módulos de almacenamiento de subrutinas nuevas

7.1. Instrucciones para análisis estructural

7.1.1. AEMPAE

La subrutina *AEMPAE* crea la tabla de propiedades por elemento *PAE()* que se utiliza para construir la matriz de rigidez general de la estructura, inicial o tangente al comienzo de ca-

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

da intervalo, dependiendo del método de análisis dinámico utilizado. La sintaxis y los parámetros que intervienen en la subrutina se describen en la Tabla 7.2.

AEMPAE MAT(), SEC(), AMS(), JRS(), PAE()

Variable	Tipo	Descripción
MAT()	Real	Tabla de materiales tipo
SEC()	Real	Tabla de secciones tipo
AMS()	Entero	Tabla de asignación de materiales y de secciones a elementos
JRS()	Real	Tabla de coeficientes de rigidez inicial de conexiones elásticas
PAE()	Real	Tabla de propiedades por elemento, creada

Tabla 7.2. Sintaxis y argumentos de la instrucción AEMPAE

Inicialmente se toma la información de las tablas de materiales tipo *MAT()* y de secciones tipo *SEC()* y, con base en el tipo de material y de sección asignado a cada elemento almacenado en la tabla *AMS()*, se crea una tabla parcial de 7 columnas con las propiedades del material y de la sección de cada elemento. Posteriormente la tabla de coeficientes de rigidez inicial de las conexiones elásticas *JRS()* se adjunta al final de la tabla parcial previamente creada, para conformar la tabla *PAE()* definitiva.

7.1.2. AECRIG

La subrutina *AECRIG* calcula y almacena en la tabla *DRX()* las coordenadas de los centros de rigidez de los diafragmas. La sintaxis de la instrucción y la descripción de los parámetros que intervienen se muestran en la Tabla 7.3.

AECRIG DRX(), KPD()

Argumento	Tipo	Descripción
KDP()	Real	Matriz de rigidez lateral $[k_\phi]$
DRX()	Real	Tabla de coordenadas de centros de masa, modificada

Tabla 7.3. Sintaxis y argumentos de la instrucción AECRIG

Inicialmente la matriz de rigidez lateral *KDP()* se invierte, conformado la matriz de flexibilidad lateral, de la cual se extraen las rotaciones $\theta_{zz}, \theta_{zx}\ y\ \theta_{zy}$ debidas a momentos unitarios, con las cuales se calculan las excentricidades e_x y e_y entre centros de masa y de rigidez, y finalmente se calculan las coordenadas x_{cr} y y_{cr} de cada diafragma. Se crea una tabla temporal con estos 4 parámetros la cual se adjunta al final de la tabla *DRX()* inicial.

7.1.3. AEDCON

La subrutina *AEDCON* crea el subvector de desplazamientos conocidos de la estructura *DBB()*, que se utiliza para conformar el vector o la matriz de desplazamientos en coordenadas globales de los nudos de la estructura, requerido en cualquiera de los métodos de análi-

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

sis dinámico. La sintaxis de la subrutina y los parámetros que intervienen se describen en la Tabla 7.4.

AEDCON NGLC, MGL(), MDC(), DBB()

Variable	Tipo	Descripción
NGLC	Entero	Número de grados de libertad conocidos
MGL()	Entero	Tabla de numeración de grados de libertad por nudo
MDC()	Entero	Tabla de desplazamientos conocidos
DBB()	Real	Subvector de desplazamientos conocidos $\{u_\beta\}$, creado

Tabla 7.4. Sintaxis y argumentos de la instrucción AEDCON

Inicialmente con la subrutina *ORFUGL* se organizan los datos de la matriz *MDC()* en un vector columna siguiendo la numeración de los grados de libertad almacenada en *MGL()*, para luego, con la subrutina *MTSUMB* extraer del vector creado el subvector *DBB()* cuyo tamaño es igual al número de grados de libertad conocidos *NGLC*.

7.1.4. AEMPRO

La subrutina *AEMPRO* crea la tabla auxiliar de propiedades de las rótulas de la estructura *PRO()*, utilizada en el método de análisis dinámico inelástico. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.5.

AEMPRO ROTE, ARE(), NRO(), ROT(), PRO()

Variable	Tipo	Descripción
NGLC	Entero	Número de rótulas de la estructura
ARE()	Entero	Tabla de asignación de rótulas tipo a elementos
NRO()	Entero	Tabla de numeración de rótulas
ROT()	Real	Tabla de propiedades de rótulas tipo
PRO()	Real	Tabla de propiedades de las rótulas de la estructura, creada

Tabla 7.5. Sintaxis y argumentos de la instrucción AEMPRO

Inicialmente se leen los tamaños de la tablas *ROT()* y *NRO()* y se inicializa la matriz *PRO()* llena de ceros. Luego se llena la matriz *PRO()* con las propiedades almacenadas en *PRO()* teniendo en cuenta el orden en que se numeraron las rótulas en *NRO()* y el tipo de cada rótula, almacenado en *ARE()*.

7.1.5. AEKGLT

La subrutina *AEKGLT* crea la matriz de rigidez general de la estructura *KGL()*, que considera todos los grados de libertad estáticos. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.6.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

AEKGLT NELE, NGLT, ELE(), INC(), PAU(), RIG(), XYZ(), PAE(), KGL()

Variable	Tipo	Descripción
NELE	Entero	Número de elementos
NGLT	Entero	Número de grados de libertad total
ELE()	Entero	Tabla de conectividades entre elementos
INC()	Entero	Tabla de numeración de grados de libertad por elemento (*incidencias*)
PAU()	Entero	Tabla de puntos auxiliares para definir ejes locales de elementos
RIG()	Real	Tabla de tipo de rigidez asociada a cada elemento
XYZ()	Real	Tabla de coordenadas de nudos y de puntos auxiliares
PAE()	Real	Tabla de propiedades de elementos
KGL()	Real	Matriz e rigidez general de la estructura $[k]$, creada

Tabla 7.6. Sintaxis y argumentos de la instrucción AEKGLT

En esta subrutina se inicializa la matriz *KGL()* cuadrada llena de ceros y de tamaño igual a *NGLT*, luego para cada elemento se crea la matriz de rigidez en coordenadas globales *KEG()* utilizando la subrutina *KPOR3D*, la cual se ensambla en *KGL()* con la subrutina *ENSAMK*. La subrutina *KPOR3D* es una de las más importantes del programa y se describe en detalle más adelante.

7.1.6. AEKGLI

La subrutina *AEKGLI* crea la matriz de rigidez condensada a los grados de libertad de interés para el análisis dinámico *KDI()*, adicionalmente y entrega las submatrices *KAA()*, *KAG()*, *KGA()* y *KGG()*, esta última invertida. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.7.

AEKGLT NGLI, NGLO, KGL(), KAA(), KAG(), KGA(), KGG(), KDI()

Variable	Tipo	Descripción
NGLI	Entero	Número de grados de libertad de interés
NGLO	Entero	Número de grados de libertad condensados
KGL()	Real	Matriz de rigidez general $[k]$
KAA()	Real	Submatriz $[k_{\alpha\alpha}]$ de $[k]$
KAG()	Real	Submatriz $[k_{\alpha\gamma}]$ de $[k]$
KGA()	Real	Submatriz $[k_{\gamma\alpha}]$ de $[k]$
KGG()	Real	Submatriz $[k_{\gamma\gamma}]^{-1}$ de $[k]$ (*invertida*)
KDI()	Real	Matriz e rigidez general de la estructura $[\tilde{k}_{\alpha\alpha}]$, creada

Tabla 7.7. Sintaxis y argumentos de la instrucción AEKGLI

A partir de *KGL()* se crean las submatrices *KAA()*, *KAG()*, *KGA()* y *KGG()* utilizando la subrutina *MTSUBM*, luego se invierte *KGG()* con *MTINVE* sobrescribiéndola y finalmente calcula la matriz de rigidez condensada *KDI()* realizando las operaciones matriciales requeridas con las subrutinas *MTMULT* y *MTREST*.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

7.1.7. AEDESP

La subrutina *AEDESP* calcula los desplazamientos en coordenadas globales de los nudos *DNG()*, a partir de los desplazamientos de los centros de masa, de la matriz de asociación de desplazamientos, de los desplazamientos conocidos y de las submatrices de la matriz de rigidez general. Se crea una matriz de igual número de columnas al de la matriz de desplazamientos de centros de masa. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.8.

```
AEDESP NGLD, NGLC, TDR(), DCM(), KGG(), KGA(), DBB(), DNG()
```

Variable	Tipo	Descripción
NGLD	Entero	Número de grados de libertad desconocidos
NGLC	Entero	Número de grados de libertad conocidos
TDR()	Real	Matriz de asociación de desplazamientos $[A]$
DCM()	Real	Matriz (*o vector*) de desplazamientos de centros de masa $[u_\phi]$
KGG()	Real	Submatriz $[k_{\gamma\gamma}]^{-1}$ de $[k]$
KGA()	Real	Submatriz $[k_{\gamma\alpha}]$ de $[k]$
DBB()	Real	Vector de desplazamientos conocidos $\{u_\beta\}$
DNG()	Real	Matriz (*o vector*) de desplazamientos de nudos $[u]$, creada

Tabla 7.8. Sintaxis y argumentos de la instrucción AEDESP

Inicialmente se lee el tamaño de la matriz *DCM()*, luego se calculan los desplazamientos de interés mediante la expresión $[u_\alpha] = [A][u_\phi]$, se calculan los desplazamientos condensados mediante la expresión $[u_\gamma] = -[k_{\gamma\gamma}]^{-1}[k_{\gamma\alpha}][u_\alpha]$ y el vector $\{u_\beta\}$ se expande a la matriz $[u_\beta]$ con un número de columnas igual al de la matriz $[u_\phi]$. Se utilizan las subrutinas *MTPORE* y *MTMULT* para realizar las operaciones matriciales requeridas y se utiliza la subrutina *MTADMR* para crear la matriz $[u]$ adjuntando las submatrices $[u_\alpha]$, $[u_\gamma]$ y $[u_\beta]$, la cual finalmente se almacena en *DNG()*.

Cuando se utiliza en el método de análisis dinámico inelástico, la subrutina crea un vector columna con el incremento estimado en los desplazamientos de los nudos $\{\Delta u^{(j)}\}$ a partir del incremento de desplazamientos de centros de masa $\{\Delta u_\phi^{(j)}\}$, y utiliza las submatrices $[k_{\gamma\gamma}(t)]^{-1}$ y $[k_{\gamma\alpha}(t)]$ de la matriz de rigidez tangente evaluada al comienzo del intervalo.

7.1.8. AEDECM

La subrutina *AEDECM* calcula las derivas de los centros de masa en direcciones x e y como fracción de la altura de piso. En las tablas *ECX()* y *ECY()* creadas cada columna corresponde a un instante de tiempo o a un modo de vibración, dependiendo del método de análisis dinámico utilizado. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.9.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

AEDECM NDIR, DCM(), DRX(), ECX(), ECY()

Variable	Tipo	Descripción
NDIR	Entero	Número de diafragmas rígidos
DCM()	Real	Matriz de desplazamientos de centros de masa $[u_\phi]$
DRX()	Real	Tabla de coordenadas de centros de masa
ECX()	Real	Tabla de derivas de centros de masa en dirección X global $[\Delta^x]$, creada
ECY()	Real	Tabla de derivas de centros de masa en dirección Y global $[\Delta^y]$, creada

Tabla 7.9. Sintaxis y argumentos de la instrucción AEDECM

En la subrutina inicialmente se lee el tamaño de la matriz *DCM()* y se inicializan las matrices *ECX()* y *ECY()* llenas de ceros. Luego, para cada diafragma, se define su altura de piso y se calculan los desplazamientos relativos respecto al centro de masa inmediatamente inferior de cada centro de masa en cada dirección x e y, dependiendo si el pórtico analizado es plano o espacial. Finalmente los desplazamientos relativos de cada diafragma se dividen en la altura de piso respectivo.

7.1.9. AEDERI

La subrutina *AEDERI* calcula las derivas de un eje de columnas a partir de los desplazamientos en coordenadas globales de los nudos que pertenecen al eje. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.10.

AEDERI IEJE, DNG(), XYZ(), MGL(), NEC(), NNE(), DER()

Variable	Tipo	Descripción
IEJE	Entero	Eje de columnas actual (*en estudio*)
DNG()	Real	Matriz de desplazamientos en coordenadas globales de los nudos $[u]$
XYZ()	Real	Tabla de coordenadas de nudos y puntos auxiliares
MGL()	Entero	Tabla de numeración de grados de libertad por nudo
NEC()	Entero	Tabla de nudos asociados a cada eje de columnas
NNE()	Entero	Tabla de cantidad de nudos asociados a cada eje de columnas
DER()	Real	Tabla de derivas del eje de columnas actual $[\Delta]$, creada

Tabla 7.10. Sintaxis y argumentos de la instrucción AEDERI

Inicialmente se lee el tamaño de la matriz *DNG()* a partir del cual se inicializa la matriz *DER()* llena de ceros. Luego, para cada instante de tiempo o para cada modo de vibración, se extraen y organizan los desplazamientos de los nudos utilizando las subrutinas *MTSUBM* y *ORGLFU*, y para cada nudo del eje, exceptuando el nudo de la base, se calculan los desplazamientos relativos en direcciones x e y, la altura de piso correspondiente, y la deriva total del nudo como fracción de la altura de piso.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

7.1.10. AECORP

La subrutina *AECORP* calcula los cortantes de piso en direcciones x e y, en cada instante de tiempo o por cada modo de vibración natural, a partir de las acciones equivalentes trasladadas a los centros de masa. Las tablas creadas almacenan los cortantes de piso ordenados por niveles, de acuerdo a como fueron ordenados los diafragmas en la respectiva tabla de datos. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.11.

AECORP `FEQ(), NDIM, VPX(), VPY()`

Variable	Tipo	Descripción
`FEQ()`	Real	Matriz de acciones equivalentes trasladadas a centros de masa $[(F_s)_\phi]$
`NDIM`	Entero	Número de dimensiones de la estructura
`VPX()`	Real	Tabla de cortantes de piso en dirección X $[V^x]$, creada
`VPY()`	Real	Tabla de cortantes de piso en dirección X $[V^y]$, creada

Tabla 7.11. Sintaxis y argumentos de la instrucción AECORP

Inicialmente se lee el tamaño de la matriz *FEQ()* para conocer el número de columnas e inicializar llenas de cero las matrices *VPX()* y *VPY()* mediante las subrutinas *PBDEMR* y *MTCONS*. Luego se calculan los cortantes de piso en cada dirección a partir de los datos de la matriz *FEQ()*, teniendo presente si el pórtico analizado es plano o espacial.

7.1.11. AEDVPI

La subrutina *AEDVPI* crea una tabla de datos que permite graficar la variación en altura de los cortantes de piso en una de las direcciones x o y, en cada instante de tiempo o correspondientes a cada modo de vibración natural, a partir de la tabla correspondiente de cortantes de piso. La primera columna de la matriz *MXX()* creada contiene las alturas de cada piso medidas desde la base hasta cada sección en la que se grafica el cortante de piso. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.12.

AEDVPI `DRX(), MXX(), DXX()`

Variable	Tipo	Descripción
`DRX()`	Real	Tabla de coordenadas de centros de masa
`MXX()`	Real	Tabla de cortantes de piso en dirección X o en dirección Y
`DXX()`	Real	Tabla de datos para graficar los diagramas de cortante en cada instante de tiempo o por cada modo de vibración, creada

Tabla 7.12. Sintaxis y argumentos de la instrucción AEDVPI

Respecto al funcionamiento de la subrutina, inicialmente se lee el tamaño de la matriz *MXX()* y se inicializa la matriz *DXX()* llena de ceros. Posteriormente, para cada sección en la que se evalúa el cortante de piso, se calcula la distancia hasta la base y el cortante correspondiente, datos que se almacenan en la matriz *DXX()* final.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

7.1.12. AENVOL

La subrutina *AENVOL* crea y adjunta las envolventes de valores máximos positivos y negativos, de los coeficientes de una tabla de datos en la que cada fila corresponde a un parámetro diferente (*desplazamientos, derivas, acciones internas, etc.*) y la primera columna almacena datos auxiliares tales como distancias.

A la tabla de datos base *MXX()* se adicionan dos columnas, la primera con la envolvente máxima positiva y la segunda con la envolvente máxima negativa. Ejemplos de estas tablas son las usadas para graficar diagramas de acciones internas o de cortantes de piso. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.13.

AENVOL `MXX()`

Variable	Tipo	Descripción
`MXX()`	Real	Tabla de datos con envolventes adjuntas al final, modificada

Tabla 7.13. Sintaxis y argumentos de la instrucción AENVOL

Dentro de la subrutina se lee el tamaño de la matriz *MXX()* y se inicializa la matriz auxiliar *MXE()* llena de ceros, la cual almacena las envolventes. Posteriormente para cada fila de *MXX()* se obtienen los valores máximos positivo y negativo sin tener en cuenta el dato de la primera celda, para lo cual se usa la función *MTMACR*. Finalmente se adjuntan las matrices *MXX()* y *MXE()* sobrescribiendo la matriz *MXX()* original.

7.1.13. AEFELE

La subrutina *AEFELE* calcula la matriz *FEL()* de acciones en coordenadas locales en los extremos de un elemento estructural, a partir de la matriz de rigidez, el vector de acciones fijas y la matriz de desplazamientos de los nudos del elemento en coordenadas locales. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.14.

AEFELE `IELE, INC(), ELE(), PAU(), RIG(), XYZ(), PAE(), DNG(), RFF(), FEL()`

Variable	Tipo	Descripción
`IELE`	Entero	Elemento actual (*en estudio*)
`INC()`	Entero	Matriz de numeración de grados de libertad por elemento (*incidencias*)
`ELE()`	Entero	Tabla de conectividades de elementos
`PAU()`	Entero	Tabla de puntos auxiliares para definir ejes locales de elementos
`RIG()`	Entero	Tabla de rigidez tipo asociada a cada elemento
`XYZ()`	Real	Tabla de coordenadas de nudos y de puntos auxiliares
`PAE()`	Real	Tabla de propiedades de elementos
`DNG()`	Real	Matriz de desplazamientos en coordenadas locales de nudos $[u]$
`RFF()`	Real	Tabla resumen de acciones fijas en coordenadas locales en los elementos, producidas por las cargas iniciales (*se usa solo en análisis dinámico inelástico*)
`FEL()`	Real	Matriz de acciones en c. locales en los extremos del elemento $[\bar{f}_e]$, creada

Tabla 7.14. Sintaxis y argumentos de la instrucción AEFELE

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

En la subrutina se lee el tamaño de las matrices *RFF() y DNG()*, se construye un vector auxiliar lleno de unos con el que posteriormente se expande el vector de acciones fijas en coordenadas locales del elemento $\{\bar{f}_e^e\}$ a la matriz $[\bar{f}_e^e]$, en la que cada columna corresponde a un instante de tiempo o modo de vibración. Con la subrutina *KPOR3D* se crea la matriz de rigidez elemental en coordenadas locales, con la subrutina *TRPO3D* se crea la matriz de transformación de coordenadas y con la subrutina *EXTRAV* se extrae de *DNG()* la matriz de los desplazamientos de los nudos del elemento. Finalmente las subrutinas *MTMULT* y *MTSUMA* realizan las operaciones matriciales requeridas para crear la matriz *FEL()*.

Cuando esta subrutina se utiliza en alguno de los métodos de análisis dinámico elástico la matriz *RFF()* debe estar llena de ceros, puesto que en estos casos el efecto de las cargas iniciales distribuidas en la luz del elemento se tiene en cuenta en otro caso de carga. Cuando se usa el método de análisis dinámico inelástico, *RFF()* puede contener valores diferentes de cero, y en ese caso *RFF()* solo se tiene en cuenta en el instante de tiempo correspondiente a $t = 0$.

7.1.14. AETRFE

La subrutina *AETRFE* crea una tabla resumen de acciones máximas en coordenadas locales en los extremos de los elementos. Cuando las acciones se calculan por análisis modal cronológico o análisis dinámico inelástico, en la tabla se almacenan las acciones máximas absolutas junto con los instantes de tiempo donde se presentan los respectivos valores máximos, mientras que cuando se usa el método de análisis modal espectral se almacenan las acciones máximas probables $\{(\bar{f}_e)_o\}$ del elemento en cada fila de la tabla resumen.

En el primer caso la tabla tiene 24 columnas mientras que en el segundo caso el número de columnas es 12. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.15.

AETRFE `IDMA, IELE, FEL(), TME(), FEM()`

Variable	Tipo	Descripción
`IDMA`	Cadena	Método de análisis dinámico utilizado
`IELE`	Entero	Elemento actual (*en estudio*)
`FEL()`	Real	Matriz de acciones en coordenadas locales en los extremos del elemento $[\bar{f}_e]$
`TME()`	Real	Vector de instantes de tiempo considerados en el análisis
`FEM()`	Real	Tabla resumen de acciones en los extremos de los elementos, modificada

Tabla 7.15. Sintaxis y argumentos de la instrucción AETRFE

En la subrutina inicialmente se lee el tamaño de la matriz *FEM()*, la cual ha sido inicializada previamente fuera de la subrutina. Posteriormente *FEM()* se redimensiona con la instrucción *Redim Preserve* manteniendo el tamaño y el contenido original.

Si se utiliza el método de análisis modal cronológico o el método de análisis dinámico inelástico, con la subrutina *ADVMAX* se crea una tabla de dos columnas, la primera colum-

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

na con los valores máximos absolutos de las acciones en los extremos, y la segunda columna con los instantes de tiempo correspondientes. Esta tabla se transpone y las dos filas resultantes se colocan una al lado de la otra con las subrutinas *MTSUBM* y *MTADMR*. Finalmente el vector fila resultante se reemplaza en la tabla resumen *FEM()* con la subrutina *MTREMV*.

Si se utiliza el método de análisis modal espectral las acciones máximas del elemento se toman de la última columna de la matriz *FEL()*, dado que esta tabla almacena el vector $\{(\bar{f}_e)_o\}$. Se utilizan las subrutinas *MTSUBM* y *MTTRAN* para extraer y transponer la última columna de *FEL()*, la cual se reemplaza en *FEM()* con la subrutina *MTREMV*.

7.1.15. AETRDE

La subrutina *AETRDE* crea una tabla resumen con los desplazamientos máximos en coordenadas globales de los nudos. Cuando se utilizan los métodos de análisis modal cronológico o análisis dinámico inelástico, se obtienen y almacenan en una misma fila los desplazamientos máximos absolutos de cada nudo y los instantes de tiempo donde estos se producen, mientras que si se utiliza el método de análisis modal espectral, se almacenan en la misma fila los desplazamientos máximos probables de cada nudo producidos por la acción simultánea de todas las componentes de aceleración de la base consideradas en el análisis.

En el primer caso el número de columnas de la tabla resumen es 12 mientras que en el segundo el número de columnas es 6. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.16.

AETRDE `IDMA, MGL(), DNG(), TME(), DEM()`

Variable	Tipo	Descripción
`IDMA`	Cadena	Método de análisis dinámico utilizado
`MGL()`	Entero	Tabla de numeración de grados de libertad por nudo
`DNG()`	Real	Matriz de desplazamientos en coordenadas globales de los nudos $[u]$
`TME()`	Real	Vector de instantes de tiempo considerados en el análisis
`DEM()`	Real	Tabla resumen de desplazamientos, modificada

Tabla 7.16. Sintaxis y argumentos de la instrucción AETRDE

Dentro de la subrutina se lee el tamaño de la matriz *DNG()* con la subrutina *PBDEMR*. Posteriormente, si se usa análisis modal cronológico o análisis dinámico inelástico, con la subrutina *ADVMAX* se crea a partir de *DNG()* una tabla de dos columnas, la primera con los desplazamientos máximos absolutos de los nudos y la segunda con los instantes de tiempo correspondientes. Estas dos tablas se organizan con la subrutina *ORGLFU* de forma que, en cada fila de cada tabla, se almacenan respectivamente los desplazamientos máximos y los tiempos correspondientes. Finalmente estas dos tablas se adjuntan con *MTADMR* creando la tabla *DEM()*.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

Si se utiliza análisis modal espectral, con la subrutina *MTSUBM* se extrae la última columna de la matriz *DNG()*, y esta se organiza en el mismo formato indicado para los otros casos de análisis con la subrutina *ORGLFU*, dando como resultado la tabla *DEM()*.

7.1.16. AEFFES

La subrutina *AEFFES* crea el vector de acciones fijas de la estructura, producidas por las cargas iniciales distribuidas en la luz de los elementos. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.17.

```
AEFFES NELE, NGLT, INC(), ELE(), XYZ(), PAE(), TWE(), WEL(), PAU(),
                                                  RIG(), FFE(), RFF()
```

Variable	Tipo	Descripción
NELE	Entero	Número de elementos
NGLT	Entero	Número total de grados de libertad
INC()	Entero	Tabla de numeración de grados de libertad por elemento (*incidencias*)
ELE()	Entero	Tabla de conectividades entre elementos
XYZ()	Real	Tabla de coordenadas de nudos y de puntos auxiliares
PAE()	Real	Tabla de propiedades de elementos
TWE()	Entero	Tabla de tipos de carga distribuida aplicada a elementos
WEL()	Real	Tabla de valores de carga distribuida aplicada a elementos
PAU()	Entero	Tabla de puntos auxiliares para definición de ejes locales
RIG()	Entero	Tabla de rigidez tipo por elemento
FFE()	Real	Vector de acciones fijas en coordenadas globales, creado
RFF()	Real	Tabla resumen de acciones fijas en coordenadas locales por elemento, creada

Tabla 7.17. Sintaxis y argumentos de la instrucción AEFFES

Dentro de la subrutina se inicializa el vector columna *FFE()* y la tabla *RFF()* llenas de ceros con la subrutina *MTCONS*. Posteriormente, para cada elemento de la estructura, se crea el vector de acciones fijas en coordenadas locales *FEG()* con la subrutina *FFPO3D*, una de las subrutinas más extensas del programa, y se crea la matriz de transformación de coordenadas *TEL()* con la subrutina *TRPO3D*.

Las acciones en coordenadas globales almacenadas en *FEG()* se ensamblan en el vector *FFE()* con la subrutina *ENSAMV*, se transforman a coordenadas locales con la subrutina *MTMULT* usando la matriz *TEL()* y se almacenan en la tabla resumen *RFF()* con la subrutina *MTREMV* reemplazando la columna de datos correspondiente al elemento.

7.1.17. AEDESI

La subrutina *AEDESI* calcula los desplazamientos de los centros de masa, los desplazamientos en coordenadas globales de los nudos y las acciones equivalentes trasladadas a los

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

centros de masa, producidos por las cargas iniciales distribuidas en los elementos. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.18.

```
AEDESI NGLI, NGLO, KDP(), FNE(), FFE(), KAG(), KGA(), KGG(), TDR(),
                                    TDT(), DBB(), DCM(), DNG(), FEQ()
```

Variable	Tipo	Descripción
NGLI	Entero	Número de grados de libertad de interés
NGLO	Entero	Número de grados de libertad condensados
KDP()	Real	Matriz de rigidez lateral $[k_\phi]$
FNE()	Real	Vector de acciones externas aplicadas en los nudos $[F]$
FFE()	Real	Vector de acciones fijas en coordenadas globales $[F^F]$
KAG()	Real	Submatriz $[k_{\alpha\gamma}]$ de $[k]$
KGA()	Real	Submatriz $[k_{\gamma\alpha}]$ de $[k]$
KGG()	Real	Submatriz $[k_{\gamma\gamma}]^{-1}$ de $[k]$
TDR()	Real	Matriz de asociación de desplazamientos $[A]$
TDT()	Real	Matriz de asociación de desplazamientos transpuesta $[A]^T$
DBB()	Real	Subvector de desplazamientos conocidos $\{u_\beta\}$
DCM()	Real	Vector de desplazamientos iniciales de centros de masa $\{u_\phi\}$, creado
DNG()	Real	Vector de desplazamientos iniciales de nudos $\{u\}$, creado
FEQ()	Real	Vector de acciones equivalentes iniciales $\{(F_s)_\phi\}$, creado

Tabla 7.18. Sintaxis y argumentos de la instrucción AEDESI

La subrutina sigue la formulación matemática presentada en el Capítulo 5. Inicialmente se crean los subvectores $\{F_\alpha^F\}$ y $\{F_\gamma^F\}$ a partir del vector $\{F^F\}$ utilizando la subrutina *MTSUBM*. Se calcula el vector $\{\tilde{F}_\alpha^{(F)}\} = \{F_\alpha^{(F)}\} - [K_{\propto\gamma}][K_{\gamma\gamma}]^{-1}\{F_\gamma^{(F)}\}$ realizando las operaciones matriciales requeridas con las subrutinas *MTMULT* y *MTREST*. Finalmente se calcula el vector $\{(F_s)_\phi\} = \{F_\phi\} = [A]^T\{F_\alpha\}$ con las subrutinas *MTMULT*, donde $\{F_\alpha\}$ previamente ha sido extraído de $\{F\}$ con la subrutina *MTSUBM*. El resultado se almacena en el vector columna *FEQ()*.

Las operaciones matriciales que implican el cálculo de los desplazamientos iniciales de los centros de masa $\{u_\phi\} = [K_\phi]^{-1}(\{F_\phi\} - \{\tilde{F}_\phi^{(F)}\})$ con $\{\tilde{F}_\phi^{(F)}\} = [A]^T\{\tilde{F}_\alpha^{(F)}\}$ se realizan con las instrucciones *MTINVE*, *MTMULT* y *MTREST*. El resultado se almacena en el vector columna *DCM()*.

Finalmente el vector de desplazamientos de los nudos $\{u\}$ se construye a partir de los subvectores $\{u_\alpha\}$, $\{u_\gamma\}$ y $\{u_\beta\}$. El primero se calcula como $\{u_\alpha\} = [A]\{u_\phi\}$ y el segundo como $\{u_\gamma\} = -[K_{\gamma\gamma}]^{-1}(\{F_\gamma^{(F)}\} + [K_{\gamma\propto}]\{u_\alpha\})$ y el tercero es dato de entrada. Las operaciones matriciales requeridas se realizan con las subrutinas *MTSUMA* y *MTMULT*, y el resultado se almacena en el vector columna *DNG()*.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

7.1.18. AEINIF

La subrutina *AEINIF* llena la primera columna de la tabla de acciones en coordenadas locales en los extremos de los elementos *FBL()*, necesaria cuanto se utiliza el método de análisis dinámico inelástico, con las acciones obtenidas del análisis para cargas iniciales, almacenadas en la tabla resumen *RFI()*. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.19.

AEINIF NELE, NGLE, RFI(), FBL()

Variable	Tipo	Descripción
NELE	Entero	Número de elementos
NGLE	Entero	Número de grados de libertad por elemento
RFI()	Real	Tabla resumen de acciones en los extremos de los elementos producidas por las cargas iniciales
FBL()	Real	Tabla de acciones en los extremos de los elementos en cada instante de tiempo, inicializada

Tabla 7.19. Sintaxis y argumentos de la instrucción AEINIF

Esta subrutina toma los datos de cada columna de la matriz *RFI()* y los almacena en las celdas correspondientes de la primera columna de la matriz *FBL()*. Para ello utiliza las subrutinas *MTSUBM* y *MTREMP*. Respecto a las celdas correspondientes a cada elemento, al primer elemento le corresponden las primeras 12 celdas, al segundo elemento las siguientes 12 celdas, etc.

Con esta subrutina solamente se llena la primera columna de la matriz *FBL()*. Las columnas restantes se llenan con los resultados del procedimiento principal del método de análisis dinámico inelástico, como se describe más adelante.

7.1.19. AEINIR

La subrutina *AEINIR* llena la primera columna de las matrices de resultados de las rótulas plásticas o resortes rotacionales, específicamente de las matrices *ERO()*, *DRO()*, *FRO()* y *KRO()*, con los resultados del análisis de la estructura para las cargas iniciales.

Las fuerzas iniciales en los resortes se toman directamente de las acciones en los extremos de los elementos, la rigidez inicial de los resortes se mantiene y las rotaciones y los parámetros elásticos de deformación se calculan a partir de los momentos y las rigideces iniciales en $t = 0$. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.20.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

AEINIR NRO(), PRO(), RFI(), KRO(), DRO(), FRO(), ERO()

Variable	Tipo	Descripción
NRO()	Entero	Tabla de numeración de rótulas
PRO()	Real	Tabla de propiedades de rótulas de la estructura
RFI()	Real	Tabla resumen de acciones en los extremos de los elementos producidas por las cargas iniciales
KRO()	real	Tabla de rigidez tangente de las rótulas plásticas, inicializada
DRO()	Real	Tabla de desplazamientos (*o rotaciones*) de rótulas, inicializada
FRO()	Real	Tabla de fuerzas (*o momentos*) de rótulas, inicializada
ERO()	Real	Tabla de parámetros de deformación elástica de las rótulas, inicializada

Tabla 7.20. Sintaxis y argumentos de la instrucción AEINIR

Inicialmente se leen los tamaños de las matrices *NRO() y DRO()* para conocer el número de elementos y el número de resortes rotacionales de la estructura. La primera columna de *KRO()* se llena con la rigidez inicial de los resortes, la cual se toma de la primera columna de la matriz *PRO()* con las subrutinas *MTSUBM* y *MTREMV*. La primera columna de *FRO()* se llena localizando el elemento y el grado de libertad al cual está asociada cada rótula plástica y tomando el momento correspondiente de la tabla resumen *RFI()*. La primera columna de *DRO()* se calcula con la expresión $\theta_i = M_i/J_i$ en la que se supone linealidad entre fuerzas y desplazamientos, utilizando los coeficientes de la primera columna de *FRO()* y de *KRO()*. Finalmente el parámetro de rotación elástica e de cada resorte rotacional se despeja de la relación $M = J_y\theta + (J_e - J_y)e$, con lo que se completa la primera columna de *ERO()*.

7.1.20. AEDIVC

La subrutina *AEDIVC* se utiliza para crear las tablas de datos que almacenan la información necesaria y ordenada para graficar los diagramas de acciones internas $N_{\bar{x}}$, $V_{\bar{y}}$, $V_{\bar{z}}$, $T_{\bar{x}}$, $M_{\bar{y}}$ y $M_{\bar{z}}$ en cada instante de tiempo, si se usa análisis modal cronológico o análisis dinámico inelástico, o para cada modo de vibración si se usa análisis modal espectral, para un conjunto de elementos estructurales conectados formando un eje de vigas (*horizontal*) o de columnas (*vertical*).

Cuando se usa análisis dinámico inelástico se tienen en cuenta las cargas iniciales aplicadas en la luz de los elementos para calcular las acciones internas en cada sección y en cada instante de tiempo. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.21.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

AEDIVC IDMA, IEJE, NAUX, EJE(), ELE(), CTS(), XYZ(), TWE(), WEL(), NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

Variable	Tipo	Descripción
IDMA	Cadena	Método de análisis dinámico utilizado
IEJE	Entero	Eje de vigas o de columnas actual
NAUX	Entero	Parámetro auxiliar que define la cantidad de columnas de las matrices de acciones en los extremos de los elementos a importar
EJE()	Entero	Tabla de elementos asociados a cada eje de vigas o columnas
ELE()	Entero	Tabla de conectividades
CTS()	Real	Tabla de número de secciones por elemento
XYZ()	Real	Tabla de coordenadas de nudos y de puntos auxiliares
TWE()	Entero	Tabla de tipos de carga inicial distribuida en la luz de los elementos
WEL()	Real	Tabla de valores de carga inicial distribuida en la luz de los elementos
NXX()	Real	Tabla de diagramas de fuerza axial, creada
VYY()	Real	Tabla de diagramas de fuerza cortante en dirección *y* local, creada
VZZ()	Real	Tabla de diagramas de fuerza cortante en dirección *z* local, creada
TXX()	Real	Tabla de diagramas de momento torsor, creada
MYY()	Real	Tabla de diagramas de momento flector en dirección *y* local, creada
MZZ()	Real	Tabla de diagramas de momento flector en dirección *z* local, creada

Tabla 7.21. Sintaxis y argumentos de la instrucción AEDIVC

Internamente la subrutina define el número de filas de cada una de las tablas de acciones internas en función del número de elementos que pertenecen al eje, y del número de secciones a tomar en cada elemento, información almacenada en las tablas *EJE()* y *CTS()*. Luego se define el número de columnas de cada matriz de acciones en los extremos de los elementos del eje, dependiendo del método de análisis dinámico escogido. Con la subrutina *MTCONS* se inicializa un vector columna denominado *SEC()* que almacena temporalmente la distancia desde el extremo inicial del primer elemento del eje hasta cada sección definida para evaluar las acciones internas. Se inicializan las seis tablas de datos donde se almacenarán temporalmente las acciones internas calculadas.

Posteriormente para cada elemento que pertenece al eje, con la subrutina *IOIMPR* se importa la matriz de acciones en coordenadas locales en los extremos, con la función *PBDIST* se calcula la longitud del elemento y se llena la parte del vector *SEC()* con las secciones del elemento. Luego para cada sección del mismo elemento se calculan las acciones internas debidas solamente a las cargas distribuidas iniciales con la subrutina *AEAICD*. Estas se calculan solo si se utiliza el método de análisis dinámico inelástico, en caso contrario se toman como cero. Con el aporte de las cargas distribuidas a las acciones en cada sección y las acciones en el extremo inicial del elemento, se calculan las acciones internas totales en cada sección, para cada instante de tiempo o para cada modo de vibración, las cuales se almacenan en las seis tablas temporales que se habían inicializado previamente. Al final a cada tabla temporal se adjunta el vector columna de secciones con la subrutina *MTADMR*, formando las seis tablas *NXX()*, *VYY()*, *VZZ()*, *TXX()*, *MYY()* y *MZZ()*.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

Respecto a las tablas de acciones en los extremos de los elementos que se importan, cuando se utiliza el método de análisis modal espectral se importa la tabla resumen de cada elemento, en la que se almacena todo el cálculo de las acciones máximas probables debidas a todas las componentes de aceleración de la base actuando simultáneamente. Sin embargo, en el cálculo de las acciones internas en cada sección se ignoran las tres últimas columnas, puesto que estos valores ya están combinados con métodos de combinación modal y direccional y por lo tanto son valores estimados que no cumplen con el equilibrio del elemento. Cuando se utiliza alguno de los otros métodos de análisis dinámico, se importa toda la tabla resumen de acciones en los extremos de cada elemento, de la cual se ignora la primera columna puesto que esta corresponde a los instantes de tiempo. Finalmente la parte restante de la tabla debe transponerse, para lo cual se usa la subrutina *MTTRAN*.

7.1.21. AEAICD

La subrutina *AEAIDC* calcula el aporte de las cargas iniciales distribuidas en la luz de un elemento a las seis acciones internas en una sección determinada del mismo elemento.

Es una subrutina interna que utiliza en la subrutina *AEDIVC* previamente mencionada, cuando se utiliza el método de análisis dinámico inelástico. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.22.

```
AEAICD IELE, LONE, XSEC, TWE(), WEL(), NXXI, VYYI, VZZI, TXXI, MYYI,
                                                                MZZI
```

Variable	Tipo	Descripción
IELE	Entero	Elemento actual
LONE	Real	Longitud del elemento actual
XSEC	Real	Distancia desde el extremo inicial del elemento hasta la sección actual de este
TWE()	Entero	Tabla de tipos de carga inicial distribuida en la luz de los elementos
WEL()	Real	Tabla de valores de carga inicial distribuida en la luz de los elementos
NXXI	Real	Fuerza axial en la sección, calculada
VYYI	Real	Fuerza cortante en dirección *y* local en la sección, calculada
VZZI	Real	Fuerza cortante en dirección *z* local en la sección, calculada
TXXI	Real	Momento torsor en la sección, calculado
MYYI	Real	Momento flector en dirección *y* local en la sección, calculado
MZZI	Real	Momento flector en dirección *z* local en la sección, calculado

Tabla 7.22. Sintaxis y argumentos de la instrucción AEAICD

Internamente en la subrutina se inicializan en cero las seis variables *NXXI*, *VYYI*, *VZZI*, *TXXI*, *MYYI* y *MZZI*. Luego, para cada carga inicial aplicada al elemento, se identifica el tipo de carga y su valor numérico con base en los datos de las tablas *TWE()* y *WEL()*, y dependiendo del tipo de carga distribuida (*uniforme o triangular*) o de la dirección de aplicación (*y o z local*) se calculan los aportes a las acciones internas de la sección buscados. El aporte de cada tipo de carga aplicada al elemento a cada acción interna en la sección se

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

acumula, de forma que al final se tengan los aportes totales de todas las cargas aplicadas al mismo elemento.

7.1.22. AEDIEN

La subrutina *AEDIEN* crea las envolventes de acciones internas máximas positivas y negativas almacenadas las tablas *NXXI*, *VYYI*, *VZZI*, *TXXI*, *MYYI* y *MZZI*, y las adjunta al final de cada tabla en dos columnas adicionales.

Esta subrutina aparece en el código principal cuando se usan los métodos de análisis modal cronológico o análisis dinámico inelástico. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.23.

AEDIEN `NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()`

Variable	Tipo	Descripción
`NXX()`	Real	Tabla de diagramas de fuerza axial, modificada
`VYY()`	Real	Tabla de diagramas de fuerza cortante en dirección *y* local, modificada
`VZZ()`	Real	Tabla de diagramas de fuerza cortante en dirección *z* local, modificada
`TXX()`	Real	Tabla de diagramas de momento torsor, modificada
`MYY()`	Real	Tabla de diagramas de momento flector en dirección *y* local, modificada
`MZZ()`	Real	Tabla de diagramas de momento flector en dirección *z* local, modificada

Tabla 7.23. Sintaxis y argumentos de la instrucción AEDIEN

Simplemente se utiliza la subrutina *AENVOL* para crear cada envolvente y adjuntarla al final de cada tabla de datos. La subrutina se construyó para reducir el número de líneas que implicaban el uso directo de la subrutina *AENVOL* en el código principal.

7.1.23. AEDFSJ

La subrutina *AEDFSJ* calcula el incremento en las acciones equivalentes trasladadas a los centros de masa $\{\Delta F_{s\phi}^{(j)}\}$, en el intervalo de tiempo actual y para una iteración j cualquiera, producidas por el incremento en los desplazamientos de los centros de masa $\{\Delta u_{\phi}^{(j)}\}$ en el mismo intervalo y en la misma iteración.

Como resultados adicionales la subrutina entrega el incremento en los desplazamientos en coordenadas globales de los nudos $\{\Delta u^{(j)}\}$ y el incremento de las acciones en coordenadas locales en los extremos de cada elemento estructural $\{\Delta \bar{f}_e^{(j)}\}$, así como los parámetros de las rótulas plásticas o resortes rotacionales (*momentos, rotaciones, rigidez tangente y parámetros elásticos de deformación*) al final del intervalo y de la iteración.

La subrutina se utiliza solamente en el caso de análisis dinámico inelástico. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.24.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
AEDFSJ ITIM, TOLD, ITEM, ERRM, NMIT, INC(), ELE(), PAU(), RIG(), TDT(),
  XYZ(), PAE(), NRO(), PRO(), KGG(), DDN(), ERO(), DRO(), FRO(), KRO(),
                                                          DFS(), FBL()
```

Variable	Tipo	Descripción
ITIM	Entero	Intervalo de tiempo actual
TOLD	Real	Tolerancia para convergencia de desplazamientos
ITEM	Real	Número máximo de iteraciones realizado en los ciclos internos realizados
ERRM	Real	Error al final de las iteraciones del ciclo interno
NMIT	Real	Número máximo de iteraciones permitido en cada ciclo interno
INC()	Entero	Tabla de incidencias
ELE()	Entero	Tabla de conectividades
PAU()	Entero	Tabla de puntos auxiliares para la definición de los ejes locales
RIG()	Entero	Tabla de rigidez tipo asignada a cada elemento
TDT()	Real	Matriz geométrica transpuesta $[A]^T$
XYZ()	Real	Tabla de coordenadas de nudos y de puntos auxiliares
PAE()	Real	Tabla de propiedades de elementos
NRO()	Entero	Tabla de numeración de rótulas
PRO()	Real	Tabla de propiedades de rótulas
KGG()	Real	Submatriz $[k_{\gamma\gamma(t)}]^{-1}$ de $[k_{(t)}]$
DDN()	Real	Vector de incremento de desplazamientos de nudos $\{\Delta u^{(j)}\}$, modificado
ERO()	Real	Tabla de parámetros de desplazamiento elástico de rótulas, modificada
DRO()	Real	Tabla de desplazamientos de rótulas, modificada
FRO()	Real	Tabla de fuerzas de rótulas, modificada
KRO()	Real	Tabla de rigidez tangente de rótulas, modificada
DFS()	Real	Vector de incremento de acciones equivalentes $\{(\Delta F_s)_\phi^{(j)}\}$, calculado
FBL()	Real	Tabla de acciones en los extremos de los elementos en cada instante de tiempo, creada

Tabla 7.24. Sintaxis y argumentos de la instrucción AEDFSJ

Internamente la subrutina calcula el número de elementos, de resortes rotacionales y de grados de libertad de interés, condensados y conocidos, leyendo el tamaño de las matrices *NRO(), KRO(), TDT(), KGG(), DDN()* y realizando algunas operaciones básicas adicionales. La tablas que almacena los incrementos en las acciones en los extremos de los elementos y los incrementos en las acciones equivalentes en los nudos, calculadas en cada iteración, se inicializan llenas de ceros. El incremento en los desplazamientos de los nudos, estimados fuera de la subrutina y almacenados en el vector *DDN()* que es dato de entrada, se toman como el primer incremento de desplazamientos de los nudos en el ciclo interno.

Posteriormente, para cada iteración del ciclo interno, se estiman los incrementos en las acciones en coordenadas locales en los extremos de todos los elementos con la subrutina *AEFELE*. Estos incrementos se toman como reales para los elementos lineales, pero para los elementos con rótulas plásticas se corrigen. Para ello con la subrutina *PBELRP* se identifica que elemento posee rótulas, y para cada rótula de cada elemento no lineal se identifica su número y el grado de libertad en el que actúa con la subrutina *AEIROT*, se calcula el

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

desplazamiento total de la rótula al final del intervalo y de la iteración actual, con la subrutina *AEFFIN* se calculan el momento, el parámetro de rotación elástica y la rigidez tangente de la rótula al final del intervalo y el incremento real del momento en la rótula. Los parámetros de cada rótula al final del intervalo y de la iteración se van almacenando en las respectivas tablas de resultados *ERO()*, *DRO()*, *FRO()* y *KRO()* y los incrementos reales en los momentos en las rótulas se almacenan en las respectivas celdas del vector de incrementos de las acciones en los extremos de cada elemento.

Con la subrutina *AEAJFE* se ajustan los incrementos de las acciones en los extremos de los elementos no lineales a partir de los incrementos reales de los momentos en las rótulas. Estas se convierten a coordenadas globales con la subrutina *MTMULT* y se ensamblan con la subrutina *ENSAMV* creando el vector $\{\Delta F_s^{(j)}\}$ el cual acumula directamente todos los incrementos $\{\Delta F_s^{(i,j)}\}$ calculados desde la primera hasta la última iteración. Los incrementos reales en las acciones de cada elemento se van acumulando en la columna correspondiente de la tabla resumen *FBL()* para tener las acciones al final del intervalo y de cada iteración, para lo cual se usan las subrutinas *MTSUBM*, *MTSUMA* y *MTREMP*.

En el vector $\{\Delta F_s^{(j)}\}$ existen acciones correspondientes diferentes a cero en los grados de libertad condensados, por lo que a partir de estas y con la subrutina *AEIDCN* se calcula un nuevo incremento en los desplazamientos de los nudos, el cual se acumula para conformar el vector real $\{\Delta u^{(j)}\}$ que se devuelve en la variable *DDN()*. Finalmente se calcula el error de convergencia con la subrutina *ADERRN* y se cuenta la iteración realizada.

El ciclo interno termina cuando el error calculado es inferior a la tolerancia y/o cuando el número de iteraciones realizadas es igual al número máximo de iteraciones permitido. En este instante la tabla *FBL()*, el vector *DDN()* y las matrices *ERO()*, *DRO()*, *FRO()* y *KRO()* contienen los resultados finales del ciclo iterativo. Finalmente se calcula el incremento en las acciones equivalentes trasladadas a los centros de masa $\{\Delta F_{s\phi}^{(j)}\}$ con el subvector $\{\Delta F_{s\alpha}^{(j)}\}$ de $\{\Delta F_s^{(j)}\}$ y la matriz geométrica transpuesta $[A]^T$, utilizando la subrutina *MTMULT*. El resultado se almacena en el vector *DFS()*.

7.1.24. AEIROT

La subrutina *AEIROT* identifica el número de la rótula actual y el grado de libertad del elemento al cual está asociada la rótula.

Esta subrutina es interna a la subrutina principal *AEDFSJ* previamente descrita. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.25.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

AEIROT IELE, ICOL, NRO(), IROT, GLRO

Variable	Tipo	Descripción
IELE	Entero	Elemento actual
ICOL	Entero	Columna actual de la tabla de numeración de rótulas *NRO()*
NRO()	Entero	Tabla de numeración de rótulas
IROT	Entero	Número de la rótula actual
GLRO	Entero	Grado de libertad del elemento correspondiente a la rótula actual, calculado

Tabla 7.25. Sintaxis y argumentos de la instrucción AEIROT

La subrutina toma el número de la rótula actual de la tabla *NRO()*, e identifica el grado de libertad del elemento dependiendo de la columna de la tabla *NRO()* especificada.

7.1.25. AEFFIN

La subrutina *AEFFIN* calcula el momento, el parámetro de rotación elástica y la rigidez tangente de una rótula o resorte rotacional, al final del intervalo de tiempo.

Esta subrutina es interna a la subrutina principal *AEDFSJ* previamente descrita. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.26.

AEFFIN IROT, ITIM, DFIN, PRO(), ERO(), DRO(), KFIN, FFIN, EFIN

Variable	Tipo	Descripción
IROT	Entero	Número de rótula plástica actual
ITIM	Entero	Intervalo de tiempo actual
DFIN	Real	Rotación de la rótula al final del intervalo
PRO()	Real	Tabla de propiedades de rótulas
ERO()	Real	Tabla de parámetros de rotación elástica de rótulas
DRO()	Real	Tabla de rotaciones de rótulas
KFIN	Real	Rigidez tangente al final del intervalo, calculada
FFIN	Real	Momento de la rótula al final del intervalo, calculada
EFIN	Real	Parámetro de rotación elástica al final del intervalo, calculado

Tabla 7.26. Sintaxis y argumentos de la instrucción AEFFIN

La subrutina sigue la formulación matemática descrita en el Capítulo 5 del modelo de Bouc – Wen, utilizado para representar las relación $M-\theta$ de las rótulas plásticas. Inicialmente se calcula el parámetro de deformación elástica y la derivada $de/d\theta$ al final del intervalo, teniendo en cuenta el signo del producto $ve_{t+\Delta t}^{(i,j)}$, y posteriormente se calculan la rigidez tangente y el momento en la rótula al final del intervalo.

7.1.26. AEAJFE

La subrutina *AEAJFE* ajusta los incrementos en las acciones en los extremos de los elementos inelásticos, a partir de los incrementos reales de los momentos en las rótulas del elemen-

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

to, los cuales han sido almacenados en parte o en la totalidad de los grados de libertad correspondientes a las rotaciones de los extremos en direcciones y o z local.

Esta subrutina es interna a la subrutina principal *AEDFSJ* previamente descrita. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.27.

AEAJFE IELE, ELE(), XYZ(), FEL()

Variable	Tipo	Descripción
IELE	Entero	Elemento actual
ELE()	Entero	Tabla de conectividades entre elementos
XYZ()	Real	Tabla de coordenadas de nudos y de puntos auxiliares
FEL()	Real	Vector de incremento de acciones en los extremos del elemento, corregido

Tabla 7.27. Sintaxis y argumentos de la instrucción AEAJFE

Internamente la subrutina redimensiona el vector *FEL()* utilizando la instrucción *Redim Preserve*, conservando la información inicial. Luego se calcula la longitud del elemento con la función *PBDI3D*. La subrutina corrige los incrementos en las fuerzas cortantes en los extremos a partir de los incrementos en los momentos flectores aplicando estática. Solamente se corrigen las fuerzas cortantes.

7.1.27. AEIDCN

La subrutina *AEIDCN* construye el vector de incremento de los nudos para la siguiente iteración del ciclo interno, a partir del vector de acciones equivalentes aplicadas en los nudos, el cual posee valores diferentes de cero en los grados de libertad condensados.

Esta subrutina es interna a la subrutina principal *AEDFSJ* previamente descrita. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.28.

AEIDCN NGLI, NGLO, NGLC, KGG(), FGL(), DNJ()

Variable	Tipo	Descripción
NGLI	Entero	Número de grados de libertad de interés
NGLO	Entero	Número de grados de libertad condensados
NGLC	Entero	Número de grados de libertad conocidos
KGG()	Real	Submatriz $[k_{\gamma\gamma}(t)]^{-1}$ de $[k(t)]$
FGL()	Real	Vector de incremento de acciones equivalentes en los nudos $\{\Delta F_s^{(j)}\}$
DNJ()	Real	Vector de incremento de desplazamientos de nudos para la siguiente iteración $\{\Delta u^{(i+1,j)}\}$, creado

Tabla 7.28. Sintaxis y argumentos de la instrucción AEIDCN

Esta subrutina extrae el subvector $\{\Delta F_{s\alpha}^{(j)}\}$ del vector $\{\Delta F_s^{(j)}\}$ con la subrutina *MTSUBM* y lo multiplica con $-[k_{\gamma\gamma}(t)]^{-1}$ usando las subrutinas *MTPORE* y *MTMULT*, para obtener el subvector $\{\Delta u_\gamma^{(i+1,j)}\}$. Los subvectores $\{\Delta u_\alpha^{(i+1,j)}\}$ y $\{\Delta u_\beta^{(i+1,j)}\}$ se construyen con la

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

subrutina *MTCONS* llenos de ceros, según lo indicado en el Capítulo 5. Con *MTADMR* se adjuntan los tres subvectores previamente mencionados, creando el vector *DNJ()* buscado.

7.1.28. AEVFRO

La subrutina *AEVFRO* actualiza la tabla *VF3()* de verificación de los momentos en las rótulas plásticas.

La tabla actualizada es similar a *FRO()*, aunque los momentos se toman de la matriz de acciones en coordenadas locales en los extremos de cada elemento estructural. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.29.

AEVFRO `IELE, NTIM, ROTU, NRO(), FEL(), VF3()`

Variable	Tipo	Descripción
`IELE`	Entero	Elemento actual
`NTIM`	Entero	Número de instantes de tiempo considerados en el análisis
`ROTU`	Cadena	Número de rótulas plásticas de la estructura
`NRO()`	Entero	Tabla de numeración de rótulas plásticas
`FEL()`	Real	Matriz de acciones en coordenadas locales en los extremos del elemento $[\bar{f}_e]$
`VF3()`	Real	Tabla resumen de fuerzas en las rótulas, modificada

Tabla 7.29. Sintaxis y argumentos de la instrucción AEVFRO

La subrutina lee el tamaño dela matriz *NRO()* para conocer el número de elementos y el número de posibles grados de libertad con rótulas. Si existen rótulas plásticas asociadas al elemento, con la subrutina *AEIROT* se identifica su número y el grado de libertad del elemento al que está asociada cada rótula, se extrae con *MTSUBM* la fila que almacena las acciones del elemento asociada a la rótula y estas se reemplazan en la tabla resumen *VF3()* con la subrutina *MTREMV*.

7.2. Instrucciones para análisis dinámico

7.2.1. ADMASA

La subrutina *ADMASA* crea la matriz de masa concentrada en los centros de masa de los diafragmas. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.30.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

ADMASA IELE, NTIM, ROTU, NRO(), FEL(), VF3()

Variable	Tipo	Descripción
NDIR	Entero	Número de diafragmas rígidos
NGDR	Entero	Número de grados de libertad considerados por diafragma
MAS()	Real	Tabla de masas traslacional y rotacional por diafragma
MDR()	Real	Matriz de masa $[m]$, creada

Tabla 7.30. Sintaxis y argumentos de la instrucción ADMASA

La subrutina calcula el tamaño de la matriz *MDR()*, la inicializa llena de ceros con la subrutina *MTCONS*, y la llena con los datos almacenados en la tabla *MAS()*, teniendo en cuenta si se trata de un pórtico plano o espacial. El resultado es una matriz diagonal y cuadrada como la descrita en el Capítulo 3.

7.2.2. ADPEFR

La subrutina *ADPEFR* calcula las frecuencias y los periodos de vibración natural de la estructura, a partir del vector de frecuencias de vibración al cuadrado obtenido de la solución del problema de valores propios. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.31.

ADPEFR VAP(), FRN(), PER()

Variable	Tipo	Descripción
VAP()	Real	Vector de frecuencias de vibración al cuadrado (*valores propios*)
FRN()	Real	Vector de frecuencias de vibración natural, creado
PER()	Real	Vector de periodos de vibración natural, creado

Tabla 7.31. Sintaxis y argumentos de la instrucción ADPEFR

En la subrutina se inicializan llenos de ceros los vectores columna *FRN() y PER()* con la subrutina *MTCONS* y se llenan aplicando las expresiones $\omega_i = (\omega_i^2)^{0.5}$ y $T_i = 2\pi/\omega_i$.

7.2.3. ADMMAM

La subrutina *ADMMAM* crea la matriz modal de amortiguamiento $[M]$, a partir de la fracción de amortiguamiento modal ξ y de las frecuencias de vibración natural ω_i de cada modo de vibración. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.32.

ADMMAM AMVI, FRN(), MMA()

Variable	Tipo	Descripción
AMVI	Real	Fracción de amortiguamiento modal ξ (*igual para todos los modos*)
FRN()	Real	Vector de frecuencias de vibración natural, creado
MMA()	Real	Matriz modal de amortiguamiento $[M]$, creada

Tabla 7.32. Sintaxis y argumentos de la instrucción ADMMAM

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La subrutina inicializa con *MTCONS* la matriz *MMA()* llena de ceros y calcula y almacena en esta misma matriz cada valor de la diagonal principal, el cual tiene la forma $C_i = 2\xi\omega_i$, dado que los modos de vibración son ortonormalizados, como se describe posteriormente para la subrutina *VPJACO*.

7.2.4. ADAMRS

La subrutina *ADAMRS* crea la matriz de amortiguamiento $[c]$ por superposición de matrices modales. Esta matriz se utiliza en el método de análisis modal cronológico para verificar la ecuación de movimiento y la ecuación de balance de energía. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.33.

ADAMRS `MMM(), MMA(), MOV(), MOT(), MDR(), AMR()`

Variable	Tipo	Descripción
`MMM()`	Real	Matriz modal de masa $[M]$
`MMA()`	Real	Matriz modal de masa $[C]$
`MOV()`	Real	Matriz de modos de vibración natural ortonormalizados $[\Phi]$
`MOT()`	Real	Matriz transpuesta de modos de vibración natural ortonormalizados $[\Phi]^T$
`MDR()`	Real	Matriz de masa concentrada $[m]$
`AMR()`	Real	Matriz de amortiguamiento $[c]$, creada

Tabla 7.33. Sintaxis y argumentos de la instrucción ADAMRS

En la subrutina se realizan las operaciones matriciales requeridas para obtener $[c]$ por el criterio de superposición de matrices modales descrito en el Capítulo 3, utilizando las subrutinas *MTINVE*, *MTMULT* y *MTCOPI*. El resultado se almacena en la matriz *AMR()*.

7.2.5. ADAMRY

La subrutina *ADAMRY* crea la matriz de amortiguamiento utilizando el criterio de Rayleigh, utilizada en el método de análisis dinámico inelástico. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.34.

ADAMRY `IDAR, ITIM, FRNI, FRNJ, AMVI, MDR(), KDP(), AMR()`

Variable	Tipo	Descripción
`IDAR`	Cadena	Forma de amortiguamiento seleccionada (*constante o tangente*)
`ITIM`	Entero	Intervalo de tiempo actual
`FRNI`	Real	Frecuencia de vibración ω_i
`FRNJ`	Real	Frecuencia de vibración ω_j
`AMVI`	Real	Fracción de amortiguamiento modal ξ (*igual para las dos frecuencias*)
`MDR()`	Real	Matriz de masa concentrada $[m]$
`KDP()`	Real	Matriz de rigidez lateral inicial $[k_\phi]$ o tangente $[k_{\phi(t)}]$
`AMR()`	Real	Matriz de amortiguamiento $[c]$, creada

Tabla 7.34. Sintaxis y argumentos de la instrucción ADAMRY

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La subrutina calcula los coeficientes a_0 y a_1 en función de la fracción de amortiguamiento y de las frecuencias de vibración ω_i y ω_j, con los que se superponen las matrices de masa y de rigidez para obtener la de amortiguamiento $[c]$.

Si se escoge la forma de amortiguamiento *constante*, la matriz de amortiguamiento se calcula una sola vez, en función de la matriz de rigidez lateral inicial $[k_\phi]$ mientras que si se escoge la forma de amortiguamiento *tangente*, la matriz de amortiguamiento se calcula al comienzo de cada intervalo de tiempo en función de la matriz de rigidez lateral tangente, evaluada también al comienzo del intervalo, es decir $[k_\phi(t)]$.

7.2.6. ADGAMA

La subrutina *ADGAMA* crea la matriz e influencia $[\gamma]$ en función del número de diafragmas y del número de grados de libertad por diafragma. El resultado se almacena en la matriz real denominada *GAM()*. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.35.

ADGAMA `NDIR, NGDR, GAM()`

Variable	Tipo	Descripción
`NDIR`	Entero	Número de diafragmas rígidos
`NGDR`	Entero	Número de grados de libertad por diafragma
`GAM()`	Real	Matriz de influencia $[\gamma]$, creada

Tabla 7.35. Sintaxis y argumentos de la instrucción ADGAMA

La subrutina inicializa la matriz *GAM()* llena de ceros con la subrutina *MTCONS* y coloca unos en las posiciones adecuadas, dependiendo de si el pórtico analizado es plano o espacial, de acuerdo con lo descrito en el Capítulo 3.

7.2.7. ADALFA

La subrutina *ADALFA* crea la matriz de factores de participación $[\alpha]$ en función de la matriz de modos de vibración, la matriz de masa y la matriz de influencia. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.36.

ADALFA `MOT(), MDR(), GAM(), ALF()`

Variable	Tipo	Descripción
`MOT()`	Real	Matriz transpuesta de modos de vibración natural ortonormalizados $[\Phi]^T$
`MDR()`	Real	Matriz de masa concentrada $[m]$
`GAM()`	Real	Matriz de influencia $[\gamma]$
`ALF()`	Real	Matriz de factores de participación $[\alpha]$, creada

Tabla 7.36. Sintaxis y argumentos de la instrucción ADALFA

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La subrutina realiza el triple producto $[\Phi]^T[m][\gamma]$ utilizando las subrutina *MTMULT* y el resultado lo almacena en la matriz real *ALF()*.

7.2.8. ADMASM

La subrutina *ADMASM* crea las matrices $[\overline{m}]$ y $[M_T]$ de masa modal efectiva y masa traslacional y rotacional total de la estructura, a partir de la matriz de factores de participación. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.37.

ADMASM `ALF(), MME(), MMP(), MTE()`

Variable	Tipo	Descripción
`ALF()`	Real	Matriz de factores de participación $[\alpha]$
`MME()`	Real	Matriz de masa modal efectiva $[\overline{m}]$, creada
`MMP()`	Real	Matriz de masa modal efectiva como fracción de la masa total, creada
`MTE()`	Real	Matriz de masa total $[M_T]$, creada

Tabla 7.37. Sintaxis y argumentos de la instrucción ADMASM

La subrutina inicializa las matrices *MME() y MMP()* llenas de ceros, con un tamaño igual al de la matriz *ALF()*, utilizando la subrutina *MTCONS*. La matriz $[\overline{m}]$ se construye elevando al cuadrado cada coeficiente de la matriz $[\alpha]$ dado que los modos de vibración son ortonormalizados. La matriz de masa total de la estructura $[M_T]$ se crea con el producto $[\alpha]^T[\alpha]$, el cual se realiza con las subrutinas *MTTRAN* y *MTMULT*. Finalmente la matriz *MMP()* se construye dividiendo cada coeficiente de la matriz $[\overline{m}]$ entre el coeficiente respectivo de la matriz $[M_T]$, de forma que *MMP(i,j)=MME(i,j)/MTE(j,j)*.

7.2.9. ADUGXY

La subrutina *ADUGXY* crea la matriz de aceleración de la base $[\ddot{u}_g]$ utilizada en los métodos de análisis modal cronológico y de análisis dinámico inelástico, a partir de los acelerogramas en dirección x e y introducidos como dato de entrada y almacenados en las matrices *UGX()* y *UGY()*, amplificadas con los factores de escala correspondientes.

Adicionalmente la subrutina crea un vector columna que almacena los instantes de tiempo considerados en el análisis, y una tabla de datos que almacena las componentes de aceleración en la base consideradas en el análisis, como fracción de la aceleración de la gravedad. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.38.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

ADUGXY NACE, DTIM, FESX, FEXY, NGDR, UGX(), UGY(), ACE(), TME(),MUO()

Variable	Tipo	Descripción
NACE	Entero	Número de datos de cada acelerograma
DTIM	Real	Tamaño del intervalo de tiempo
FESX	Real	Factor de escala de la aceleración de la base en dirección X
FESY	Real	Factor de escala de la aceleración de la base en dirección Y
NGDR	Entero	Número de grados de libertad por diafragma
UGX()	Real	Tabla que almacena los datos del acelerograma en dirección X
UGY()	Real	Tabla que almacena los datos del acelerograma en dirección Y
ACE()	Real	Tabla de acelerogramas
TME()	Real	Vector columna de instantes de tiempo
MUO()	Real	Matriz de aceleración en la base $[\ddot{u}_g]$, creada

Tabla 7.38. Sintaxis y argumentos de la instrucción ADUGXY

La subrutina construye el vector *TME()* en función del número de datos de cada acelerograma y del tamaño del intervalo de tiempo, de forma que los datos quedan con un espaciamiento igual al tamaño del intervalo de tiempo.

Con la subrutina *MTCONS* inicializa las matrices *ACE()* y *MUO()* llenas de ceros, con un numero de filas igual al número de grados de libertad por diafragma *NGDR*. En la primera matriz almacena secuencialmente los datos almacenados en las tablas *UGX()* y/o *UGY()* y en la segunda almacena los mismos datos, pero multiplicados por los factores de escala *FESX* y/o *FESY* según el tipo de estructura (*plana o espacial*). Si el pórtico es espacial la tercera columna, que corresponde a la aceleración angular de la base en dirección *z* global $\ddot{\theta}_{gz}$, queda llena de ceros.

7.2.10. ADACEL, ADWILS y ADNEWM

Las subrutinas *ADACEL, ADWILS y ADNEWM* se utilizan en el método de análisis modal cronológico para calcular los desplazamientos, las velocidades y las aceleraciones generalizadas en cada instante de tiempo, solución de las ecuaciones modales desacopladas.

La primera subrutina resuelve las ecuaciones modales desacopladas utilizando el método de la aceleración lineal, la segunda subrutina las resuelve el método basado en la linealización de la excitación externa (d*enominado aquí el método de la excitación lineal*) y la última subrutina utiliza la forma implícita (*iterativa*) del método de Newmark.

Las tres subrutinas utilizan los mismos parámetros de entrada y entregan como resultado las mismas matrices de resultados, por lo que se agruparon en el mismo numeral. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.39.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

```
ADACEL AMVI, FRN(), DTIM, TEI(), DCG(), VCG(), ACG()
ADWILS AMVI, FRN(), DTIM, TEI(), DCG(), VCG(), ACG()
ADNEWM AMVI, FRN(), DTIM, TEI(), DCG(), VCG(), ACG()
```

Variable	Tipo	Descripción
AMVI	Real	Número de datos de cada acelerograma
FRN()	Real	Vector de frecuencias de vibración natural de la estructura
DTIM	Real	Tamaño del intervalo de tiempo
TEI()	Real	Matriz de acciones modales externas equivalentes a la aceleración de la base $[\bar{F}_{ext}]$, creada
DCG()	Real	Matriz de desplazamientos generalizados $[\eta]$, creada
VCG()	Real	Matriz de velocidades generalizadas $[\dot{\eta}]$, creada
ACG()	Real	Matriz de aceleraciones generalizadas $[\ddot{\eta}]$, creada

Tabla 7.39. Sintaxis y argumentos de las instrucciones ADACEL, ADWILS y ADNEWM

En la subrutina *ADACEL* se inicializan, con la subrutina *MTCONS*, las matrices *DCG(), VCG() y ACG()* llenas de ceros, con el mismo tamaño de la matriz *TEI()*. Luego, para cada intervalo de tiempo, se calculan las aceleraciones, las velocidades y los desplazamientos generalizados al final del intervalo a partir de los mismos valores al comienzo del intervalo, de la fracción de amortiguamiento modal, de las frecuencias de vibración y de las acciones modales al final del intervalo. Cada aceleración generalizada en el instante $t = 0$ se despeja de la respectiva ecuación modal en el mismo instante, asumiendo la velocidad y el desplazamiento generalizado iguales a cero.

En la subrutina *ADWILS* nuevamente se inicializan con la subrutina *MTCONS* las matrices *DCG(), VCG() y ACG()* llenas de ceros y con el mismo tamaño de la matriz *TEI()*. Luego, para la solución de cada ecuación modal, se calculan las diferentes constantes necesarias y finalmente se calculan las coordenadas generalizadas al final del intervalo, a partir del desplazamiento y de la velocidad generalizada al comienzo del intervalo y de las acciones modales al comienzo y al final del intervalo, siguiendo la formulación matemática indicada en el Capítulo 4.

De forma similar, en la subrutina *ADNEWM* nuevamente se inicializan con la subrutina *MTCONS* las matrices *DCG(), VCG() y ACG()* llenas de ceros y con el mismo tamaño de la matriz *TEI()*. Se utilizan por defecto los parámetros $\gamma = 1/4$ y $\beta = 1/6$. Para cada ecuación modal y en cada intervalo de tiempo se asume la aceleración generalizada al final de intervalo igual a la aceleración generalizada al comienzo del intervalo, luego con las ecuaciones básicas del método de Newmark se calculan la velocidad generalizada y el desplazamiento generalizado al final del intervalo, y con estas se recalcula la aceleración generalizada al final del intervalo, la cual se compara con la supuesta, obteniendose un error. Para la nueva iteración se asume la aceleración al final del intervalo que se acaba de calcular, y esta se recalcula nuevamente hasta que el error sea inferior a una tolerancia especificada.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

7.2.11. ADVMAX

La subrutina *ADVMAX* crea una tabla resumen de dos columnas, que almacena en la primera columna los valores máximos absolutos y en la segunda columna, los instantes de tiempo en los que se presenta cada valor máximo, de cualquier parámetro de respuesta variable en el tiempo (*desplazamientos, fuerzas internas, etc.*).

En la tabla de datos cada fila corresponde a un parámetro distinto y cada columna corresponde a un instante de tiempo. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.40.

ADVMAX `MXX(), TME(), MXR()`

Variable	Tipo	Descripción
`MXX()`	Real	Tabla de datos del parámetro de respuesta de la estructura
`TME()`	Real	Vector columna de instantes de tiempo considerados en el análisis
`MXR()`	Real	Tabla resumen de valores máximos y tiempos, creada

Tabla 7.40. Sintaxis y argumentos de las instrucciones ADVMAX

Internamente se lee el tamaño de la matriz *MXX()*, y se inicializa la tabla *MXR()* de dos columnas e igual número de filas a *MXX()*. Luego se obtiene y se almacena en la primera columna de *MXR()* el valor máximo absoluto de cada fila de datos, y de *TME()* se toma el instante de tiempo correspondiente, el cual se almacena en la segunda columna de *MXR()*.

7.2.12. ADESPT

La subrutina *ADESPT* interpola del espectro de pseudo – aceleraciones los valores de S_a correspondientes a los periodos de vibración natural de la estructura. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.41.

ADESPT `MXX(), TME(), MXR()`

Variable	Tipo	Descripción
`PER()`	Real	Vector columna de periodos de vibración natural de la estructura
`ESP()`	Real	Tabla que almacena los datos del espectro de pseudo - aceleraciones
`SAT()`	Real	Tabla de pseudo – aceleraciones interpoladas, creada

Tabla 7.41. Sintaxis y argumentos de las instrucciones ADESPT

En la subrutina se inicializa llena de ceros la matriz *SAT()* de dos columnas e igual número de filas al de *PER()*. Luego cada periodo de vibración se copia en la primera columna de *SAT()* y en la segunda columna se almacenan los valores interpolados de S_a.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

7.2.13. ADCGEN

La subrutina *ADCGEN* crea la matriz $[\eta_o]$ de desplazamientos generalizados máximos por modo y por componente de aceleración de la base considerada en el análisis. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.42.

ADCGEN `FESX, FESY, SAT(), ALF(), FRN(), GEN()`

Variable	Tipo	Descripción
`FESX`	Real	Factor de escala del espectro en dirección X
`FESY`	Real	Factor de escala del espectro en dirección Y
`SAT()`	Real	Tabla de pseudo – aceleraciones interpoladas
`ALF()`	Real	Matriz de factores de participación $[\alpha]$
`FRN()`	Real	Vector columna de frecuencias de vibración natural de la estructura
`GEN()`	Real	Matriz de desplazamientos generalizados máximos $[\eta_o]$, creada

Tabla 7.42. Sintaxis y argumentos de las instrucciones ADCGEN

La subrutina sigue la formulación mostrada en el Capítulo 4 para el cálculo de los desplazamientos generalizados máximos. Los resultados se almacenan en la matriz *GEN()* en la que cada fila corresponde a un modo de vibración y cada columna corresponde a una componente de aceleración de la base. El tamaño de la matriz *GEN()* coincide con el de la matriz *ALF()*.

7.2.14. ADDESP

La subrutina *ADDESP* crea las matrices de desplazamientos máximos por modo de los centros de masa, cuando se aplican las componentes de aceleración de la base en las direcciones x e y global de forma independiente, denominadas $[(u_\phi)_o^x]$ y $[(u_\phi)_o^y]$ respectivamente. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.43.

ADDESP `GEN(), MOV(), DCX(), DCY()`

Variable	Tipo	Descripción
`GEN()`	Real	Matriz de desplazamientos generalizados máximos $[\eta_o]$
`MOV()`	Real	Matriz de modos de vibración $[\Phi]$
`DCX()`	Real	Matriz de desplazamientos de centros de masa por sismo en dirección X, creada
`DCY()`	Real	Matriz de desplazamientos de centros de masa por sismo en dirección Y, creada

Tabla 7.43. Sintaxis y argumentos de las instrucciones ADDESP

Las matrices *DCX()* y *DCY()* se inicializan llenas de ceros con la subrutina *MTCONS*. Luego, con la subrutina *MTDIAV*, se crea la matriz diagonal $[\eta_o^x]$ descrita en el Capítulo 4, la cual se premultiplica con la matriz de modos de vibración *MOV()* para crear *DCX()*. Si el pórtico analizado es espacial, adicionalmente se crea la matriz diagonal $[\eta_o^y]$ para calcular *DCY()*.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

7.2.15. ADMCOR

La subrutina *ADMCOR* crea la matriz de coeficientes de correlación $[\rho]$ utilizada en el método de la *combinación cuadrática completa (CCC)* para la combinación de la respuesta máxima por modo. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.44.

ADMCOR `IDMA, IDCM, AMVI, FRN(), COR()`

Variable	Tipo	Descripción
`IDMA`	Cadena	Método de análisis dinámico
`IDCM`	Cadena	Método de combinación modal
`AMVI`	Real	Fracción de amortiguamiento modal
`FRN()`	Real	Frecuencias de vibración natural de la estructura
`COR()`	Real	Matriz de coeficientes de correlación $[\rho]$, creada

Tabla 7.44. Sintaxis y argumentos de las instrucciones ADMCOR

La subrutina crea la matriz cuadrada *COR()* de tamaño igual al número de modos de vibración siguiendo la formulación matemática descrita en el Capítulo 4, siempre y cuando se hayan escogido como método de análisis dinámico y como método de combinación modal los métodos "*análisis modal espectral*" y "*CQC*" respectivamente. De lo contrario se creará una tabla llena de ceros que no será usada en ninguna subrutina.

7.2.16. ADCOMM

La subrutina *ADCOMM* realiza la combinación modal de la respuesta máxima, calculada de forma separada para cada modo de vibración y para cada componente de aceleración de la base considerada en el análisis. Se dispone de los métodos *CCC* y *RCSC* para tal fin. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.45.

ADCOMM `IDCM, COR(), MXX(), VXX()`

Variable	Tipo	Descripción
`IDCM`	Cadena	Método de combinación modal
`COR()`	Real	Matriz de coeficientes de correlación $[\rho]$
`MXX()`	Real	Tabla de datos de parámetros a combinar
`VXX()`	Real	Vector que contiene los parámetros combinados, creado

Tabla 7.45. Sintaxis y argumentos de las instrucciones ADCOMM

La subrutina utiliza la formulación matemática indicada en el Capítulo 4, relacionada con los métodos de combinación modal. En la tabla *MXX()* cada fila debe corresponder a un mismo parámetro de respuesta y cada columna a un modo de vibración, y la tabla debe almacenar la respuesta producida por una de las componentes de aceleración de la base, por lo que la subrutina debe aplicarse una vez por cada componente.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

7.2.17. ADCOMD

La subrutina *ADCOMD* realiza la combinación direccional de las respuestas máximas probables de la estructura, producidas por cada componente de aceleración de la base actuando de forma independiente. Se dispone de los métodos *100/30* y *RCSC* para tal fin. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.46.

ADCOMD `IDCD, MX1(), MX2(), MAX()`

Variable	Tipo	Descripción
`IDCD`	Cadena	Método de combinación direccional
`MX1()`	Real	Vector de respuesta máxima probable por aceleración de la base en dirección X
`MX2()`	Real	Vector de respuesta máxima probable por aceleración de la base en dirección Y
`MAX()`	Real	Vector de respuesta máxima probable por aceleración de la base en direcciones X e Y de forma simultánea, creado

Tabla 7.46. Sintaxis y argumentos de las instrucciones ADCOMD

La subrutina utiliza la formulación matemática descrita en el Capítulo 4 para realizar la combinación direccional de la respuesta máxima probable de la estructura, cuando actúa cada componente de aceleración de la base de forma individual, y así estimar la respuesta máxima producida por todas las componentes de aceleración de la base actuando de forma simultánea.

7.2.18. ADTRAE

La subrutina *ADTRAE* crea una tabla resumen de respuesta máxima, que almacena simultáneamente la respuesta máxima por modo y por componente de aceleración de la base, la respuesta máxima probable producida por cada componente de aceleración de la base (*obtenida por combinación modal*) y la respuesta máxima probable producida por todas las componentes de aceleración de las base actuando en forma simultánea (*obtenida por combinación direccional*). La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.47.

ADTRAE `IDCM, IDCD, COR(), MXX(), MYY(), MRR()`

Variable	Tipo	Descripción
`IDCM`	Cadena	Método de combinación modal
`IDCD`	Cadena	Método de combinación direccional
`COR()`	Real	Matriz de coeficientes de correlación
`MXX()`	Real	Matriz de respuesta máxima por modo, producida por la aceleración de la base en dirección X
`MYY()`	Real	Matriz de respuesta máxima por modo, producida por la aceleración de la base en dirección Y
`MRR()`	Real	Tabla resumen de respuesta máxima de la estructura, creada

Tabla 7.47. Sintaxis y argumentos de las instrucciones ADTRAE

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La subrutina crea una tabla resumen con un número de columnas igual a $2N + 3$ siendo N el número de modos de vibración. En las primeras N columnas se almacena la respuesta máxima por modo, producida por la aceleración de la base en dirección x, en las siguientes N columnas se almacena la respuesta máxima por modo producida por la aceleración de la base en dirección y, en las siguientes dos columnas se almacena la respuesta máxima probables de la estructura producida por cada componente en x e y de la aceleración de la base actuando de forma independiente, calculada utilizando algún método de combinación modal con la subrutina *ADCOMM*.

En la última columna se almacena la respuesta máxima probable producida por la acción simultanea de la componentes x e y de la aceleración de la base, calculada con algún método de combinación direccional con la subrutina *ADCOMD*. Finalmente con la subrutina *MTADVM* se adjuntan las cinco matrices individuales para conformar la matriz *MRR()* final.

7.2.19. ADACCM

La subrutina *ADACCM* crea una tabla resumen completa de acciones internas para cada eje de vigas o de columnas, calculadas por análisis modal espectral, a partir de la tabla de acciones internas por modo producida con la subrutina *AEDIVC*. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.48.

ADACCM `IDCM, IDCD, NMOD, COR(), MXX(), MRR()`

Variable	Tipo	Descripción
`IDCM`	Cadena	Método de combinación modal
`IDCD`	Cadena	Método de combinación direccional
`NMOD`	Entero	Número de modos de vibración natural
`COR()`	Real	Matriz de coeficientes de correlación
`MXX()`	Real	Matriz de acciones internas que contiene resultados por modo producidos por la aceleración de la base en direcciones X e Y, generada con la subrutina *AEDIVC*
`MRR()`	Real	Tabla resumen de acciones internas por análisis modal espectral, creada

Tabla 7.48. Sintaxis y argumentos de las instrucciones ADACCM

La tabla resumen generada posee $2N + 5$ columnas, en la primera columna almacena las secciones donde se calculan las acciones internas, las siguientes $2N$ columnas almacenan las acciones máximas por modo, las siguientes dos columnas almacenan las acciones máximas probables calculadas por combinación modal, la penúltima columna almacena las acciones máximas probables calculadas por combinación direccional, y la última columna almacena los valores negativos de la penúltima columna.

Las dos últimas columnas corresponderían a las envolventes de acciones máximas positivas y negativas. Como en la subrutina *ADTRAE*, nuevamente se utilizan las subrutinas *ADCOMM*, *ADCOMD* y *MTADVM*.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

7.2.20. ADTRDI

La subrutina *ADTRDI* crea las tablas resumen *NXX(), VYY(), VZZ(), TXX(), MYY()* y *MZZ()* de acciones internas, calculadas por análisis modal espectral. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.49.

```
ADTRDI IDCM, IDCD, NMOD, COR(), NXX(), VYY(), VZZ(), TXX(), MYY(),
                                                              MZZ()
```

Variable	Tipo	Descripción
IDCM	Cadena	Método de combinación modal
IDCD	Cadena	Método de combinación direccional
NMOD	Entero	Número de modos de vibración natural
COR()	Real	Matriz de coeficientes de correlación
NXX()	Real	Tabla resumen de fuerza axial, creada
VYY()	Real	Tabla resumen de fuerza cortante en dirección *y* local, creada
VZZ()	Real	Tabla resumen de fuerza cortante en dirección *z* local, creada
TXX()	Real	Tabla resumen de momento torsor, creada
MYY()	Real	Tabla resumen de momento flector en dirección *y* local, creada
MZZ()	Real	Tabla resumen de momento flector en dirección z local, creada

Tabla 7.49. Sintaxis y argumentos de las instrucciones ADTRDI

La subrutina construye cada tabla resumen mediante la subrutina *ADACCM* previamente descrita. Su construcción se realizó básicamente para reducir el número de líneas en el código principal.

7.2.21. ADACIN

La subrutina *ADACIN* calcula las aceleraciones de los centros de masa en el instante $t = 0$, necesarias en el método de análisis dinámico inelástico, inicializando la primera columna de la matriz $[\ddot{u}_\phi]$. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.50.

```
ADACIN MDR(), FEQ(), FEX(), ACM()
```

Variable	Tipo	Descripción
MDR()	Real	Matriz de masa $[m]$
FEQ()	Real	Matriz de acciones equivalentes trasladadas a centros de masa $[(F_s)_\phi]$
FEX()	Real	Matriz de acciones externas $[F_{ext}]$
ACM()	Real	Matriz de aceleraciones de centros de masa $[\ddot{u}_\phi]$, inicializada

Tabla 7.50. Sintaxis y argumentos de las instrucciones ADACIN

La subrutina extrae la primera columna de las matrices *FEX() y FEQ()* con la subrutina *MTSUBM*, asume que las acciones iniciales de amortiguamiento son cero (*puesto que las velocidades iniciales de los centros de masa se asumieron también como cero*) y despeja de

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

la ecuación de movimiento en $t = 0$ las aceleraciones, las cuales finalmente almacena en la primera columna de la matriz *ACM()* con la subrutina *MTREMV*.

7.2.22. ADCONS

La subrutina *ADCONS* calcula las constantes c_1 a c_9 de la familia de métodos de Newmark, necesarias para solucionar la ecuación matricial de movimiento en forma incremental. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.51.

ADCONS `IDSD, DTIM, CSD()`

Variable	Tipo	Descripción
`IDSD`	Real	Forma del método de Newmark utilizada
`DTIM`	Real	Tamaño del intervalo de tiempo
`CSD()`	Real	Vector de constantes de los métodos de solución directa

Tabla 7.51. Sintaxis y argumentos de las instrucciones ADCONS

La subrutina selecciona los coeficientes γ y β dependiendo de la forma del método de Newmark seleccionada (*aceleración promedio o aceleración lineal*) y calcula las 9 constantes descritas en el Capítulo 5 en la formulación del método, las cuales almacena en el vector *CSD()*. Adicionalmente la subrutina almacena los parámetros Δt, γ y β al final del mismo vector.

7.2.23. ADMAAB

La subrutina *ADMAAB* crea las matrices auxiliares $[a(t)]$ y $[b(t)]$ requeridas para calcular el vector $\{\Delta\hat{F}_{ext}\}$ en cada intervalo de tiempo. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.52.

ADMAAB `IDAR, ITIM, MDR(), AMR(), CSD(), AAA(), BBB()`

Variable	Tipo	Descripción
`IDAR`	Cadena	Forma de amortiguamiento de Rayleigh
`ITIM`	Entero	Intervalo de tiempo actual
`MDR()`	Real	Matriz de masa $[m]$
`AMR()`	Real	Matriz de amortiguamiento $[c]$ o $[c(t)]$
`CSD()`	Real	Vector de constantes de los métodos de solución directa
`AAA()`	Real	Matriz auxiliar $[a]$ o $[a(t)]$
`AAA()`	Real	Matriz auxiliar $[b]$ o $[b(t)]$

Tabla 7.52. Sintaxis y argumentos de las instrucciones ADMAAB

La subrutina utiliza la formulación matemática presentada en el Capítulo 5 para construir las matrices *AAA()* y *BBB()* en función de las matrices de masa y amortiguamiento. Cuando se escoge la forma de amortiguamiento *constante*, las matrices *AAA()* y *BBB()* se

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

calculan una sola vez, en el primer intervalo de tiempo, mientras que si se escoge la forma de amortiguamiento *tangente*, las matrices *AAA()* y *BBB()* se calculan al comienzo de cada intervalo de tiempo utilizando la matriz de amortiguamiento tangente $[c(t)]$.

7.2.24. ADMARA

La subrutina *ADMARA* crea las matrices $[\hat{k}_\phi(t)]$, $[\hat{k}_\phi(t)]^{-1}$ y $\left([\hat{k}_\phi(t)] - [k_\phi(t)]\right)$ a partir de la matriz $[k_\phi(t)]$. Todas las matrices son evaluadas al comienzo del intervalo. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.53.

ADMARA `CSD(),MDR(), AMR(), KDP(), KTE(), KEI(), KTR()`

Variable	Tipo	Descripción
`CSD()`	Real	Vector de constantes de los métodos de solución directa
`MDR()`	Real	Matriz de masa $[m]$
`AMR()`	Real	Matriz de amortiguamiento $[c]$ o $[c(t)]$
`KDP()`	Real	Matriz de rigidez lateral tangente $\left[k_\phi(t)\right]$
`KTE()`	Real	Matriz de rigidez lateral tangente efectiva $[\hat{k}_\phi(t)]$, creada
`KEI()`	Real	Inversa de la matriz de rigidez lateral tangente efectiva $[\hat{k}_\phi(t)]^{-1}$, creada
`KTR()`	Real	Matriz resta $[\hat{k}_\phi(t)] - [k_\phi(t)]$, creada

Tabla 7.53. Sintaxis y argumentos de las instrucciones ADMARA

La subrutina sigue la formulación matemática mostrada en el Capítulo 5 para calcular cada una de las matrices previamente mencionadas. Para realizar las operaciones matriciales se utilizan las subrutinas *MTPORE*, *MTSUMA* y *MTINVE*.

7.2.25. ADIEFE

La subrutina *ADIEFE* calcula el vector $\{\Delta\hat{F}_{ext}\}$ de incremento efectivo en el intervalo de las acciones externas equivalentes a la aceleración de la base. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.54.

ADIEFE `ITIM, CSD(), FEX(), AAA(), BBB(), VCM(), ACM(), FEI()`

Variable	Tipo	Descripción
`ITIM`	Entero	Intervalo de tiempo actual
`CSD()`	Real	Vector de constantes de los métodos de solución directa
`FEX()`	Real	Matriz de acciones externas $[F_{ext}]$
`AAA()`	Real	Matriz auxiliar $[a]$ o $\left[a_{(t)}\right]$
`BBB()`	Real	Matriz auxiliar $[b]$ o $\left[b_{(t)}\right]$
`VCM()`	Real	Matriz de velocidades de centros de masa $\left[\dot{u}_\phi\right]$
`ACM()`	Real	Matriz de aceleraciones de centros de masa $\left[\ddot{u}_\phi\right]$
`FEI()`	Real	Vector de incremento efectivo de las acciones externas $\{\Delta\hat{F}_{ext}\}$, creado

Tabla 7.54. Sintaxis y argumentos de las instrucciones ADIEFE

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La subrutina extrae las velocidades y las aceleraciones de los centros de masa al comienzo del intervalo y calcula el incremento en las acciones externas en el intervalo. A partir de estas tres cantidades, y de las matrices auxiliares *AAA()* y *BBB()*, se calcula el vector $\{\Delta\hat{F}_{ext}\}$ buscado, utilizando las subrutinas *MTSUMA y MTMULT* para realizar las operaciones matriciales requeridas.

7.2.26. ADINVA

La subrutina *ADINVA* inicializa las variables necesarias en el proceso iterativo principal del método de Newmark para la solución directa de las ecuaciones de movimiento. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.55.

```
ADINVA ITIM, NGLT, ITED, ITEF, ERRF, FEI(), DDN(), DRJ(), ERO(), DRO(),
                                                    FRO(), FEQ(), FBL()
```

Variable	Tipo	Descripción
ITIM	Entero	Intervalo de tiempo actual
NGLT	Entero	Número de grados de libertad total
ITED	Entero	Número de iteraciones para convergencia de desplazamientos, inicializado
ITEF	Entero	Número de iteraciones para convergencia de fuerzas, inicializado
ERRF	Real	Error para convergencia de fuerzas, inicializado
FEI()	Real	Vector de incremento efectivo de las acciones externas $\{\Delta\hat{\mathrm{F}}_{\mathrm{ext}}\}$
DDN()	Real	Incremento de desplazamientos de nudos en el intervalo $\{\Delta u\}$, inicializado
DRJ()	Real	Acciones residuales para la iteración *j*, $\{\Delta R^{(j)}\}$, modificada
ERO()	Real	Tabla de parámetros de deformación elástica de rótulas, modificada
DRO()	Real	Tabla de desplazamientos de rótulas, modificada
FRO()	Real	Tabla de fuerzas de rótulas, modificada
FEQ()	Real	Matriz de acciones equivalentes trasladadas a centros de masa $[F_{s\phi}]$, modificada
FBL()	Real	Tabla resumen de acciones en los extremos de los elementos, modificada

Tabla 7.55. Sintaxis y argumentos de las instrucciones ADINVA

La subrutina inicializa en cero las variables *ITED*, *ITEF* y *ERRF*, crea un vector lleno de ceros para *DDN()*, el vector *DRJ()* lo inicializa con los valores del vector *FEI()* mediante la subrutina *MTCOPI* y con la subrutina *MTCPCO* copia los datos de la columna correspondiente al comienzo del intervalo de las tablas *ERO(), DRO(), FRO(), FEQ() y FBL()* en la columna siguiente, es decir al final del intervalo.

7.2.27. ADREFI

La subrutina *ADREFI* calcula los desplazamientos, las velocidades y las aceleraciones de los centros de masa al final del intervalo, en función de los incrementos de desplazamientos en el intervalo y de los mismos parámetros al comienzo del intervalo.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

Esta subrutina se utiliza en el método de análisis dinámico inelástico. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.56.

ADREFI ITIM, NDIN, CSD(), DDC(), DCM(), VCM(), ACM()

Variable	Tipo	Descripción
ITIM	Entero	Intervalo de tiempo actual
NDIN	Entero	Número de grados de libertad dinámicos de la estructura
CSD()	Real	Vector de constantes de los métodos de solución directa
DDC()	Real	Incremento de desplazamientos de centros de masa en el intervalo $\{\Delta u_\phi\}$
DCM()	Real	Matriz de desplazamientos de centros de masa $[u_\phi]$, modificada
VCM()	Real	Matriz de velocidades de centros de masa $[\dot{u}_\phi]$, modificada
ACM()	Real	Matriz de aceleraciones de centros de masa $[\ddot{u}_\phi]$, modificada

Tabla 7.56. Sintaxis y argumentos de las instrucciones ADREFI

La subrutina extrae con *MTSUMB* los desplazamientos, las velocidades y las aceleraciones de los centros de masa al comienzo del intervalo, calcula los incrementos de las velocidades y las aceleraciones utilizando las ecuaciones descritas en el Capítulo 5 realizando las operaciones necesarias con las subrutinas *MTPORE*, *MTSUMA* y *MTREST*, calcula los desplazamientos, las velocidades y las aceleraciones al final del intervalo y con la subrutina *MTREMV*, y las almacena en la columna correspondiente de las matrices *DCM()*, *VCM()* y *ACM()* respectivamente.

7.2.28. ADERRN

La subrutina *ADERRN* calcula el error en la convergencia de desplazamientos, comparando las normas de Euclides, o de las magnitudes de los vectores de incrementos en desplazamientos involucrados. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.57.

ADERRN DCJ(), DDC(), ERRN

Variable	Tipo	Descripción
DCJ()	Real	Incremento de desplazamientos en el intervalo y en la iteración actual $\{\Delta u^{(j)}\}$
DDC()	Real	Incremento de desplazamientos en el intervalo, acumulados hasta la iteración actual $\{\Delta u\}$
ERRN	Real	Error por comparación de normas de vectores de desplazamientos, calculado

Tabla 7.57. Sintaxis y argumentos de las instrucciones ADERRN

La subrutina calcula las normas de los vectores *DCJ()* y *DDC()*, a partir de las cuales calcula el error requerido como el cociente de las dos normas, de acuerdo a lo indicado en el Capítulo 5 sobre criterios de convergencia.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

7.2.29. ADERRD

La subrutina *ADERRD* calcula el error en la convergencia de desplazamientos, utilizando el criterio de la comparación del trabajo realizado por las acciones residuales en cada iteración, y el trabajo realizado por el incremento efectivo de las acciones externas. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.58.

ADERRD `DRJ(), DCJ(), FEI(), DDC(), ERRD`

Variable	Tipo	Descripción
`DRJ()`	Real	Acciones residuales para la iteración *j*, $\{\Delta R^{(j)}\}$
`DCJ()`	Real	Incremento de desplazamientos en el intervalo y en la iteración actual $\{\Delta u^{(j)}\}$
`FEI()`	Real	Vector de incremento efectivo de las acciones externas $\{\Delta \hat{\mathrm{F}}_{\mathrm{ext}}\}$
`DDC()`	Real	Incremento de desplazamientos en el intervalo, acumulados hasta la iteración actual $\{\Delta u\}$
`ERRD`	Real	Error por comparación de normas de vectores de desplazamientos, calculado

Tabla 7.58. Sintaxis y argumentos de las instrucciones ADERRD

La subrutina calcula los trabajos realizados por las acciones residuales $\{\Delta R^{(j)}\}$ y por las acciones efectivas $\{\Delta \hat{\mathrm{F}}_{\mathrm{ext}}\}$, y calcula el error requerido como el cociente de los dos trabajos, de acuerdo a lo indicado en el Capítulo 5 sobre criterios de convergencia.

7.2.30. ADFEQF

La subrutina *ADFEQF* calcula las acciones equivalentes trasladadas a los centros de masa y al final del intervalo $\{F_{s\phi}(t+\Delta t)\}$, asi como el incremento efectivo de las acciones equivalentes en cada iteración del ciclo externo $\{\Delta F^{(j)}\}$, a partir del incremento de las acciones equivalentes $\{\Delta F_{s\phi}^{(j)}\}$ en la misma iteración.

Las acciones equivalentes al final del intervalo se almacenan en la columna correspondiente de la matriz $[F_{s\phi}]$. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.59.

ADFEQF `ITIM, KTR(), DCJ(), DFS(), FEQ(), DFJ()`

Variable	Tipo	Descripción
`ITIM`	Entero	Intervalo de tiempo actual
`KTR()`	Real	Matriz resta $[\hat{k}_{\phi}(t)] - [k_{\phi}(t)]$
`DCJ()`	Real	Incremento de desplazamientos en el intervalo y en la iteración actual $\{\Delta u^{(j)}\}$
`DFS()`	Real	Incremento de acciones equivalentes trasladadas a los centros de masa en el intervalo y en la iteración actual $\{\Delta F_{s\phi}^{(j)}\}$
`FEQ()`	Real	Matriz de acciones equivalentes $[F_{s\phi}]$, modificada
`DFJ()`	Real	Incremento efectivo de acciones equivalentes $\{\Delta F^{(j)}\}$, creado

Tabla 7.59. Sintaxis y argumentos de las instrucciones ADFEQF

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La subrutina calcula las acciones equivalentes al final del intervalo sumando el incremento almacenado en *DFS()* a las acciones al comienzo del intervalo, y las almacena en la columna respectiva de la matriz *FEQ()* con la subrutina *MTREMV*. Adicionalmente para el cálculo del vector *DFJ()* las operaciones matriciales requeridas se realizan con las subrutinas *MTMULT()* y *MTSUMA()*.

7.2.31. ADAMFI

La subrutina *ADAMFI* calcula las acciones de amortiguamiento al final del intervalo. La subrutina se usa en el método de análisis dinámico inelástico. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.60.

ADAMFI `ITIM, VCM(), AMR(), FAM()`

Variable	Tipo	Descripción
`ITIM`	Entero	Intervalo de tiempo actual
`VCM()`	Real	Matriz de velocidades de centros de masa $[\dot{u}_\phi]$
`AMR()`	Real	Matriz de amortiguamiento $[c]$ o $[c(t)]$
`FAM()`	Real	Matriz de acciones de amortiguamiento $[F_{a\phi}]$, modificada

Tabla 7.60. Sintaxis y argumentos de las instrucciones ADAMFI

La subrutina calcula las acciones de amortiguamiento al final del intervalo sumando el incremento en las acciones de amortiguamiento a las acciones al comienzo del intervalo, y las almacena en la columna respectiva de la matriz *FAM()* con la subrutina *MTREMV*. El incremento de las acciones de amortiguamiento se calcula con base en el incremento en las velocidades de los centros de masa $[\Delta\dot{u}_\phi]$ y en la matriz de amortiguamiento $[c]$ o $[c(t)]$ dependiendo de la forma de amortiguamiento escogida.

7.2.32. ADENED

La subrutina *ADENED* calcula en cada instante de tiempo y utilizando expresiones directas, la energía cinética total de la estructura en cualquiera de los casos de análisis dinámico, o la energía de deformación elástica para el caso de análisis modal cronológico. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.61.

ADENED `MXX(), DXX(), EXX()`

Variable	Tipo	Descripción
`MXX()`	Real	Matriz de masa $[m]$ o matriz de rigidez lateral $[k_\phi]$
`DXX()`	Real	Matriz de velocidades $[\dot{u}_\phi]$ o de desplazamientos de centros de masa $[u_\phi]$
`EXX()`	Real	Vector fila de energía cinética o de energía de deformación elástica, calculado

Tabla 7.61. Sintaxis y argumentos de las instrucciones ADENED

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La subrutina inicializa con la subrutina *MTCONS* el vector fila *EXX()* lleno de ceros. Posteriormente para cada instante de tiempo extrae un vector columna $\{V\}$ de la matriz *DXX()* asociado al mismo instante de tiempo, lo transpone y realiza la operación $0.5\{V\}^T[M]\{V\}$, la cual corresponde a la energía en ese instante.

Cuando la subrutina se utiliza para calcular la energía cinética, la matriz $[M]$ corresponde a la matriz de masa $[m]$ y el vector $\{V\}$ corresponde a las velocidades de los centros de masa $\{\dot{u}_\phi\}$. Cuando se utiliza para calcular la energía de deformación elástica, la matriz $[M]$ corresponde a la matriz de rigidez lateral $[k_\phi]$ y el vector $\{V\}$ corresponde a los desplazamientos de los centros de masa $\{u_\phi\}$.

7.2.33. ADENIN

La subrutina *ADENIN* calcula cualquier componente de energía utilizando integración directa. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.62.

ADENIN `DTIM, FXX(), VXX(), EXX()`

Variable	Tipo	Descripción
`DTIM`	Real	Tamaño del intervalo de tiempo
`FXX()`	Real	Matriz de acciones $[F]$
`VXX()`	Real	Matriz de velocidades $[u]$
`EXX()`	Real	Vector fila de energía en cada instante de tiempo, calculado

Tabla 7.62. Sintaxis y argumentos de las instrucciones ADENIN

La subrutina inicializa con la subrutina *MTCONS* el vector fila *EXX()* lleno de ceros. Posteriormente aproxima de forma numérica el incremento de energía correspondiente a la suma de integrales $\sum_{i=1}^{n}(\int_{t-\Delta t}^{t} F_i v_i dt)$ descrita en el Capítulo 4, en función de las fuerzas y de las velocidades al comienzo y al final del intervalo. Cada incremento se suma a la energía al comienzo del intervalo para obtener la energía al final del intervalo.

La subrutina se utiliza para calcular la energía disipada por amortiguamiento natural y la energía inducida por las acciones externas en cualquiera de los métodos de análisis dinámico en los que se puede verificar la ecuación de balance de energía. Adicionalmente se usa en el método de análisis dinámico inelástico para calcular la energía de deformación total.

7.2.34. ADENSY

La subrutina *ADENSY* crea el vector fila con la energía de deformación inelástica o energía disipada por histéresis en las rótulas plásticas desde $t = 0$ hasta cualquier otro instante de tiempo. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.63.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

ADENSY PRO(), DRO(), FRO(), ESY()

Variable	Tipo	Descripción
PRO()	Real	Tabla de propiedades de rótulas plásticas
DRO()	Real	Tabla de rotaciones de las rótulas
FRO()	Real	Tabla de momentos en las rótulas
ESY()	Real	Vector fila de energía de deformación inelástica, calculado

Tabla 7.63. Sintaxis y argumentos de las instrucciones ADENSY

La subrutina calcula la energía disipada por todas las rótulas plásticas en función de las áreas bajo las curvas respectivas $M - \theta$ como se describió en el Capítulo 5. Las áreas bajo las curvas se evalúan de forma numérica.

7.2.35. ADBAEN

La subrutina *ADBAEN* crea una tabla resumen que almacena todas las componentes de energía mecánica presentes en la estructura y en cada instante de tiempo. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.64.

ADBAEN ECI(), EAM(), ESE(), ESY(), EEX(), VF2(), ENE()

Variable	Tipo	Descripción
ECI()	Real	Vector fila de energía cinética
EAM()	Real	Vector fila de energía disipada por amortiguamiento natural
ESE()	Real	Vector fila de energía de deformación elástica
ESY()	Real	Vector fila de energía de deformación inelástica
EEX()	Real	Vector fila de energía inducida por las acciones externas
VF2()	Real	Vector fila para verificar la energía inducida
ENE()	Real	Tabla resumen de verificación de balance de energía, creada

Tabla 7.64. Sintaxis y argumentos de las instrucciones ADBAEN

En la primera fila se almacena la energía cinética, en la segunda fila se almacena la energía disipada por amortiguamiento natural, en la tercera fila se almacena la energía de deformación elástica, en la cuarta fila se almacena la energía de deformación inelástica, en la quinta fila se almacena la energía de entrada y en la sexta fila se almacena la suma de las 4 primeras componentes de energía, la cual se puede comparar con la energía de entrada para ver el grado de error de los métodos numéricos utilizados en la solución de la ecuación de movimiento. Se utiliza la instrucción *MTADMR* para adjuntar los diferentes vectores fila y así conformar la matriz *ENE()* definitiva.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

7.3. Instrucciones para crear matrices de rigidez y vectores de acciones fijas elementales

7.3.1. KPOR3D

La subrutina *KPOR3D* crea la matriz de rigidez del elemento pórtico espacial en coordenadas locales o globales, considerando o no deformaciones por cortante y conexiones parcialmente rígidas a flexión en los extremos. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.65.

KPOR3D MXX(), XYZ(), ELE(), IELE, PAE(), TISC, IPAU, IRIG

Variable	Tipo	Descripción
MXX()	Real	Matriz de rigidez del elemento, creada
XYZ()	Real	Tabla de coordenadas de nudos y de puntos auxiliares
ELE()	Entero	Tabla de conectividades entre elementos
IELE	Entero	Número del elemento actual
PAE()	Real	Tabla de propiedades de elementos
TISC	Entero	Sistema de coordenadas de la matriz de rigidez (*si TISC=0 entonces sistema local, si TISC=1 entonces sistema global*)
IPAU	Entero	Punto auxiliar del elemento para definición de ejes locales (*si IPAU=0 los ejes locales se definen por defecto*)
IRIG	Entero	Tipo de rigidez considerada en el elemento (si *IRIG=0 se crea una matriz de rigidez básica, si IRIG=1 se consideran deformaciones por cortante, si IRIG=2 se consideran simultáneamente deformaciones por cortante y conexiones parcialmente rígidas*)

Tabla 7.65. Sintaxis y argumentos de las instrucciones KPOR3D

Para cada elemento la subrutina crea la matriz de rigidez básica, sin considerar deformaciones por cortante ni conexiones parcialmente rígidas a flexión en los extremos, en función de los parámetros mecánicos del material, de las propiedades geométricas de la sección y de la longitud del elemento.

Si se escoge el tipo de rigidez *001* se crea una matriz auxiliar con los coeficientes de reducción de rigidez por deformaciones por cortante, la cual se multiplica término a término con la matriz de rigidez básica a través de la subrutina *MTPROD* para formar *MXX()*. Si se escoge el tipo de rigidez *002* se crea otra matriz auxiliar con los coeficientes de reducción por deformaciones por cortante y conexiones parcialmente rígidas a flexión, la cual se multiplica término a término con la matriz de rigidez básica para formar la matriz de rigidez almacenada en *MXX()*.

En todos los casos se utiliza la formulación matemática descrita en el Capítulo 3 para la construcción de la matriz de rigidez elemental en coordenadas locales. Finalmente cuando se escoge el tipo de sistema coordenado *002* (*coordenadas globales*), se utiliza la subrutina *TRPO3D* para formar la matriz de transformación de coordenadas y con la subrutina

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

MTMUL3 se realiza el triple producto requerido para obtener la matriz de rigidez elemental en coordenadas globales.

7.3.2. FFPO3D

La subrutina *FFPO3D* crea el vector de acciones fijas del elemento tipo pórtico espacial en coordenadas locales o globales, considerando o no deformaciones por cortante y conexiones parcialmente rígidas a flexión en los extremos, a partir de las cargas distribuidas aplicadas en la luz del elemento. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.66.

FFPO3D IELE, ELE(), XYZ(), PAE(), TWE(), WEL(), IPAU, TISC, IRIG, FEL()

Variable	Tipo	Descripción
IELE	Entero	Número del elemento actual
ELE()	Entero	Tabla de conectividades entre elementos
XYZ()	Real	Tabla de coordenadas de nudos y de puntos auxiliares
PAE()	Real	Tabla de propiedades de elementos
TWE()	Entero	Tabla de tipos de carga distribuida aplicada a elementos
WEL()	Real	Tabla con valores de carga distribuida aplicada a elementos
IPAU	Entero	Punto auxiliar del elemento para definición de ejes locales (*si IPAU=0 los ejes locales se definen por defecto*)
TISC	Entero	Sistema de coordenadas de la matriz de rigidez (*si TISC=0 entonces sistema local, si TISC=1 entonces sistema global*)
IRIG	Entero	Tipo de rigidez considerada en el elemento (si *IRIG=0 se crea una matriz de rigidez básica, si IRIG=1 se consideran deformaciones por cortante, si IRIG=2 se consideran simultáneamente deformaciones por cortante y conexiones parcialmente rígidas*)
FEL()	Real	Vector de acciones fijas del elemento, creado

Tabla 7.66. Sintaxis y argumentos de las instrucciones FFPO3D

En la subrutina inicialmente se calculan las constantes de deformaciones por cortante y conexiones parcialmente rígidas a flexión en los extremos, para cualquiera de los tipos de carga distribuida disponibles (*rectangular o triangular en direcciones y o z local*). Luego, para cada una de las cargas aplicadas a los elementos, se identifica el tipo y el valor numérico de la carga a partir de la información almacenada en las tablas *TWE()* y *WEL()*, y se calculan las acciones fijas en coordenadas locales producidas por cada tipo de carga. Cuando más de un tipo de carga se aplica a un mismo elemento, las acciones fijas se acumulan de forma que al final el vector de acciones fijas almacene el aporte de todas las cargas aplicadas al elemento.

Finalmente cuando se escoge el tipo de sistema coordenado *002* (*coordenadas globales*), se utiliza la subrutina *TRPO3D* para formar la matriz de transformación de coordenadas y con la subrutina *MTMUL3* se realiza el triple producto requerido para obtener la matriz de rigidez elemental en coordenadas globales, similar a la subrutina *KPOR3D*.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

7.4. Instrucciones para calcular modos de vibración

7.4.1. VPJACO

La subrutina *VPJACO* resuelve el problema de valores propios, cuya solución corresponde a las frecuencias de vibración (*al cuadrado*) y a los modos de vibración natural de la estructura. Los modos de vibración obtenidos son ortonormalizados (*o normalizados respecto a la matriz de masa*) y están ordenados en forma ascendente respecto a las frecuencias de vibración, de forma que el primer modo, que se almacena en la primera columna de $[\Phi]$, le corresponde la frecuencia de vibración más pequeña. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.67.

VPJACO `KDP(), MDR(), VEP(), VAP(), ICIC, TOL`

Variable	Tipo	Descripción
`KDP()`	Real	Matriz de rigidez lateral $[k_\phi]$
`MDR()`	Entero	Matriz de masa $[m]$
`VEP()`	Real	Vector de frecuencias de vibración al cuadrado (*valores propios*), calculado
`VAP()`	Real	Matriz de modos de vibración ortonormalizados (*vectores propios*), calculada
`ICIC`	Entero	Número de ciclos iterativos realizados
`TOL`	Real	Tolerancia especificada para convergencia de valores propios (*por defecto 0.01*)

Tabla 7.67. Sintaxis y argumentos de las instrucciones VPJACO

La subrutina sigue la formulación del método de Jacobi generalizado presentada en el Capítulo 3 para la solución del problema de valores propios. Esta subrutina es original de PEFiCA, la única modificación que se le realizó fue la parte en la que los modos de vibración se organizan en forma ascendente, lo cual se realiza mediante la subrutina *ORVAPR*.

7.5. Instrucciones para realizar operaciones matriciales básicas

7.5.1. MTDIAV

La subrutina *MTDIAV* crea una matriz diagonal con los valores de un vector fila o un vector columna. El vector de datos y las matriz diagonal son de tipo real. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.68.

MTDIAV `VXX(), MXX()`

Variable	Tipo	Descripción
`VXX()`	Real	Vector fila o columna
`MXX()`	Real	Matriz diagonal, creada

Tabla 7.68. Sintaxis y argumentos de las instrucciones MTDIAV

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La subrutina inicializa la matriz cuadrada del mismo tamaño que el vector fila o columna. Posteriormente en cada celda de la diagonal principal almacena el valor correspondiente de cada celda del vector.

7.5.2. MTADVM

La subrutina *MTADVM* crea una matriz resumen adjuntando 5 matrices individuales, una al lado de la otra. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.69.

MTADVM `MX1(), MX2(), MX3(), MX4(), MX5(), MXR()`

Variable	Tipo	Descripción
`MX1()`	Real	Submatriz N° 1
`MX2()`	Real	Submatriz N° 2
`MX3()`	Real	Submatriz N° 3
`MX4()`	Real	Submatriz N° 4
`MX5()`	Real	Submatriz N° 5
`MXR()`	Real	Matriz resumen, creada

Tabla 7.69. Sintaxis y argumentos de las instrucciones MTADVM

La subrutina utiliza varias veces la subrutina *MTADMR* para crear la matriz resumen *MXR()* buscada. Las matrices pueden tener diferente número de columnas pero todas deben tener el mismo número de filas.

7.5.3. MTREMV

La subrutina *MTREMV* reemplaza los valores de una fila o de una columna de una matriz real existente con los valores de un vector real. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.70.

MTREMV `MXX(), VXX(), TIPO, IDFC`

Variable	Tipo	Descripción
`MXX()`	Real	Matriz existente, modificada
`VXX()`	Real	Vector fila o columna a reemplazar
`TIPO`	Real	Tipo de vector a reemplazar (*"FILA" o "COLUMNA"*)
`IDFC`	Real	Número de fila o de columna a reemplazar

Tabla 7.70. Sintaxis y argumentos de las instrucciones MTREMV

La subrutina sobrescribe adecuadamente los coeficientes de la matriz existente, manteniendo los demás coeficientes sin modificaciones. El número de filas del vector columna o el número de columnas del vector fila deben coincidir respectivamente con el número de filas o de columnas de la matriz a reemplazar.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

7.5.4. MTREMP

La subrutina *MTREMP* reemplaza el contenido de una submatriz real dentro de una matriz real existente. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.71.

MTREMP MXX(), MSS(), POSF, POSC, NFIL, NCOL

Variable	Tipo	Descripción
MXX()	Real	Matriz existente, modificada
MSS()	Real	Submatriz a reemplazar
POSF	Entero	Fila en la que se almacenará el termino (1,1) de la submatriz
POSC	Entero	Columna en la que se almacenará el termino (1,1) de la submatriz
NFIL	Entero	Número de filas de la submatriz
NCOL	Entero	Número de columnas de la submatriz

Tabla 7.71. Sintaxis y argumentos de las instrucciones MTREMP

La subrutina reemplaza el contenido de la submatriz manteniendo los demás coeficientes de la matriz original sin modificaciones. Se debe tener en cuenta que la submatriz debe quedar siempre dentro de la matriz principal, sin que queden datos por fuera.

7.5.5. MTCPCO

La subrutina *MTCPCO* copia los datos de una columna de una matriz real existente en la siguiente columna de la misma matriz. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.72.

MTCPCO ICOL, MXX()

Variable	Tipo	Descripción
ICOL	Entero	Número de la columna a copiar
MXX()	Real	Matriz existente, modificada

Tabla 7.72. Sintaxis y argumentos de las instrucciones MTCPCO

La subrutina extrae la columna $ICOL$ de la matriz *MXX()* y la reemplaza en la columna $ICOL + 1$ de la misma matriz, manteniendo los demás coeficientes de la matriz inalterados. Para ello utiliza las subrutinas *MTSUBM* y *MTREMV*.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

7.6. Instrucciones para numeración de grados de libertad

7.6.1. NGLNCD

La subrutina *NGLNCD* crea la tabla de numeración de grados de libertad por nudo preparada para el proceso de condensación estructural. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.73.

NGLNCO `MGL(), MRE(), NGLI, DRX(), XYZ()`

Variable	Tipo	Descripción
`MGL()`	Entero	Tabla de numeración de grados de libertad, creada
`MRE()`	Entero	Tabla de clasificación de grados de libertad
`NGLI`	Entero	Número de grados de libertad de interés
`DRX()`	Real	Tabla de coordenadas de centros de masa
`XYZ()`	Real	Tabla de coordenadas de nudos y de puntos auxiliares

Tabla 7.73. Sintaxis y argumentos de las instrucciones NGLNCO

La numeración de los grados de libertad se realiza comenzando por los grados de libertad de interés, siguiendo por los grados de libertad condensados y terminando por los grados de libertad conocidos. Adicionalmente, los grados de libertad de interés de los nudos asociados a un mismo diafragma rígido se numeran de forma consecutiva y en el orden en que definieron los diafragmas rígidos en las tablas *DRX()* y *MAS()*, lo cual facilita la construcción de la matriz geométrica $[A]$ y controla la presencia de errores al asociar los desplazamientos de interés de un nudo con los desplazamientos de un diafragma incorrecto.

7.6.2. NGLROT

La subrutina *NGLROT* crea la tabla de numeración de rótulas plásticas y calcula el número total de rótulas plásticas de la estructura. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.74.

NGLROT `ARE(), NRO(), ROTE`

Variable	Tipo	Descripción
`ARE()`	Entero	Tabla de asignación de rótulas plásticas tipo
`NRO()`	Entero	Tabla de numeración de rótulas plásticas, creada
`ROTE`	Entero	Número de rótulas plásticas de la estructura, calculado

Tabla 7.74. Sintaxis y argumentos de las instrucciones NGLROT

La subrutina inicializa la matriz entera *NRO()* llena de ceros con la subrutina *PBDEMI*, y coloca un indicador en las mismas posiciones en que la matriz *ARE()* tiene un coeficiente diferente de cero, numerando primero las rótulas de un mismo elemento. El último indicador colocado corresponde al número total de rótulas de la estructura *ROTE*.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

7.7. Instrucciones para organización de matrices

7.7.1. ORNDIR

La subrutina *ORNDIR* crea las tablas de cantidad y de indicadores de nudos asociados a cada diafragma rígido de la estructura. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.75.

ORNDIR DRX(), XYZ(), MX(), VX()

Variable	Tipo	Descripción
DRX()	Real	Tabla de coordenadas de centros de masa
XYZ()	Real	Tabla de coordenadas de nudos y de puntos auxiliares
MX()	Entero	Tabla de nudos asociados a diafragmas, creada
VX()	Entero	Vector columna de cantidad de nudos asociados a diafragmas, creado

Tabla 7.75. Sintaxis y argumentos de las instrucciones ORNDIR

La subrutina identifica que nudos pertenecen a un diafragma comparando la coordenada z del nudo con la coordenada z del centro de masa del diafragma. Cada nudo asociado a un mismo diafragma lo almacena en una fila de la matriz *MX()* mientras que el número total de nudos asociados al mismo diafragma lo almacena en cada fila del vector columna *VX()*.

7.7.2. ORECOL

Con la subrutina *ORECOL* se crean las tablas de nudos y de elementos asociados a cada eje de columnas, denominadas *NEC(), NNE()* y *COL()*, en función de las tablas de coordenadas de los nudos, de los centros de masa y de la tabla de conectividades. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.76.

ORECOL NEC(), NNE(), COL(), CXY(), XYZ(), ELE(), NELE

Variable	Tipo	Descripción
NEC()	Entero	Tabla de nudos asociados a cada eje de columnas, creada
NNE()	Entero	Tabla de cantidad de nudos asociados a cada eje de columnas, creada
COL()	Entero	Tabla de elementos asociados a cada eje de columnas, creada
CXY()	Real	Tabla de coordenadas X,Y de ejes de columnas
XYZ()	Real	Tabla de coordenadas de nudos y de puntos auxiliares
ELE()	Entero	Tabla de conectividades entre elementos
NELE	Entero	Número de elementos

Tabla 7.76. Sintaxis y argumentos de las instrucciones ORECOL

Esta subrutina busca los nudos con las mismas coordenadas x, y de los ejes de columnas, los cuenta y los organiza según su coordenada z, almacenándolos el cada fila de la ta-

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

bla *NEC()*. Adicionalmente busca los elementos asociados a estos nudos y los organiza de forma secuencial en la tabla *COL()*.

7.7.3. ORDIRE

La subrutina *ORDIRE* crea tres tablas de datos, las primeras dos con los desplazamientos de los centros de masa en direcciones x e y, y la tercera con las rotaciones de los diafragmas en dirección z global, a partir de la matriz de desplazamientos de centros de masa $[u_\phi]$. En cada tabla creada los datos se presentan en el orden en que se numeraron los diafragmas (*la primera fila corresponde al primer diafragma*). La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.77.

ORDIRE NDIM, MXX(), DRX(), XCR(), YCR(), ZCR()

Variable	Tipo	Descripción
NDIM	Entero	Número de dimensiones de la estructura
MXX()	Real	Matriz de desplazamientos de centros de masa
DRX()	Real	Tabla de coordenadas de centros de masa
XCR()	Real	Tabla de desplazamientos de centros de masa en dirección X, creada
YCR()	Real	Tabla de desplazamientos de centros de masa en dirección Y, creada
ZCR()	Real	Tabla de rotaciones de diafragmas en dirección Z, creada

Tabla 7.77. Sintaxis y argumentos de las instrucciones ORDIRE

La subrutina organiza los datos de la matriz *MXX()* en el orden en que se numeraron los diafragmas. Se crea la matriz *XCR()* independientemente si la estructura es plana o espacial, mientras que las matrices *YCR()* y *ZCR()* se crean solamente si el pórtico es espacial.

7.7.4. ORMPRO

La subrutina *ORMPRO* actualiza la tabla de propiedades de elementos *PAE()* con las rigideces tangente de las rótulas plásticas asociadas a los elementos de la estructura, evaluadas al comienzo del intervalo. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.78.

ORMPRO ITIM, NRO(), KRO(), PAE()

Variable	Tipo	Descripción
ITIM	Entero	Intervalo de tiempo actual
NRO()	Entero	Tabla de numeración de rótulas plásticas
KRO()	Real	Tabla de rigidez tangente de rótulas plásticas
PAE()	Real	Tabla de propiedades de elementos, actualizada

Tabla 7.78. Sintaxis y argumentos de las instrucciones ORMPRO

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La subrutina reemplaza los valores de las rigideces de las conexiones parciales de los extremos de los elementos, con las rigideces tangentes de las rótulas calculadas al comienzo del intervalo de tiempo especificado. La tabla *PAE()* actualizada se utiliza en la formación de la matriz de rigidez tangente.

7.7.5. ORVAPR

La subrutina *ORVAPR* organiza los valores y los vectores propios calculados con el método de Jacobi generalizado, programado en la subrutina *VPJACO*. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.79.

ORVAPR `VAP(), VEP()`

Variable	Tipo	Descripción
`VAP()`	Real	Vector columna de valores propios, actualizada
`VEP()`	Real	Matriz de vectores propios, actualizada

Tabla 7.79. Sintaxis y argumentos de las instrucciones ORVAPR

Los valores y los vectores propios se organizan en forma ascendente, de forma que al primer vector propio le corresponda el menor valor propio. El primer vector propio queda almacenado en la primera columna de la matriz *VEP()* y el primer valor propio queda almacenado en la primera celda del vector *VAP()*.

7.8. Instrucciones para cambios de base

7.8.1. TRPO3D

La subrutina *TRPO3D* construye la matriz de transformación de coordenadas del elemento tipo pórtico espacial, en función de las coordenadas de los nudos del elemento y de la forma escogida para orientar los ejes locales del elemento. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.80.

TRPO3D `MX(), XYZ(), ELE(), IELE, IPAU`

Variable	Tipo	Descripción
`MX()`	Real	Matriz de transformación de coordenadas, creada
`XYZ()`	Real	Tabla de coordenadas de nudos y de puntos auxiliares
`ELE()`	Entero	Tabla de conectividades
`IELE`	Entero	Elemento actual
`IPAU`	Entero	Punto auxiliar utilizado para definir ejes locales (*opcional*)

Tabla 7.80. Sintaxis y argumentos de las instrucciones TRPO3D

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

En la subrutina se han implementado los dos criterios descritos en el Capítulo 3 para definir la orientación de los ejes locales $\bar{y}$ y $\bar{z}$ del elemento. Si $IPAU \neq 0$ se utiliza el criterio del punto auxiliar (*en este caso el punto auxiliar corresponde al valor de* $IPAU$), mientras que si $IPAU = 0$ se utiliza la forma por defecto de definir la orientación de los ejes locales.

7.8.2. TRDIRZ

La subrutina *TRDIRZ* crea la matriz de asociación de desplazamientos (*o matriz geométrica*) $[A]$ a partir de las coordenadas de los nudos y de los centros de masa, y de las tablas de nudos asociados a diafragmas. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.81.

TRDIRZ `MX(), XYZ(), DRX(), NDR(), CND(), NGDR`

Variable	Tipo	Descripción
`MX()`	Real	Matriz de asociación de desplazamientos, creada
`XYZ()`	Real	Tabla de coordenadas de nudos y de puntos auxiliares
`DRX()`	Real	Tabla de coordenadas de centros de masa
`NDR()`	Entero	Tabla de nudos asociados a cada diafragma rígido
`CND()`	Entero	Tabla de cantidad de nudos asociados a cada diafragma rígido
`NGDR`	Entero	Número de grados de libertad por diafragma rígido

Tabla 7.81. Sintaxis y argumentos de las instrucciones TRDIRZ

La subrutina utiliza la formulación matemática descrita en el Capítulo 3, en la parte de asociación de grados de libertad.

7.9. Instrucciones para importar y exportar matrices

7.9.1. IOIMPR

La subrutina *IOIMPR* importa una matriz real de un archivo de texto separado por espacios, y la almacena en la matriz real *MXX()*. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.82.

IOIMPR `MXX(), FILE, NFIL, NCOL`

Variable	Tipo	Descripción
`MXX()`	Real	Matriz importada
`FILE`	Cadena	Nombre del archivo de texto
`NFIL`	Entero	Número de filas de la matriz *MXX()*
`NCOL`	Entero	Número de columnas de la matriz *MXX()*

Tabla 7.82. Sintaxis y argumentos de las instrucciones IOIMPR

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

La subrutina importa los datos del archivo de texto de nombre $FILE.txt$, que debe almacenarse en el mismo directorio donde se almacena el libro de cálculo ejecutado. Los datos se importan por filas y se separan por columnas para conformar la matriz *MXX()*.

7.9.2. IOEXMR

La subrutina *IOEXMR* crea y exporta una tabla real de resultados en función del tiempo, a partir de una matriz de datos en las que cada fila corresponde a un parámetro y cada columna corresponde a un instante de tiempo. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.83.

IOEXPR MXX(), TME(), NOMB

Variable	Tipo	Descripción
MXX()	Real	Matriz resumen exportada
TME()	Real	Vector columna de instantes de tiempo considerados en el análisis
NOMB	Cadena	Nombre del archivo de texto

Tabla 7.83. Sintaxis y argumentos de las instrucciones IOEXPR

La subrutina transpone la matriz de datos *MXX()* con *MTTRAN*, y la adjunta al vector de instantes de tiempo *TME()* para formar la matriz que posteriormente exporta con *IOEXPR* a un archivo de texto en el que los datos quedan separados por espacios. La primera columna del archivo contiene los instantes de tiempo y las columnas restantes contienen la variación de cada parámetro.

7.9.3. IOEXDI

La subrutina *IOEXDI* exporta parte o la totalidad de las tablas de acciones internas de cada eje de vigas y de columnas, dependiendo si la estructura analizada es plana o espacial. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.84.

IOEXDI IEJE, NDIM, TIPO, NXX(), VYY(), VZZ(), TXX(), MYY(), MZZ()

Variable	Tipo	Descripción
IEJE	Entero	Matriz resumen exportada
NDIM	Real	Vector columna de instantes de tiempo considerados en el análisis
TIPO	Cadena	Nombre del archivo de texto
NXX()	Real	Tabla de diagramas de fuerza axial, exportada
VYY()	Real	Tabla de diagramas de fuerza cortante en dirección y local, exportada
VZZ()	Real	Tabla de diagramas de fuerza cortante en dirección z local, exportada
TXX()	Real	Tabla de diagramas de momento torsor, exportada
MYY()	Real	Tabla de diagramas de momento flector en dirección y local, exportada
MZZ()	Real	Tabla de diagramas de momento flector en dirección z local, exportada

Tabla 7.84. Sintaxis y argumentos de las instrucciones IOEXDI

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013

Internamente se utiliza la subrutina principal *IOEXPR* para exportar cada tabla de datos. Si la estructura es plana la subrutina exporta solamente las tablas *NXX(), VYY() y MZZ()*, en caso contrario la subrutina exporta las seis tablas de acciones internas.

7.10. Instrucciones internas de uso común

7.10.1. PBELRP

La subrutina *PBELRP* identifica si un elemento posee rótulas plásticas asociadas. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.85.

PBELRP `IELE, NRO(), ROTU`

Variable	Tipo	Descripción
`IEJE`	Entero	Matriz resumen exportada
`NRO()`	Entero	Tabla de numeración de rótulas plásticas
`ROTU`	Cadena	Indicador de rótulas en el elemento (*"SI" o "NO"*)

Tabla 7.85. Sintaxis y argumentos de las instrucciones PBELRP

La subrutina entrega un valor de "*SI*" en la variable *ROTU* cuando encuentra un indicador diferente de cero en la fila de la matriz *NRO()* correspondiente al elemento en estudio, o entrega el valor de "*NO*" cuando no encuentra un indicador diferente de cero.

7.10.2. PBRKRO

La subrutina *PBRKRO* identifica si hay cambios apreciables en la rigidez tangente de las rótulas plásticas de un intervalo de tiempo a otro. La sintaxis de la instrucción y la descripción de los parámetros se muestran en la Tabla 7.86.

PBRKRO `IELE, NRO(), ROTU`

Variable	Tipo	Descripción
`KRO()`	Real	Matriz de rigidez tangente de rótulas plásticas
`ITIM`	Entero	Intervalo de tiempo actual
`CROT`	Cadena	Indicador de cambio apreciable en la rigidez de las rótulas (*"SI" o "NO"*)

Tabla 7.86. Sintaxis y argumentos de las instrucciones PBRKRO

La subrutina compara las rigideces tangentes de las rótulas pláticas de la estructura, obtenidas para dos instantes de tiempo consecutivos. Si en alguna de las rótulas encuentra que la diferencia de rigideces es superior a una cierta tolerancia (*0.01 por defecto*), entrega como resultado un "*SI*" en la variable *CROT*. De lo contrario entrega como resultado un "*NO*" en la misma variable.

© Francisco Leonardo Noy Hilarión – Universidad Nacional de Colombia, 2013