

24
2er.



UNIVERSIDAD NACIONAL
AVENIDA DE
MEXICO

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
CAMPUS "ACATLAN"

ANALISIS DINAMICO PASO A PASO DE
ESTADOS PLANOS POR EL METODO DEL
ELEMENTO FINITO

T E S I S

QUE PARA OBTENER EL TITULO DE
LICENCIADO EN MATEMATICAS
APLICADAS Y COMPUTACION
P R E S E N T A N :
ALICIA GARCIA MENDOZA
LUIS FERNANDO MAGAÑA JAIME



ASESOR: M.I. VICTOR JOSE PALENCIA GOMEZ

ACATLAN, ESTADO DE MEXICO

1997

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

GRACIAS

A *Dios*: Por estar cerca de mí siempre y a quien simplemente le debo todo.

A mis Padres "*Joaquín y Celia*": Por su incondicional apoyo, ternura y sustento. Por recibir un consejo en el momento oportuno; por haber sembrado en mí el deseo de buscar y crecer en todo momento; y sobre todo por haberme enseñado y brindado la oportunidad de escoger mi propio destino.

A mis hermanas "*Lupita, Angie, Ley, Magia y Mary*": Por haber encontrado apoyo incondicional y con quienes he compartido los mejores momentos de mi vida de estudiante ya que tienen la fortaleza suficiente para lograrlo todo en la vida.

Al "*M.I. Víctor José Palencia Gómez y M.I. Fernando Vera Badila*": Por el apoyo brindado, el tiempo dedicado y los conocimientos transmitidos.

A mi compañero y amigo "*Luis*": Por todo el apoyo, comprensión y amistad brindada durante la elaboración de esta tesis. Por haber defendido este proyecto aún con todos los obstáculos encontrados en el camino; ya que no ha sido fácil llegar a la meta.

A mis amigos y compañeros de carrera: Por que con su amistad, apoyo y comprensión lograron convertir esta etapa en una experiencia muy agradable.

A todos aquellos amigos que siempre han compartido mis logros y de los cuales he recibido un apoyo incondicional en todo momento.

A "*Poli*": Por el apoyo, comprensión, paciencia y sobretodo por el cariño y ternura brindada para culminar esta meta. Por creer siempre en mí.

G R A C I A S

A DIOS:

POR DEJARME LLEGAR A ESTE MOMENTO.

A MI MADRE:

POR DARMER LA VIDA Y ENSEÑARME EL CAMINO CORRECTO.

A MI ESPOSA:

POR SU CARÍÑO, APOYO Y COMPRESION.

A MIS HERMANOS:

POR PREOCUPARSE POR MI.

AL M.I. VICTOR J. PALENCIA GÓMEZ:

POR LA ACERTADA DIRECCION DE ESTE TRABAJO.

AL M.I. FERNANDO VERA BADILLO:

POR HABER SUGERIDO EL TEMA.

A LOS SÍNODALES:

POR EL TIEMPO DEDICADO A LA REVISION DE LA TESIS.

A MIS AMIGOS ALICIA, ALEJANDRO, ASYADETH, BEATRIZ, EDUARDO,
GABRIEL, JUAN JOSE, LETICIA, POLICARPO Y RICARDO:

POR DARMER SU AMISTAD INCONDICIONAL Y DEMOSTRAR SU
INTERES HACIA MI.

L U I S

INDICE

INTRODUCCION	1
CAPITULO 1	
ECUACION DE EQUILIBRIO PARA EL ESTADO DE ESFUERZOS PLANOS Y SU SOLUCION POR EL METODO DEL ELEMENTO FINITO	3
1.1 Ecuación de equilibrio del medio continuo	3
1.2 Discretización por elemento finito	7
1.3 Ecuaciones de forma	8
CAPITULO 2	
OBTENCION DE LAS MATRICES DE MASAS Y RIGIDECES DEL CONTINUO	15
2.1 Introducción	15
2.2 Ensamble de la matriz de rigideces de la estructura	16
2.3 Ensamble de la matriz de masas de la estructura	18
CAPITULO 3	
METODO β DE NEWMARK	19
3.1 Método generalizado	22
3.2 Algoritmo numérico	25
CAPITULO 4	
CRITERIOS DE PROGRAMACION Y SISTEMA PARA EL ANALISIS	27
4.1 Aspectos generales	27
4.1.1 Semblanza histórica	27
4.1.2 Jerarquía de los lenguajes	27
4.2 Criterios de programación	29
4.2.1 Formas de programar	29
4.2.2 Programas de aplicación	29
4.3 Visual BASIC	30
4.3.1 Historia	30
4.3.2 Estructura	30
4.3.3 Aplicaciones	31
4.3.4 Ventajas y desventajas de Visual BASIC	31
4.4 FORTRAN	32
4.4.1 Historia	32
4.4.2 Estructura	32
4.4.3 Aplicaciones	32
4.4.4 Ventajas y desventajas	32

4.5 Organización del sistema	33
4.5.1 Organización	33
4.5.2 Proceso de solución	34

CAPITULO 5

PREPROCESADOR	35
5.1 Generalidades	35
5.2 Organización del programa	35
5.3 Manual del usuario	38

CAPITULO 6

PROCESADOR 1	46
6.1 Generalidades	46
6.2 Organización	46
6.3 Manual del Usuario	48

CAPITULO 7

PROCESADOR 2	51
7.1 Generalidades	51
7.2 Organización	52
7.3 Manual del usuario	53
7.3.1 Entrada de datos	55
7.3.2 Graficación	55
7.3.3 Impresión	57
7.3.4 Salida	57

CAPITULO 8

PROCESADOR 3	58
8.1 Generalidades	58
8.2 Organización	59
8.3 Manual del Usuario	60

EJEMPLO DE APLICACION

9.1 Estructura ejemplo	64
9.2 Proceso de cálculo	64
9.3 Resultados	69
9.4 Refinado de malla (Caso 2)	70

CONCLUSIONES Y RECOMENDACIONES	72
--------------------------------------	----

BIBLIOGRAFIA	76
--------------------	----

APENDICE A	77
APENDICE B	107
APENDICE C	122

INTRODUCCIÓN

El proceso de análisis dinámico de estructuras ha tenido un cambio significativo hasta hace unos pocos años con el desarrollo de las computadoras.

Inicialmente los procesos de cálculo consideraban únicamente el uso de calculadoras, por lo que los modelos matemáticos para resolver estructuras estaban muy limitados por los mismos y únicamente empresas grandes o Centros de Investigación tenían acceso a computadoras main-frame, junto con programas tipo Caja-Negra que de alguna manera generaban dependencia tecnológica para su uso.

Con el pasar del tiempo, el avance en la producción de computadoras ha permitido no sólo aumentar considerablemente la capacidad de las mismas, sino también su costo por lo que actualmente, es relativamente sencillo adquirir una computadora personal en el ámbito profesional.

Esto ha dado como resultado varios cambios que son:

- a) La forma de trabajar para el análisis de estructuras.
- b) Se han podido resolver varios modelos matemáticos, lo que ha permitido nuevas opciones para el análisis de estructuras.
- c) Se han creado nuevas necesidades, sobre todo lo relacionado con el desarrollo de software para ser utilizado en forma confiable y adecuada.

Por lo dicho anteriormente, es incuestionable la necesidad de elaborar software técnico, así como estrategias que permitan perfeccionar el desarrollo del mismo.

Con base en lo anterior, este trabajo pretende desarrollar un sistema eficiente no únicamente en el aspecto de programación sino ergonómico que permita analizar medios continuos ante cargas sísmicas mediante el procedimiento del Método del Elemento Finito.

Para lograr dicho objetivo, se requiere por un lado tener conocimientos en modelos matemáticos, análisis de estructuras, así como en computación, y por otro lado conocer las necesidades de los especialistas en el estudio de las estructuras; también se consideró realizar un sistema estructurado de tal manera que sea flexible y se adapte a las necesidades futuras del usuario.

En el capítulo 1 se presenta la ecuación variacional del principio del trabajo virtual y su desarrollo para obtener las ecuaciones de equilibrio dinámico de una estructura, en un contexto lineal, isótropo y homogéneo.

En el capítulo 2 se explica cómo se calcula la matriz de rigideces de un elemento finito: cuadrilátero de 8 puntos de la familia Serendipity mediante cuadratura Gaussiana y la discretización para calcular la correspondiente matriz de masas de ese mismo elemento de establecer mediante el proceso de ensamblaje de las matrices anteriores para cada elemento, la matriz de masas y rigideces de la estructura.

En el capítulo 3 se presenta la fundamentación y se desarrolla la ecuación del método β de Newmark para el análisis dinámico de estructuras y se explica lo relacionado con la discretización de acelerogramas que se utilizarán como excitación del sistema.

En el capítulo 4 se presentan los criterios de programación para el desarrollo del sistema, así como la fundamentación en relación a su eficiencia con otros compiladores y se explica con detalle la forma como está estructurado el sistema de análisis dinámico, así como su utilización.

En el capítulo 5 se presentan las características del programa Preprocesador que forma el archivo de datos del sistema.

En el capítulo 6 se presentan las características del programa Procesador 1 que calcula las matrices de rigideces y masas de la estructura.

En el capítulo 7 se presentan las características del programa Procesador 2 que procesa los acelerogramas del D. F.

En el capítulo 8 se presentan las características del programa Procesador 3 que efectúa el método β de Newmark.

En el capítulo 9 se exponen dos ejemplos numéricos con diferentes datos de un mismo medio continuo y se presentan los resultados.

1. ECUACIÓN DE EQUILIBRIO PARA EL ESTADO DE ESFUERZOS PLANOS Y SU SOLUCIÓN POR EL MÉTODO DEL ELEMENTO FINITO.

El método del elemento finito en términos generales es un método para resolver ecuaciones diferenciales de problemas en la frontera o de valores en la frontera e iniciales.

En la mecánica estructural, el método del elemento finito es una extensión de los métodos matriciales (estructuras esqueléticas) para analizar un medio continuo, el cual se discretiza como una estructura formada por un número de elementos interconectados entre sí en los puntos nodales como se indica en la figura 1.1. Este método resulta muy útil cuando la geometría del continuo es compleja.

Para plantear las ecuaciones de equilibrio de la teoría de la elasticidad (estado plano de esfuerzos) y proponer su solución por el método del elemento finito, se pueden seguir dos caminos, uno es mediante el método de los residuos pesados en donde se demuestra que el principio del trabajo virtual es una forma débil de las ecuaciones de equilibrio y mediante una aproximación de Galerkin se obtienen las ecuaciones del método del elemento finito que se utilizan para la solución. Otro criterio es por los métodos variacionales, el que a continuación se expone. Por ambos caminos se llega a las mismas ecuaciones.

1.1 ECUACIONES DE EQUILIBRIO DEL MEDIO CONTINUO.

En la teoría de la elasticidad lineal la ecuación variacional viene dada por el principio del trabajo virtual, el cual se obtiene de las ecuaciones de Cauchy del movimiento. En forma esquemática el enunciado del principio es:

$$\left(\begin{array}{c} \text{FUERZAS} \\ \text{INTERNAS} \end{array} \right) + \left(\begin{array}{c} \text{FUERZAS DE} \\ \text{INERCIA} \end{array} \right) = \left(\begin{array}{c} \text{FUERZAS DE} \\ \text{SUPERFICIE} \end{array} \right) + \left(\begin{array}{c} \text{FUERZAS DE} \\ \text{CUERPO} \end{array} \right)$$

esto en notación matricial es:

$$\int_{\Omega} \delta \underline{\epsilon}' \underline{\sigma} d\Omega + \int_{\Omega} \rho \delta \underline{U}' \underline{\dot{U}} d\Omega = \int_{\Omega} \delta \underline{U}' \underline{\sigma} d\Gamma + \int_{\Omega} \rho \delta \underline{U}' \underline{f} d\Omega \quad (1.1)$$

en donde:

$$\begin{array}{ll} \Omega & = \text{Punto interior de la estructura.} \\ \Gamma & = \text{Condiciones de frontera} \end{array}$$

- δ = Primera variación que opera sobre las cantidades que proceden
 $\underline{\sigma}_{(n)}$ = Componentes de las cargas que actúan por unidad de superficie.
 \underline{f} = Vector de cargas de cuerpo.
 $\underline{\varepsilon}$ = Componentes del vector de deformación.
 $\underline{\sigma}$ = Componentes del tensor de esfuerzos.
 \underline{U} = Vector de desplazamiento.
 \underline{P} = Densidad de masa por unidad de volumen.
 $\underline{\ddot{U}}$ = Vector de aceleraciones.

Al considerar una solución aproximada del campo de desplazamientos:

$$\underline{U} \approx \underline{\bar{U}} \quad (1.2)$$

en donde $\underline{\bar{U}}$ es el vector solución aproximada de desplazamiento.

La solución aproximada se propone mediante funciones de forma y se tiene:

$$\underline{U} \approx \underline{\bar{U}} = \underline{N}\underline{U} \quad (1.3)$$

en donde $\underline{\bar{U}}$ resultan los desplazamientos que se van a obtener en la solución y \underline{N} son las funciones de forma. De acuerdo a la teoría de la elasticidad se pueden definir las siguientes ecuaciones:

$$\underline{\varepsilon} = \underline{L}\underline{U} \quad (1.4)$$

$$\underline{\sigma} = \underline{D}\underline{\varepsilon} \quad (1.5)$$

en donde $\underline{\varepsilon}$ y $\underline{\sigma}$ están definidas en la ecuación (1.1), \underline{L} es una matriz operacional ecuación (1.6) y \underline{D} es una matriz de coeficientes elásticos para el caso de estado plano de esfuerzos ecuación (1.7)

$$\underline{L} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \quad (1.6)$$

$$\underline{D} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1-\nu) \end{bmatrix} \quad (1.7)$$

donde:

E = Módulo de elasticidad

ν = Relación de Poisson

al considerar la solución aproximada se sustituye la ecuación (1.3) en las ecuaciones (1.4) y (1.5)

$$\underline{\epsilon} = \underline{LN}\bar{U} \quad (1.8)$$

$$\underline{\sigma} = \underline{DLN}\bar{U} \quad (1.9)$$

se define:

$$\underline{B} = \underline{LN} \quad (1.10)$$

al sustituir la ecuación (1.1) en las ecuaciones (1.8) y (1.9)

$$\underline{\sigma} = \underline{DB}\bar{U} \quad (1.11)$$

$$\underline{\epsilon} = \underline{B}\bar{U} \quad (1.12)$$

al considerar la primera variación con los desplazamientos aproximados se tiene:

$$\delta\underline{U} \approx d\underline{U} = \underline{N}\delta\bar{U} \quad (1.13)$$

$$\delta\underline{U}' \approx d\underline{U}' = \delta\underline{U}'\underline{N}' \quad (1.14)$$

al sustituir las ecuaciones (1.11) a (1.14) en (1.1) se tiene:

$$\int_{\Omega} \delta \bar{U}' B' DB \bar{U} d\Omega + \int_{\Omega} \rho \delta \bar{U}' N \bar{U} d\Omega = \int_{\Gamma} \delta \bar{U}' N' \sigma_{(n)} d\Gamma + \int_{\Omega} \delta \bar{U}' N' f d\Omega \quad (1.15)$$

al factorizar y simplificar:

$$\int_{\Omega} B' DB \bar{U} d\Omega + \int_{\Omega} \rho N' N \bar{U} d\Omega = \int_{\Gamma} N' \sigma_{(n)} d\Gamma + \int_{\Omega} \rho N' f d\Omega \quad (1.16)$$

la ecuación (1.16) es válida para toda la región del continuo ($\Omega + \Gamma$)

al reordenar los términos se tiene:

$$\int_{\Omega} B' DB d\Omega \bar{U} + \int_{\Omega} \rho N' N d\Omega \bar{U} = \int_{\Gamma} N' \sigma_{(n)} d\Gamma + \int_{\Omega} \rho N' f d\Omega \quad (1.17)$$

y por definición:

$$\underline{K} = \int_{\Omega} B' DB d\Omega \quad (1.18)$$

$$\underline{M} = \rho \int_{\Omega} N' N d\Omega \quad (1.19)$$

$$\underline{f}_c = \int_{\Omega} N' f' d\Omega \quad (1.20)$$

$$\underline{f}_s = \int_{\Gamma} N' \sigma_{(n)} d\Gamma \quad (1.21)$$

donde:

(1.21)

\underline{K} = La matriz de rigideces de la estructura.

\underline{M} = Es la matriz de masas de la estructura.

\underline{f}_c = El vector de fuerzas de cuerpo de la estructura.

\underline{f}_s = El vector de fuerzas de superficie de la estructura.

Al sustituir las ecuaciones (1.18) a (1.21) en la ecuación (1.17) se tiene:

$$\underline{M} \bar{U} + \underline{K} \bar{U} = \underline{f}_s + \underline{f}_c \quad (1.22)$$

La ecuación (1.22) es la ecuación de equilibrio si considera que las fuerzas externas varían con el tiempo y además a esta ecuación se le puede agregar el término que relaciona el amortiguamiento por lo que se puede escribir.

$$\underline{M}\ddot{\underline{U}} + \underline{C}\dot{\underline{U}} + \underline{K}\underline{U} = \underline{p}(t) \quad (1.23)$$

donde:

\underline{C} = Es la matriz de amortiguamiento.

$\ddot{\underline{U}}$ = Es el vector de aceleración.

$\underline{p}(t)$ = Son las fuerzas de superficie y de cuerpo que varían con el tiempo.

$\dot{\underline{U}}$ = Es el vector de velocidad.

La solución de la ecuación (1.23) se va a efectuar mediante el método β de Newmark que se presenta en el capítulo 3, pero antes se discute el tipo de elemento a considerar así como se efectúa la formación de las matrices de rigideces y masas de la estructura en el capítulo 2.

1.2 DISCRETIZACION POR ELEMENTO FINITO

Al aplicar el método del elemento finito en la solución de la ecuación (1.23) se tienen las siguientes hipótesis:

a) El medio continuo se divide en un número finito de regiones y a cada una de estas regiones se le llama elemento finito.

b) Los elementos finitos están intercomunicados en un número finito de puntos nodales situados en las fronteras de los mismos y los desplazamientos de los puntos nodales son las incógnitas básicas del problema.

c) Se define en forma única el campo de desplazamientos en cualquier punto del elemento finito en función de los desplazamientos de los puntos nodales.

d) Conocidos los desplazamientos se pueden definir en forma única las deformaciones y los esfuerzos.

Según las hipótesis anteriores la región $\Omega + \Gamma$ se divide en elementos finitos, esto es:

$$\Omega = \sum_{i=1}^{nelem} \Omega_i^e \quad \Gamma = \sum_{i=1}^{nelem} \Gamma_i^e \quad (1.24)$$

con : $nelem = N^\circ$ de elementos

y la ecuación 1.16 se aplica en forma aislada a todos los elementos finitos por lo que se tiene:

$$\int_{\Omega^e} \underline{B}^t \underline{B} \underline{D} \underline{U}^e \, d\Omega^e + \int_{\Omega^e} \rho \underline{N}^{e^t} \underline{N}^e \underline{U} \, d\Omega^e = \int_{\Gamma^e} \underline{N}^{e^t} \underline{\sigma}_{(n)} \, d\Gamma^e + \int_{\Omega^e} \underline{N}^{e^t} \underline{f} \, d\Omega^e \quad (1.25)$$

Ω^e = región de cada elemento.

Γ^e = frontera de cada elemento.

en donde:

$$\underline{K}^e = \int_{\Omega^e} \underline{B}^t \underline{D} \underline{B} \, d\Omega^e \quad (1.26)$$

\underline{K}^e = matriz de rigidez del elemento

$$\underline{M}^e = \int_{\Omega^e} \rho \underline{N}^{e^t} \underline{N}^e \, d\Omega^e \quad (1.27)$$

\underline{M}^e = matriz de masas del elemento.

$$\underline{f}_s^e = \int_{\Gamma^e} \underline{N}^{e^t} \underline{\sigma}_{(n)} \, d\Gamma^e$$

\underline{f}_s^e = vectores de fuerzas de superficie del elemento

$$\underline{f}_c^e = \int_{\Omega^e} \underline{N}^{e^t} \underline{f} \, d\Omega^e \quad (1.29)$$

\underline{f}_c^e = vector de fuerzas de cuerpo del elemento.

al sustituir las ecuaciones de la (1.26) a la (1.29) en la (1.25) se obtiene el modelo discreto de cada elemento, esto es:

$$\underline{K}^e \underline{U}^e + \underline{M}^e \underline{\dot{U}} = \underline{f}_s^e + \underline{f}_c^e \quad (1.30)$$

La ecuación es válida únicamente en la región del elemento y a partir de calcular las matrices de rigideces y de masas de cada elemento es posible obtener las matrices de masas y rigideces del continuo discretizado en elementos.

1.3 ECUACIONES DE FORMA

En la solución aproximada considerada se deben definir las funciones de forma, las cuales dependen de los siguientes factores:

a) Geometría del dominio global.- Esto se refiere al tipo de espacio donde se van a integrar las ecuaciones, puede ser unidimensional, bidimensional o tridimensional.

b) El grado de aproximación deseado en la solución. Generalmente las funciones de forma son de interpolación, las cuales son polinomios con funciones exponenciales o trigonométricas. Si son polinomios lineales, únicamente se requieren los puntos nodales en las esquinas del elemento, si se utilizan polinomios cuadráticos se deben adicionar puntos nodales en la frontera del elemento.

c) Facilidad de integración sobre el dominio del elemento. La función de forma debe permitir la representación de cualquier forma lineal, de manera que se satisfaga el criterio de deformación constante.

d) Las incógnitas deben presentar continuidad entre elementos.

Existen varias familias de funciones de forma que cumplen con las características anteriores (polinomial, Lagrange, etc.) pero al considerar un grado de aproximación aceptable, se utiliza el elemento rectangular cuadrático de 8 puntos de la familia "serendipity". En la figura 1.2 se presenta este elemento con sus funciones de forma, la obtención de estas funciones de forma fue deducida por mera observación de tal manera que al sustituir en la función asociada al nodo su correspondiente coordenada tenga valor 1 y si en esa misma función se sustituye la coordenada de los otros puntos del elemento, su valor es cero.

Una vez definidas las funciones de forma, existe la posibilidad de mapear en dos sistemas de referencia elementos distorsionados ver figura 1.3 y esto es posible cuando existe una relación biunívoca entre estas transformaciones mediante las funciones de forma. En este caso existen dos grupos de funciones de forma, unas para definir la geometría y para hacer el mapeo de un sistema de referencia a otro y otras para hacer la interpolación del elemento. Esto en forma esquemática se puede escribir así:

Aproximación	Mapeo
$U = N U^u$	$X = N' \zeta$ $Y = N' \zeta$
para puntos nodales	sistemas de referencia $x - y$ sistema de referencia $\zeta - \zeta$ para m puntos nodales.

Cuando se hacen coincidir los puntos nodales y las funciones de forma para la aproximación y el mapeo, esto significa que $N^e = N'$ y $n=m$, a estos elementos se les conoce con el nombre de elementos isoparamétricos.

Al utilizar elementos isoparamétricos cuadráticos de la familia "serendipity", se deben definir las ecuaciones que efectúan ese mapeo y su influencia en las ecuaciones de equilibrio.

Las ecuaciones (1.26) a la (1.29) están referenciadas con respecto a un sistema local ζ, η . Para efectuar la transformación de estas ecuaciones al sistema de referencia $x - y$ se considera la ecuación (1.31) para el cambio de coordenadas.

$$\begin{aligned}
 X_q &= N_1 x_1 + N_2 x_2 + N_3 x_3 + N_4 x_4 + N_m x_m + N_n x_n + N_o x_o + N_p x_p \\
 Y_q &= N_1 y_1 + N_2 y_2 + N_3 y_3 + N_4 y_4 + N_m y_m + N_n y_n + N_o y_o + N_p y_p \\
 \forall q &= i, j, k, l, m, n, o, p
 \end{aligned}
 \tag{1.31}$$

en donde las funciones de forma están referenciadas al sistema ζ, η . La ecuación (1.31) permite obtener los puntos en la referencia $x - y$ donde se aplican los esfuerzos al resolver las ecuaciones de equilibrio en el sistema de referencia ζ, η .

Existe otra transformación a considerar, las derivadas de las funciones de forma, las cuales se deben de expresar en función de coordenadas locales ζ, η y cambiar convenientemente los límites de la integración, esta relación es:

$$\begin{bmatrix} \frac{\partial N_q}{\partial x} \\ \frac{\partial N_q}{\partial y} \end{bmatrix} = \underline{J}^{-1} \begin{bmatrix} \frac{\partial N_q}{\partial \zeta} \\ \frac{\partial N_q}{\partial \eta} \end{bmatrix} \quad q = i, j, k, l, m, n, o, p \quad (1.32)$$

en donde:

$$\underline{J}^{-1} = \frac{1}{|J|} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \zeta} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \end{bmatrix} \quad (1.33)$$

$$|J| = \frac{\partial x}{\partial \zeta} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \zeta} \quad (1.34)$$

Definida la relación de transformación en las ecuaciones (1.32) a (1.33) se pueden expresar las ecuaciones (1.26) y (1.29) en función de coordenadas locales y se tiene:

$$\underline{K}^e = \int_{-1}^1 \int_{-1}^1 J(\zeta, \eta) \underline{B}'(\zeta, \eta) d\zeta d\eta \quad (1.34)$$

$$\underline{f}_e^o = \rho t \int_{-1}^1 \int_{-1}^1 J(\zeta, \eta) \underline{N}'(\zeta, \eta) \underline{f} d\zeta d\eta \quad (1.35)$$

puesto que \underline{f} es el vector de fuerzas por unidad de masa, $\rho \underline{f}$ es un vector de fuerzas por unidad de volumen que se puede escribir:

$$\rho \underline{f} = \rho v \begin{bmatrix} \cos \alpha \\ \text{sen} \alpha \end{bmatrix} \quad (1.36)$$

ρv = peso volumétrico del material.

α = dirección de la gravedad en el sistema de referencia global.

Donde debido a que el vector $\rho \underline{f}$ no es función de las coordenadas locales puede salir fuera del operador integral por lo que la ecuación (1.35) queda:

$$\underline{f}_e^o = t \left[\int_{-1}^1 \int_{-1}^1 J(\zeta, \eta) \underline{N}'(\zeta, \eta) d\zeta d\eta \right] \rho \underline{f} \quad (1.37)$$

Para efectuar la integración de las ecuaciones (1.34), (1.35) se utiliza la cuadratura gaussiana por lo que las ecuaciones (1.34) y (1.35) quedan:

$$\underline{K}^e = t \sum_{m=1}^2 \sum_{n=1}^2 H_m H_n J(\zeta_m, \eta_n) \underline{B}^t(\zeta_m, \eta_n) \underline{D} \underline{B}(\zeta_m, \eta_n) \quad (1.38)$$

$$\underline{f}_c^e = t \sum_{m=1}^2 \sum_{n=1}^2 H_m H_n J(\zeta_m, \eta_n) \underline{N}^t(\zeta_m, \eta_n) \rho \underline{f} \quad (1.39)$$

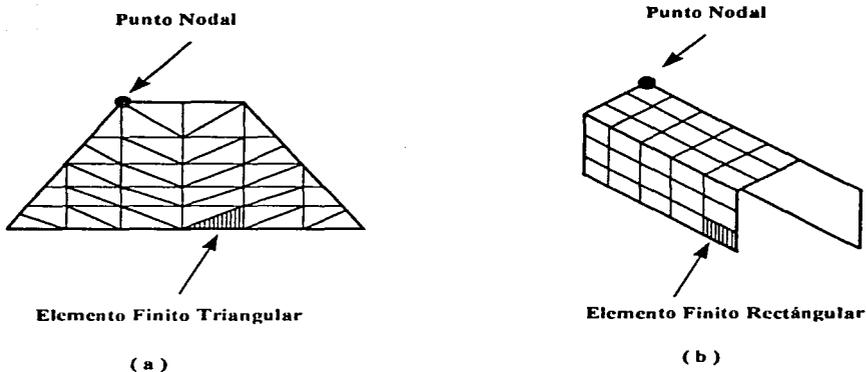
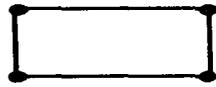


figura 1.1 Idealización de un continuo por elemento finito:
 a) Sección de la cortina de una presa.
 b) Cubierta.



(a) Lineal



(b) Cuadrático

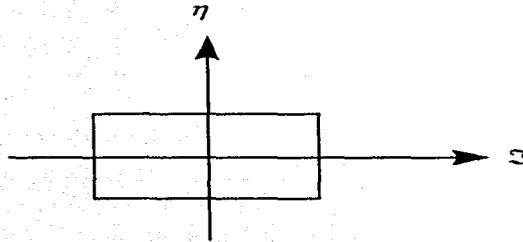


(c) Cúbico

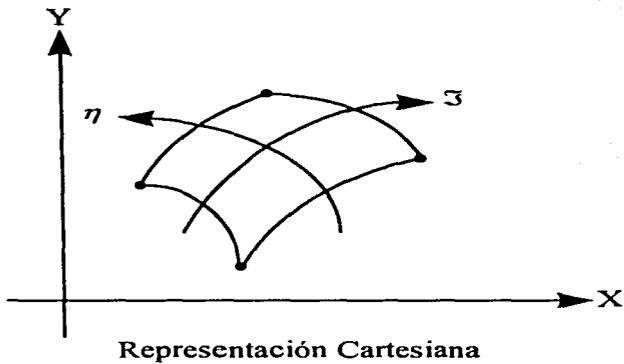


(d) Cuártico

(fig. 1.2) Funciones de forma para un elemento cuadrático de la familia "Serendipity".



Representación Local



(fig. 1.3) Transformación en dos dimensiones.

2. OBTENCION DE LAS MATRICES DE MASAS Y RIGIDECES DEL CONTINUO

2.1 INTRODUCCION

Para obtener las matrices de rigideces y de masas de cada elemento, se utiliza el elemento cuadrilátero de 8 puntos de la familia "Serendipity", Así, las matrices se pueden calcular a partir de las ecuaciones (1.37) y (1.38) respectivamente.

Por otra parte, el equilibrio de la estructura se establece al plantear el equilibrio de cada uno de los elementos que la componen.

Una parte del proceso de solución de la ecuación de movimiento es obtener las matrices de rigideces y de masas de la estructura a partir de las matrices de rigideces y masas de cada elemento, a este proceso se le conoce como ensamble.

En forma simbólica, se puede escribir el ensamble de la matriz de rigideces de la estructura como:

$$\underline{K} = \sum_{i=1}^{nbe} \underline{k}_{(i)} \quad (2.1)$$

donde:

nbe = número de elementos de la estructura.

$\underline{k}_{(i)}$ = es la matriz de rigideces del elemento i -ésimo.

\underline{K} = es la matriz de rigideces de la estructura.

En forma análoga, el ensamble de la matriz de masas de la estructura se puede escribir:

$$\underline{M} = \sum_{i=1}^{nbe} \underline{m}_{(i)} \quad (2.2)$$

donde:

nbe = número de elementos de la estructura.

$\underline{m}_{(i)}$ = es la matriz de masas del elemento i -ésimo.

\underline{M} = es la matriz de masas de la estructura.

Es importante destacar que las ecuaciones (2.1) y (2.2) únicamente representan el concepto de ensamble, esto es, que

la matriz de rigideces de la estructura se obtiene a partir de las matrices de rigideces de cada elemento, pero no es una sumatoria consecutiva, sino la colocación conforme al criterio del indicador de ecuación.

2.2 ENSAMBLE DE LA MATRIZ DE RIGIDECES DE LA ESTRUCTURA

Para explicar el procedimiento de ensamble en forma adecuada, se presenta un ejemplo a partir de una estructura formada por los elementos.

Definida la matriz de rigideces de cada elemento (matriz de 8×8), sus componentes ocupan una localidad en la matriz de rigideces global según el desplazamiento al que están asociados. En nodos comunes a dos elementos se sumarán en la misma localidad de la matriz de rigideces de la estructura, las contribuciones de cada elemento asociado a ese desplazamiento. En forma esquemática se presenta un ejemplo desde la figura 2.3 a la 2.6.

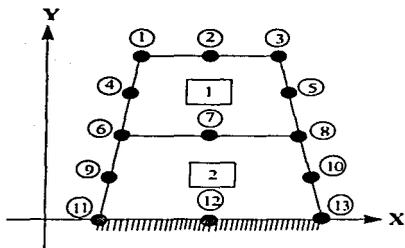


Fig. 2.3

División en elementos cuadráticos del continuo formando una malla estructurada.

Nodos	1	2	3	4	5	6	7	8	9	10	11	12	13
Desplazamientos x	1	3	5	7	9	11	13	15	17	19	0	0	0
Desplazamientos y	2	4	6	8	10	12	14	16	18	20	0	0	0

Fig. 2.4.

Indicadores de ecuación de la estructura de la figura 2.3

ELEMENTO 1

	Indicador de ecuación															
	5	6	3	4	1	2	7	8	11	12	13	14	15	16	9	10
5	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
6	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
3	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
4	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
2	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
7	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
8	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
11	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
12	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
13	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
14	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
15	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
16	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
9	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
10	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

↑ Indicador de ecuación

ELEMENTO 2

	Indicador de ecuación															
	15	16	13	14	11	12	17	18	0	0	0	0	0	0	19	20
15	y	y	y	y	y	y	y	y							y	y
16	y	y	y	y	y	y	y	y							y	y
13	y	y	y	y	y	y	y	y							y	y
14	y	y	y	y	y	y	y	y							y	y
11	y	y	y	y	y	y	y	y							y	y
12	y	y	y	y	y	y	y	y							y	y
17	y	y	y	y	y	y	y	y							y	y
18	y	y	y	y	y	y	y	y							y	y
0																
0																
0																
0																
0																
0																
19	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
20	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y

↑ Indicador de ecuación

Figura 2.6.3 Matriz de rigideces de los elementos 1 y 2.

└─ Indicador de ecuación

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0
2	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0
3	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0
4	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0
5	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0
6	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0
7	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0
8	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0
9	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0
10	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0
11	x	x	x	x	x	x	x	x	x	x	x+y	x+y	x+y	x+y	x+y	x+y	y	y	y	y
12	x	x	x	x	x	x	x	x	x	x	x+y	x+y	x+y	x+y	x+y	x+y	y	y	y	y
13	x	x	x	x	x	x	x	x	x	x	x+y	x+y	x+y	x+y	x+y	x+y	y	y	y	y
14	x	x	x	x	x	x	x	x	x	x	x+y	x+y	x+y	x+y	x+y	x+y	y	y	y	y
15	x	x	x	x	x	x	x	x	x	x	x+y	x+y	x+y	x+y	x+y	x+y	y	y	y	y
16	x	x	x	x	x	x	x	x	x	x	x+y	x+y	x+y	x+y	x+y	x+y	y	y	y	y
17	0	0	0	0	0	0	0	0	0	0	y	y	y	y	y	y	y	y	y	y
18	0	0	0	0	0	0	0	0	0	0	y	y	y	y	y	y	y	y	y	y
19	0	0	0	0	0	0	0	0	0	0	y	y	y	y	y	y	y	y	y	y
20	0	0	0	0	0	0	0	0	0	0	y	y	y	y	y	y	y	y	y	y

└─ Indicador de ecuación

Figura 1.6.4 Matriz de rigideces de la estructura

Para explicar el procedimiento de ensamble en forma adecuada, se presenta un ejemplo a partir de una estructura formada por sus elementos.

2.3 ENSAMBLE DE LA MATRIZ DE MASAS DE LA ESTRUCTURA

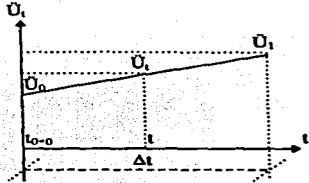
En forma análoga al ensamble de la matriz de rigideces de la estructura se obtiene la matriz de masas de la misma.

Una vez que se calcula la matriz de masas de cada elemento ecuación (1.38), con el indicador de ecuación que se obtuvo en el inciso anterior, se efectúa el ensamble de la matriz de cada elemento en la matriz de la estructura.

Al utilizar el mismo indicador de ecuación, se deduce que el tamaño de las matrices de rigideces y masas de la estructura es el mismo y depende del número de puntos nodales con desplazamiento diferente de cero, así como también del tamaño de las matrices de rigideces y de masas de cada elemento es el mismo.

3. METODO β DE NEWMARK

Con base en la ecuación de la recta y tomando en consideración la variación del desplazamiento representada por \ddot{U} , se observa la figura 3.1 y se obtiene la ecuación (3.1).



(figura 3.1)

$$\ddot{U}_i = \ddot{U}_0 + \frac{1}{\Delta t} (\ddot{U}_1 - \ddot{U}_0) t \quad (3.1)$$

Integrando por primera vez:

$$\dot{U}_i = \dot{U}_0 t + \frac{t^2}{2\Delta t} (\ddot{U}_1 - \ddot{U}_0) + c \quad (3.2)$$

Aplicando las condiciones frontera ($t=0, \dot{U}_i = \dot{U}_0$) se obtiene el valor de c el cual es sustituido en la ecuación (3.2) y obtenemos:

$$\dot{U}_i = \dot{U}_0 + \ddot{U}_0 t + \frac{t^2}{2\Delta t} (\ddot{U}_1 - \ddot{U}_0) \quad (3.3)$$

Integrando por segunda vez:

$$U_i = \dot{U}_0 t + \ddot{U}_0 \frac{t^2}{2} + \frac{t^3}{6\Delta t} (\ddot{U}_1 - \ddot{U}_0) + c \quad (3.4)$$

Aplicando las condiciones frontera ($t=0, U_i = U_0$) se obtiene el valor de c el cual es sustituido en la ecuación (3.4) y obtenemos:

$$\underline{U}_1 = \underline{U}_0 + \dot{\underline{U}}_0 t + \ddot{\underline{U}}_0 \frac{t^2}{2} + \frac{t^3}{6} \frac{1}{\Delta t} (\ddot{\underline{U}}_1 - \ddot{\underline{U}}_0) \quad (3.5)$$

Las ecuaciones (3.1), (3.3) y (3.5) se transforman al evaluarse para $(t = \Delta t); (\ddot{\underline{U}}_1 = \ddot{\underline{U}}_1); (\dot{\underline{U}}_1 = \dot{\underline{U}}_1); (\underline{U}_1 = \underline{U}_1)$ de la siguiente manera:

$$\dot{\underline{U}}_1 = \dot{\underline{U}}_1 \quad (3.6)$$

$$\dot{\underline{U}}_1 = \dot{\underline{U}}_0 + \frac{\Delta t}{2} (\ddot{\underline{U}}_1 + \ddot{\underline{U}}_0) \quad (3.7)$$

$$\underline{U}_1 = \underline{U}_0 + \Delta t \dot{\underline{U}}_0 + \frac{\Delta t^2}{6} (\ddot{\underline{U}}_1 + 2\ddot{\underline{U}}_0) \quad (3.8)$$

Ahora considerando que la ecuación de movimiento al establecer el equilibrio dinámico en un sistema discreto es la siguiente:

$$\underline{M}\ddot{\underline{U}}_1 + \underline{C}\dot{\underline{U}}_1 + \underline{K}\underline{U}_1 = \underline{P}_1 \quad (3.9)$$

Para resolver el modelo matemático de la ecuación (3.9) se debe entender que \underline{P}_1 resulta tener una variación similar a los acelerogramas registrados en la base, por lo que los métodos más recomendables para su integración son los numéricos paso a paso. Los métodos numéricos más utilizados son los de Superposición modal y el grupo de los directos que son el β de Newmark y θ de Wilson entre otros.

Los métodos paso a paso pueden ser utilizados cuando las ecuaciones de equilibrio del medio continuo proporcionen directamente la información necesaria para cuantificar los elementos del diseño del medio continuo (Elementos mecánicos y Cinemáticos).

Si se emplea el método de cálculo paso a paso de respuestas a temblores específicos, podrá acudirse a acelerogramas de temblores reales o de movimientos simulados, o a la combinación de estos siempre que se usen no menos de 4 movimientos representativos independientes entre sí.

Sustituyendo las ecuaciones (3.6), (3.7), y (3.8) y despejando $\ddot{\underline{U}}_1$, obtenemos:

(3.10)

$$\dot{U}_1 = \left[P_1 - C(\dot{U}_0 + \frac{\Delta t}{2} \ddot{U}_0) - K(U_0 + \Delta t \dot{U}_0 + \frac{\Delta t^2}{3} \ddot{U}_0) \right] \left[M + C \frac{\Delta t}{2} + K \frac{\Delta t^2}{3} \right]^{-1}$$

donde:

$$a = \dot{U}_0 + \frac{\Delta t}{2} \ddot{U}_0$$

$$b = U_0 + \Delta t \dot{U}_0 + \frac{\Delta t^2}{3} \ddot{U}_0$$

$$F = \left[M + C \frac{\Delta t}{2} + K \frac{\Delta t^2}{6} \right]^{-1}$$

En forma explícita tenemos:

$$\dot{U}_1 = F \left[P_1 - C(a) - K(b) \right] \quad (3.11)$$

Conociendo el valor de \dot{U}_1 , se puede conocer los valores de \ddot{U}_1 & U_1 , basándose en las ecuaciones (3.6), (3.7) y (3.8) como se puede observar.

$$\dot{U}_1 = a + \frac{\Delta t}{2} \ddot{U}_1 \quad (3.12)$$

$$U_1 = b + \frac{\Delta t^2}{6} \ddot{U}_1 \quad (3.13)$$

Las ecuaciones (3.11), (3.12) y (3.13) son las ecuaciones Predictoras ó Generalización Lineal.

Ahora considerando el Amortiguamiento de Rayleigh $C = \alpha M + \mu K$ el cual se sustituye en F se obtiene:

$$F = \left[\left(1 + \alpha \frac{\Delta t}{2}\right) M + \left(\mu \frac{\Delta t}{2} + \frac{\Delta t^2}{6}\right) K \right]^{-1} \quad (3.14)$$

Sustituyendo el Amortiguamiento de Rayleigh se obtiene el valor de F en las ecuaciones (3.11), (3.12) y (3.13), se

obtiene las Ecuaciones Predictoras con el criterio de Rayleigh las cuales se presentan a continuación:

$$\ddot{\mathbf{U}}_1 = [\mathbf{P}_1 - \alpha \mathbf{a} \mathbf{M} + (-\mu \mathbf{a} - \mathbf{b}) \mathbf{K}] \mathbf{F} \quad (3.15)$$

$$\dot{\mathbf{U}}_1 = \mathbf{a} + \frac{\Delta t}{2} \ddot{\mathbf{U}}_1 \quad (3.16)$$

$$\mathbf{U}_1 = \mathbf{b} + \frac{\Delta t^2}{6} \ddot{\mathbf{U}}_1 \quad (3.17)$$

Donde:

$$\mathbf{a} = \dot{\mathbf{U}}_0 + \frac{\Delta t}{2} \ddot{\mathbf{U}}_0$$

$$\mathbf{b} = \mathbf{U}_0 + \Delta t \dot{\mathbf{U}}_0 + \frac{\Delta t^2}{3} \ddot{\mathbf{U}}_0$$

$$\mathbf{F} = \left[\left(1 + \alpha \frac{\Delta t}{2} \right) \mathbf{M} + \left(\mu \frac{\Delta t}{2} + \frac{\Delta t^2}{6} \right) \mathbf{K} \right]^{-1}$$

3.1 METODO GENERALIZADO

El método de Newmark, con base en el Método de la Aceleración Lineal, generaliza la variación de la aceleración relativa, al introducir dos parámetros adimensionales cuyos valores se definen de tal forma que el método resulta eficiente.

Newmark generaliza las ecuaciones predictoras (3.6), (3.7) y (3.8) al introducir dos parámetros adimensionales β y γ como sigue:

$$\ddot{\mathbf{U}}_1 = \ddot{\mathbf{U}}_0 \quad (3.6)$$

$$\dot{\mathbf{U}}_1 = \dot{\mathbf{U}}_0 + \frac{\Delta t}{2} (\ddot{\mathbf{U}}_1 + \ddot{\mathbf{U}}_0) \quad (3.7)$$

$$\underline{U}_1 = \underline{U}_0 + \Delta t \dot{\underline{U}}_0 + \frac{\Delta t^2}{6} (\ddot{\underline{U}}_1 + 2\ddot{\underline{U}}_0) \quad (3.8)$$

que conducen a:

$$\ddot{\underline{U}}_1 = \ddot{\underline{U}}_1 \quad (3.18)$$

$$\dot{\underline{U}}_1 = \dot{\underline{U}}_0 + (1-\gamma)\Delta t \ddot{\underline{U}}_0 + \gamma \Delta t \ddot{\underline{U}}_1 \quad (3.19)$$

$$\underline{U}_1 = \underline{U}_0 + \Delta t \dot{\underline{U}}_0 + \left(\frac{1}{2} - \beta\right) \Delta t^2 \ddot{\underline{U}}_0 + \beta \Delta t^2 \ddot{\underline{U}}_1 \quad (3.20)$$

El parámetro β está relacionado con la estabilidad del método y el parámetro γ con el amortiguamiento matemático.

Para el caso $\beta = 1/6$ y $\gamma = 1/2$ se tiene las ecuaciones (3.6), (3.7) y (3.8) del Método de Aceleración Lineal.

Al sustituir las ecuaciones (3.18), (3.19) & (3.20) en la Ecuación de Movimiento (3.1) al final del paso y al despejar el valor de \underline{U}_1 , se obtiene:

$$(3.21)$$

$$\underline{U}_1 = \left[\underline{P}_1 - \underline{C}[\dot{\underline{U}}_0 + (1-\gamma)\Delta t \ddot{\underline{U}}_0] - \underline{K} \left[\underline{U}_0 + \Delta t \dot{\underline{U}}_0 + \left(\frac{1}{2} - \beta\right) \Delta t^2 \ddot{\underline{U}}_0 \right] \right] * [\underline{M} + \underline{c}\gamma \Delta t + \underline{k}\beta \Delta t^2]$$

Donde:

$$\underline{a} = \dot{\underline{U}}_0 + (1-\gamma)\Delta t \ddot{\underline{U}}_0$$

$$\underline{b} = \underline{U}_0 + \Delta t \dot{\underline{U}}_0 + \left(\frac{1}{2} - \beta\right) \Delta t^2 \ddot{\underline{U}}_0$$

$$\underline{F} = [\underline{M} + \underline{C}\gamma \Delta t + \underline{K}\beta \Delta t^2]^{-1}$$

de forma explícita se obtiene lo siguiente:

$$\underline{U}_1 = \underline{F}[\underline{P}_1 - \underline{C}(\underline{a}) - \underline{K}(\underline{b})] \quad (3.22)$$

Conociendo el valor de $\underline{\dot{U}}_1$, se puede conocer los valores de $\underline{\dot{U}}_1$ & \underline{U}_1 , basándose en las ecuaciones (3.18), (3.19) y (3.20) como se puede observar.

$$\underline{\dot{U}}_1 = \underline{a} + \gamma \Delta t \underline{\dot{U}}_1 \quad (3.23)$$

$$\underline{U}_1 = \underline{b} + \beta \Delta t^2 \underline{\dot{U}}_1 \quad (3.24)$$

donde:

$$\underline{a} = \underline{\dot{U}}_0 + (1 - \gamma) \Delta t \underline{\ddot{U}}_0$$

$$\underline{b} = \underline{U}_0 + \Delta t \underline{\dot{U}}_0 + \left(\frac{1}{2} - \beta\right) \Delta t^2 \underline{\ddot{U}}_0$$

$$\underline{F} = [\underline{M} + \underline{C} \gamma \Delta t + \underline{K} \beta \Delta t^2]^{-1}$$

Las ecuaciones (3.22), (3.23) y (3.24) se conocen como las Ecuaciones Predictoras Generalizadas con el Criterio de Newmark

Ahora al considerar el Amortiguamiento de Rayleigh $\underline{C} = \alpha \underline{M} + \mu \underline{K}$ el cual se sustituye en \underline{F} se obtiene:

$$\underline{F} = [(1 + \alpha \gamma \Delta t) \underline{M} + (\mu \gamma \Delta t + \beta \Delta t^2) \underline{K}]^{-1} \quad (3.25)$$

Sustituyendo el Amortiguamiento de Rayleigh se obyiene el valor de \underline{F} en las ecuaciones (3.22), (3.23) y (3.24) se obtiene las Ecuaciones Predictoras de Newmark con Amortiguamiento de Rayleigh las cuales se presentan a continuación:

$$\underline{\dot{U}}_1 = [\underline{P}_1 - \alpha \underline{a} \underline{M} + (-\mu \underline{a} - \beta) \underline{K}] \underline{F} \quad (3.26)$$

$$\underline{\dot{U}}_1 = \underline{a} + \gamma \Delta t \underline{\dot{U}}_1 \quad (3.27)$$

$$\underline{U}_1 = \underline{b} + \beta \Delta t^2 \underline{\dot{U}}_1 \quad (3.28)$$

Donde:

$$\underline{a} = \underline{\dot{U}}_0 + (1 - \gamma)\Delta t \underline{\ddot{U}}_0$$

$$\underline{b} = \underline{U}_0 + \Delta t \underline{\dot{U}}_0 + \left(\frac{1}{2} - \beta\right)\Delta t^2 \underline{\ddot{U}}_0$$

$$\underline{F} = \left[(1 + \alpha\gamma\Delta t)\underline{M} + (\mu\gamma\Delta t + \beta\Delta t^2)\underline{K} \right]^{-1}$$

Para efecto de facilitar el manejo de las ecuaciones se plantea la solución como un sistema de ecuaciones del tipo $\underline{Ax} = \underline{b}$ y se sustituye $\underline{F} = \underline{\bar{K}}$.

$$\underline{\bar{K}}\underline{\dot{U}}_1 = \underline{r} \quad (3.29)$$

donde:

$$\underline{\bar{K}} = (1 + \alpha\gamma\Delta t)\underline{M} + (\mu\gamma\Delta t + \beta\Delta t^2)\underline{K} \quad (3.30)$$

$$\underline{r} = -\alpha\underline{M}\underline{a} - (\mu\alpha + \beta)\underline{K}\underline{b} + \underline{p}_1 \quad (3.31)$$

Al resolver la ecuación (3.29), se puede conocer el valor de $\underline{\dot{U}}_1$ y \underline{U}_1 de las ecuaciones (3.27) y (3.28) respectivamente

3.2 ALGORITMO NUMERICO

Con base en las ecuaciones del inciso anterior es posible escribir el algoritmo numérico para el método de la Aceleración Generalizada de Newmark de la siguiente manera:

1.- Se determinan las constantes:

$$E0 = \gamma\Delta t$$

$$E1 = \beta\Delta t\Delta t$$

$$EK1 = 1 - E0\alpha$$

$$EK2 = E0\mu + E1$$

$$EA1 = \Delta t - E0$$

$$EB1 = (1/2 - \beta)\Delta t\Delta t$$

2.- Se calcula la matriz $\underline{\bar{K}}$

$$\underline{\bar{K}} = EK1 \underline{M} + EK2 \underline{K}$$

3.- Se realiza la triangulación de \bar{K}

4.- Se determinan los vectores \underline{a} , \underline{b} y \underline{r} para cada punto del acelerograma:

$$\underline{a} = \underline{\dot{U}}_0 + EAI\underline{\dot{U}}_1$$

$$\underline{b} = \underline{U}_0 + \Delta t \underline{\dot{U}}_0 + EBI\underline{\dot{U}}_0$$

$$\underline{r} = -\underline{\dot{U}}_0 M - \alpha M \underline{a} - K(\mu \underline{a} + \underline{b})$$

donde:

$-\underline{\dot{U}}_0 M = \underline{P}$, que respresenta cada punto del acelerograma multiplicado por la masa.

5.- Se resuelve el sistema de ecuaciones:

$$\bar{K}\underline{U}_1 = \underline{r}$$

6.- Se Obtienen los vectores:

$$\underline{\dot{U}}_1 = \underline{a} + EO\underline{\dot{U}}_1$$

$$\underline{U}_1 = \underline{b} + EI\underline{U}_1$$

4. CRITERIOS DE PROGRAMACIÓN Y SISTEMA PARA EL ANÁLISIS

4.1 ASPECTOS GENERALES

4.1.1 SEMBLANZA HISTÓRICA

La historia de las computadoras comienza en 1944 a raíz de la segunda guerra mundial, con el objetivo de calcular con mayor precisión las trayectorias balísticas; el resultado del citado proyecto militar fue una "máquina de calcular" llamada ENIAC (Calculadora e Integradora Numérica Electrónica) que contenía unos 18,000 tubos electrónicos de vacío (bulbos), a esta se le considera parte de la primera generación de computadoras. Los bulbos generaban tanto calor que era necesario un aire acondicionado casi del mismo tamaño que la computadora, por lo que se desarrolló el transistor el cual reemplazó a los bulbos. De aquí parte la segunda generación de computadoras, aunque disminuyó mucho el tamaño de las máquinas aún continuaban siendo grandes y aún era necesario un gran sistema de enfriamiento.

Los sistemas de transistores comenzaron a mostrar una disminución en precio y tamaño por lo que aparecieron los primeros Circuitos Integrados marcando la tercera generación de computadoras; los circuitos integrados también pasaron a formar parte de los sistemas de integración a gran escala, es decir, agrupar a cientos de miles de transistores en una pequeña pastilla conocida como CHIP, la cual es conocida hasta nuestros días como la cuarta generación de computadoras.

Basados en la tecnología del CHIP o Microchip se han desarrollado varios tipos de computadoras según su tamaño y aplicación que van desde la Microcomputadora o Computadora Personal, la Minicomputadora, la Macrocomputadora o Mainframe y la Supercomputadora.

4.1.2 JERARQUÍA DE LOS LENGUAJES

Von Newman (1903-1957) propuso el concepto de programa almacenado de acuerdo a las siguientes características:

- 1.- Disponer de un medio para codificar instrucciones a fin de poder almacenarlas en la memoria de la máquina.
- 2.- Almacenar las instrucciones en la memoria junto con cualquier otra instrucción necesaria para el trabajo específico de que se trate.

3.- Al correr el programa, tomar las instrucciones directamente de la memoria.

Al cubrir estos tres puntos se tienen las siguientes ventajas:

1.- Velocidad: el acceso a la información es en forma directa.

2.- Flexibilidad: Al tener varios programas almacenados se pueden realizar combinaciones entre éstos.

3.- Automodificación: Se pueden escribir programas que se puedan modificar o ajustar por sí mismos.

La jerarquía de los lenguajes de programación establece tres diferentes clasificaciones: el nivel bajo lo tienen los lenguajes de ensamble en los cuales se reemplazan los 1's y 0's por códigos simbólicos llamados *mnemotécnicos*; éstos por su complejidad, son muy similares al lenguaje máquina por lo que se deben tener en cuenta varios detalles, tales como el indexado y las localizaciones de almacenamiento; el nivel medio es una combinación de elementos del lenguaje de alto nivel con la funcionalidad del ensamblador o lenguaje de bajo nivel; los lenguajes de alto nivel llamados lenguajes *compiladores* tales como el FORTRAN, COBOL, Pascal, ALGOL, BASIC, etc., son casi independientes de la máquina ya que los programas escritos en este tipo de lenguajes pueden correrse, con modificaciones muy pequeñas, en máquinas diferentes lo cual se conoce como portabilidad. Un compilador es un programa especial que traduce los programas escritos en este tipo de lenguajes de alto nivel, a lenguaje de máquina. El programa original se denomina el programa fuente y su traducción recibe el nombre de programa objeto véase figura 4.1.

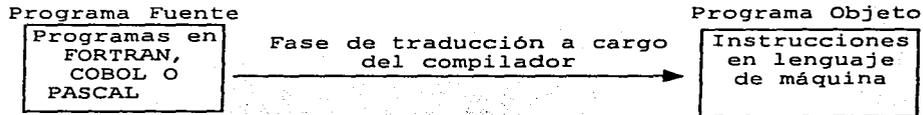


Fig. 4.1

El Visual BASIC está considerado dentro de los lenguajes de alto nivel pero no es un compilador ya que no genera código objeto, más bien es un lenguaje intérprete ya que para correr un programa es necesario estar dentro del editor de Visual

BASIC y la ejecución del programa es en la memoria de la computadora, aunque tiene la opción de generar un archivo ejecutable.

4.2 CRITERIOS DE PROGRAMACIÓN

4.2.1 FORMAS DE PROGRAMAR

Un criterio para analizar los diferentes compiladores o intérpretes es a partir de considerar su estilo de programación; existe la programación secuencial, cuando todo el código del programa está dentro del programa principal, por ejemplo el intérprete de BASIC, la programación modular o estructurada, cuando el código del programa está segmentado en varios subprogramas que se pueden ejecutar en cualquier punto del programa principal, por ejemplo FORTRAN, Pascal, C, COBOL, etc. En los últimos años, se han desarrollado otros estilos de programación como son la Programación Orientada a Objetos la cual consiste en considerar a cada módulo del programa como un objeto independiente y con características exclusivas, y la Programación Orientada a Eventos la cual se aplica a técnicas de ambiente gráfico como Visual BASIC, esto es, poder determinar el comportamiento del programa a partir de los eventos que puedan ocurrir durante la ejecución del mismo.

4.2.2 PROGRAMAS DE APLICACIÓN

Los programas de aplicación se pueden clasificar de acuerdo al funcionamiento para que fueron creados:

Financieros: En los que se encuentran los que sirven para Manejo de Nómina, Cuentas por Cobrar, Cuentas por Pagar, Facturación, Bancos, IMSS o SAR.

Matemáticos: Calculadoras, Manejadores de Ecuaciones Diferenciales, Programación Lineal y no Lineal, etc.

Diseño Arquitectónico: Arquitectura, Ingeniería Mecánica, Ingeniería Civil, Ingeniería Eléctrica, etc.

Publicidad: Diseño Gráfico.

Administrativos: Procesadores de Texto, Hojas de Cálculo, Presentaciones, Manejadores de Bases de Datos, etc.

Entretenimiento: Juegos de Cartas, Ajedrez, Pistas de Carreras, Simuladores de Vuelo, etc.

Educativos: Niveles de Primaria, Secundaria y Preparatoria, Personas Discapacitadas, etc.

Médicos: Programas para Tomografía, Endoscopia, Optometría, etc.

4.3 VISUAL BASIC

4.3.1 HISTORIA

Con el advenimiento de Microsoft Windows los usuarios de PC trabajaron en un entorno intuitivo y gráficamente rico. Una interfase gráfica de usuario permite que las aplicaciones sean fáciles de aprender y de usar. En lugar de aprender a escribir largos y complicados comandos, el usuario simplemente selecciona una opción (o comando) de un menú con un "Clic" de un botón del ratón. Diversas ventanas en la pantalla permiten que el usuario pueda ejecutar más de un programa a la vez. Cuando un programa necesita información o que el usuario tome decisiones aparecen los cuadros de diálogo.

Muchos creyeron que Windows representaba el fin de la programación aficionada. En el mundo de MS-DOS, los profesionales de áreas no relacionadas con la informática podían (en líneas generales) aprender lo suficiente de programación como para escribir aplicaciones sencillas que les ayudaran en su trabajo, les ahorraran tediosos cálculos u organizaran los datos rápidamente. Pero, ¿podía alguien hacer eso mismo en Windows, donde los requerimientos de programación eran tan complicados, incluso para las aplicaciones más sencillas?

La respuesta surgió en 1991 cuando Microsoft presento Visual BASIC. El lenguaje de programación Visual BASIC maneja la complejidad de Windows combinando las probadas posibilidades del lenguaje BASIC, con herramientas de diseño visual proporciona simplicidad y facilidad de uso, sin sacrificar características gráficas que hacen de Windows un entorno tan amigable para trabajar. Los menús, fuentes, cuadros de diálogo, campos de texto con desplazamiento y todo lo demás se diseñan con facilidad, y esas características no requieren más que unas pocas líneas de programa para controlarlas.

4.3.2 ESTRUCTURA

Visual BASIC es uno de los primeros lenguajes de programación de los llamados "Orientado a Eventos", un estilo de programación especialmente adaptado a las interfases gráficas

de usuario. Tradicionalmente, la programación ha estado muy orientada al proceso, al paso a paso, de manera muy similar al de una receta. Uno de los inconvenientes de este estilo consiste en que la persona que escribió el programa, es la que determina qué hay que hacer en cada momento. Pero en los modernos programas de computadoras el objetivo es que el usuario determine en cada momento que es lo que quiere hacer y eso es exactamente lo que proporciona la programación orientada a eventos. En lugar de escribir un programa que determina cada uno de los pasos en un orden determinado, el programador escribe un programa que responde a las acciones del usuario: elegir un comando, hacer "clic" en una ventana, mover el ratón. En vez de escribir un gran programa, el programador crea una aplicación que es realmente una colección de microprogramas que cooperan entre ellos y que se ejecutan a raíz de eventos iniciados por el usuario. Con Visual BASIC, una aplicación así se puede escribir con una rapidéz y facilidad sin precedentes.

4.3.3 APLICACIONES

Debido al gran auge de los ambientes gráficos como Windows o Windows 95 se plantea como un estándar la utilización de éstos como la interfase con el usuario, por lo que el desarrollo de aplicaciones bajo esta plataforma es cada día más común. El lenguaje BASIC como tal surgió como una necesidad de crear aplicaciones sencillas sin utilizar instrucciones complicadas. Visual BASIC retoma este principio creando aplicaciones en entornos gráficos con pocas líneas de código, las aplicaciones pueden ser en programas tanto administrativos como científicos, ya que en un programa desarrollado en lenguaje BASIC pueden ser utilizadas las subrutinas e incluirlas en los formularios de Visual BASIC sin afectar el resultado de las mismas, además de que se mejora en mucho la presentación de los programas.

4.3.4 VENTAJAS Y DESVENTAJAS DE VISUAL BASIC

Como se ha mencionado, el mayor atractivo de Visual BASIC es el entorno gráfico en el que se desarrolla, además de que tiende a ser un estándar en los ambientes de programación.

La desventaja más común es la que presenta cualquier programa de nueva aparición en el mercado es la de los requerimientos de operación como son: velocidad del procesador, memoria extendida y contar con un sistema operativo de ambiente gráfico.

4.4 FORTRAN

4.4.1 HISTORIA

El lenguaje FORTRAN surge de la necesidad de realizar operaciones matemáticas y cálculos complicados en el menor tiempo posible, de aquí que su nombre se compone de las palabras traducción de fórmulas (FORMula TRANslation).

4.4.2 ESTRUCTURA

El lenguaje FORTRAN contempla los siguientes aspectos técnicos:

- 1.- Es preciso en los cálculos complicados.
- 2.- Es rápido ya que crea un código objeto libre de errores y ese código puede ser reutilizado sin tener que compilarlo.
- 3.- Se han establecido estándares mínimos y versiones extendidas por lo que existe una variación entre sus dialectos.
- 4.- Es relativamente poderoso en lo que respecta a las operaciones de entrada/salida.

4.4.3 APLICACIONES

Debido a que los cálculos del método β de Newmark requieren de operaciones simples de entrada y salida de datos, manejo de los mismos y una gran cantidad de cálculos, se tomo en cuenta el lenguaje FORTRAN. Una característica es que utiliza una notación para facilitar la escritura de fórmulas matemáticas en su empleo de la programación, por lo cual el FORTRAN es el lenguaje propio de la mayoría de los ingenieros y científicos. A partir de su utilidad en cálculos, se han desarrollado diversos compiladores de FORTRAN para diferentes tipos y tamaños de computadoras.

4.4.4 VENTAJAS Y DESVENTAJAS

Entre las ventajas del lenguaje de programación FORTRAN en el desarrollo de programas de ingeniería estructural esta el que el compilador FORTRAN posee proposiciones que lo hacen más potente en el caso de su rapidez, alta precisión en los cálculos matemáticos, etc.

Las desventajas son al trabajar en forma interactiva, ya que su configuración fue hecha para trabajar con lectura de tarjetas perforadas, además no incluye la creación de gráficos.

4.5 ORGANIZACION DEL SISTEMA

4.5.1 ORGANIZACION

Para poder explicar la organización del sistema se describirá el objetivo de cada una de las partes que la componen:

PREPROCESADOR

El diseño del programa Preprocesador cuenta con 5 Formas o Formularios que a su vez contienen código asociado a cada elemento que conforma el formulario; también contiene un módulo de declaraciones y procedimientos comunes en todo el programa.

El programa captura los datos iniciales del medio continuo a analizar, solicitando el número de Puntos Nodales y sus coordenadas en el plano XY, el número de Elementos en el que se dividirá el medio continuo y los Puntos Nodales que componen cada Elemento, las restricciones o Condiciones Frontera a que está sujeto el medio continuo y por último, el número de Materiales y sus datos técnicos, que se utilizarán en el análisis.

Una vez completados los datos es posible generar una impresión de los datos para su análisis en el escritorio, así como también la generación del archivo con extensión .dat que servirá para alimentar al Procesador 1.

PROCESADOR 1

Se encarga de realizar los cálculos de las matrices de Masas y Rigideces por el Método del Elemento Finito de 8 puntos. Los datos que solicita el Procesador 1 son: nombre del archivo con extensión .dat, la unidad de disco donde se encuentra y por último el nombre del archivo de resultados. El archivo de resultados servirá para alimentar al Procesador 3.

PROCESADOR 2

El Procesador 2 está elaborado en Turbo Pascal y fue desarrollado por el Lic. en Matemáticas Aplicadas y Computación Oscar Alejandro González Bustamante; de éste se

obtienen los datos de los acelerogramas que se utilizarán para el Procesador 3.

PROCESADOR 3

El Procesador 3, también desarrollado en FORTRAN, es el encargado de aplicar el método β de Newmark con los datos generados por el Procesador 1 (Matrices de masas y rigideces) además de las constantes α , β , γ , y μ la información del acelerograma del Procesador 2.

4.5.2 PROCESO DE SOLUCION

Se presenta la secuencia de cálculo a seguir:

- a) Se discretiza el medio continuo a analizar mediante una malla de elementos cuadriláteros de 8 puntos.
- b) Se numeran los puntos nodales y los elementos del medio continuo.
- c) Se calcula la matriz de rigideces de cada elemento con la ecuación 1.37
- d) Se calcula la matriz de masas de cada elemento con la ecuación 1.38
- e) Se ensambla la matriz de rigideces del medio continuo mediante la ecuación 2.1
- f) Se obtiene la matriz de masas a partir de la ecuación 1.38

Se aplica el método β de Newmark utilizando los valores del acelerograma para obtener los vectores de Aceleración, Velocidad y Desplazamiento.

5. PREPROCESADOR

El termino preprocesador se refiere al proceso mediante el cual se capturan y se preparan los datos requeridos de una estructura para su análisis matemático. Estos datos se almacenan en un archivo que alimenta al programa procesador para realizar los cálculos matemáticos necesarios para la toma de decisiones.

5.1 GENERALIDADES

La función principal del programa Preprocesador es la de crear un archivo en el que se contendrán los datos iniciales de un medio continuo para ser analizados por los procesadores, así como permitir la impresión de los datos en papel. Una de las facilidades que ofrece el Preprocesador es el manejo de menús de ventanas colgantes los cuales permiten acceder fácilmente a las opciones a través de las teclas de navegación o con algunas combinaciones de teclas; también presenta una barra de botones en los cuales se ilustra gráficamente la acción que realiza cada uno. Una ventaja que ofrece el programa es el manejo de archivos ya que permite abrir archivos existentes creados por el Preprocesador, guardar datos capturados del medio continuo, imprimir los datos en el papel para su análisis en el escritorio y borrar archivos innecesarios. Su principal atractivo es que opera bajo un ambiente completamente gráfico como lo es Microsoft Windows.

5.2 ORGANIZACIÓN DEL PROGRAMA

Como se mencionó anteriormente, el ambiente de programación de Visual BASIC es completamente diferente al modo convencional de programar, principalmente por que se programan los eventos que pueden ocurrir durante la ejecución del programa.

Al realizar un programa en Visual BASIC primeramente se diseña la interfase gráfica con el usuario, para lo cual Visual BASIC proporciona un formulario, este equivale a una ventana de aplicación en Windows. A cada formulario se le agregan objetos que pueden ser botones, barras de desplazamiento, cuadros de texto, cuadrículas, etc. A cada uno se le asocian ciertas propiedades las cuales dependen del objeto en sí; así mismo estos están relacionados a una serie de eventos que pueden suceder durante la ejecución del programa. Cada evento se considera una subrutina sólo si se agrega código y es de esta manera como se determina la acción

que realizará el programa al ocurrir dicho evento. También es posible crear subrutinas independientes que controlen otras acciones del programa o realicen el enlace entre los objetos y los formularios. Otra cualidad que tiene Visual BASIC es la facilidad de diseñar menús de ventanas colgantes los cuales pueden ser accedados mediante combinaciones de teclas, teclas de navegación o directamente con el puntero del mouse.

En este programa se utilizaron 6 formularios, el primero es la interfase con el usuario mediante la cual puede acceder a las diferentes opciones, 4 corresponden a las pantallas de captura de datos y el último contiene información acerca de los autores del programa.

El formulario principal llamado MENU.FRM contiene los objetos básicos para acceder a las opciones del programa como son barras de botones, menús de ventanas colgantes y cuadros de texto, dentro del código del formulario existen 17 subrutinas que realizan las funciones de manejo de archivos y acceso a las pantallas de captura de datos, así como la opción salir del programa. Las subrutinas mencionadas son las siguientes:

Subrutina	Función
Nuevo	Inicializa las pantallas para captura de datos.
Abrir	Abre un archivo de datos existente en disco.
Borrar	Borra un archivo de datos del disco.
Guardar	Guarda los datos capturados del medio continuo en un archivo.
Imprimir	Muestra la ventana para impresión de datos.
Imparchivo	Envía la información capturada a un archivo para su posterior impresión.
Impresora	Envía la información capturada a la impresora.
CargaPuntos	Carga en memoria el formulario para capturar los datos de Puntos Nodales.
CargaElementos	Carga en memoria el formulario para capturar los datos de los Elementos.
CargaCondiciones	Carga en memoria el formulario para capturar los datos de las Condiciones de Frontera.

CargaMateriales	Carga en memoria el formulario para capturar los datos de los Materiales.
LeerPuntos	Lee los datos de Puntos Nodales de un archivo existente y los carga al arreglo de Puntos nodales.
LeerElementos	Lee los datos de los Elementos de un archivo existente y los carga al arreglo de los Elementos.
LeerCondiciones	Lee los datos de las Condiciones de Frontera de un archivo existente y los carga al arreglo de las Condiciones de Frontera.
LeerMateriales	Lee los datos de los Materiales de un archivo existente y los carga al arreglo de los Materiales.
Salir	Sale del programa.
Acercade	Carga el formulario Acerca.Frm.

Los siguientes formularios son los que se utilizan como ventanas para la captura de los datos de los puntos nodales, los elementos, las condiciones de frontera y los materiales. Cada uno de estos formularios contiene un cuadro de texto, dos botones y una cuadrícula que se ajusta al número de datos deseados. Cada objeto contiene un código para realizar las acciones que se requieren para los posibles eventos y éste es el mismo para todos. El formulario que presenta la información de los autores contiene un botón de aceptar, etiquetas y un cuadro de dibujo que contiene el escudo de la universidad. Los nombres de los formularios y las acciones que cada uno realiza son los siguientes:

Formulario	Función
Acerca.Frm	Muestra la información acerca de los autores del programa y el escudo de la universidad.
Puntos.Frm	Contiene los objetos necesarios para crear la ventana donde se capturan los datos de los Puntos Nodales requeridos.
Elemento.Frm	Contiene los objetos necesarios para crear la ventana donde se capturan los datos de los elementos requeridos.

Frontera.Frm Contiene los objetos necesarios para crear la ventana donde se capturan los datos de las Condiciones de Frontera requeridos.

Material.Frm Contiene los objetos necesarios para crear la ventana donde se capturan los datos de los Materiales requeridos.

Visual BASIC utiliza módulos que contienen sólo código y que se utilizan para la compartición de subrutinas entre los diferentes formularios del programa así como el poder manejar variables que sean comunes en todos los formularios. En este proyecto se cuenta con un módulo llamado DISEÑO.BAS el cual contiene la declaración de las variables globales que se utilizan en el programa y también 4 subrutinas para diseñar las tablas donde se insertan los datos de acuerdo al tamaño de las matrices necesarias. Las subrutinas son las siguientes:

Subrutina	Función
DiseñoPuntos	Diseña la tabla para captura de datos de acuerdo al número de Puntos Nodales requeridos.
DiseñoElementos	Diseña la tabla para captura de datos de acuerdo al número de Elementos requeridos.
DiseñoCondiciones	Diseña la tabla para captura de datos de acuerdo al número de Condiciones de Frontera requeridos.
DiseñoMateriales	Diseña la tabla para captura de datos de acuerdo al número de materiales requeridas.

5.3 MANUAL DE USUARIO

Los requerimientos mínimos para la utilización del programa son:

- Microcomputadora con procesador 80386 o superior
- 4 Mbytes de memoria RAM
- Sistema Operativo 5.0 o superior
- Windows 3.1, Windows 3.11 o Windows 95

Dado que el programa Preprocesador trabaja completamente en ambiente Windows es recomendable la utilización de un Mouse aunque el programa puede ser utilizado también con el teclado.

Para ejecutar el programa Preprocesador es necesario primero realizar la instalación del programa ya que se requieren algunos archivos controladores que son grabados en el subdirectorio SYSTEM de WINDOWS.

Para instalar el programa se deben realizar los siguientes pasos

1. Insertar el diskette que contiene el programa PREPROCESADOR en el manejador A:
2. Dentro de Windows, seleccionar la opción Archivo, Ejecutar.
3. Teclear en el cuadro de dialogo A:\SETUP y dar <ENTER>.

Posteriormente se entra al programa instalador y se debe continuar con la secuencia que envíe el programa. Al terminar de instalar se crea un grupo de trabajo y dentro de éste un ícono que simboliza la aplicación de PREPROCESADOR. Para iniciar la ejecución del Preprocesador, seleccionar el ícono con el mouse y dar doble click, inmediatamente aparece la pantalla inicial (fig. 5.1).

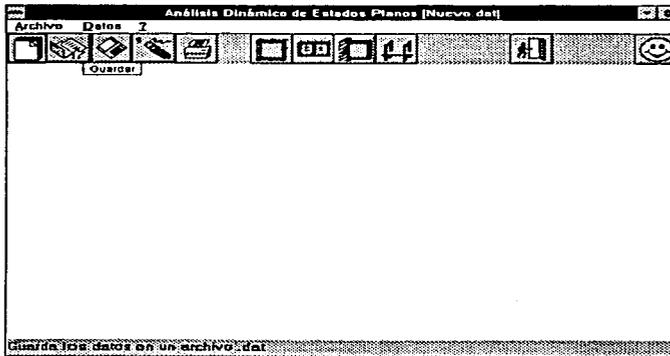


Fig 5.1

La pantalla inicial presenta una barra de títulos con el nombre del programa y el nombre del archivo que se esté utilizando; una barra de menús desplegables con tres opciones: Archivo, Datos y ?, cada uno de ellos presenta subopciones y pueden ser accedidos presionando la tecla ALT y la letra subrayada de cada opción o señalando cada opción

directamente con el puntero del mouse y una barra de botones en la cual pueden ser accedadas directamente todas las opciones. En la parte inferior de la pantalla se presenta una línea de texto que indica la acción que realiza cada opción (fig. 5.2)



Fig. 5.2

En el menú de Archivo se presentan las opciones para manejo de archivos, la opción **NUEVO** inicializa las pantallas de datos para capturar un nuevo archivo de datos. La opción **ABRIR** abre un archivo existente en disco y que ha sido creado por el Preprocesador. Al acceder a esta opción se abre el cuadro de dialogo comun de Windows para abrir archivos en el cual se puede seleccionar la unidad de disco, buscar en los diferentes subdirectorios, presentar archivos por tipo de extensión y seleccionar el archivo deseado (fig 5.3)

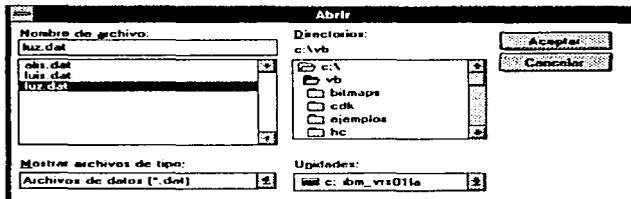


Fig. 5.3

Un cuadro similar aparece para las opciones de **GUARDAR** y **BORRAR** pero realiza la acción solicitada. En el caso de la opción **IMPRIMIR** se presenta un cuadro de diálogo diferente en el que aparecen opciones referentes a la impresión de datos (fig. 5.4)

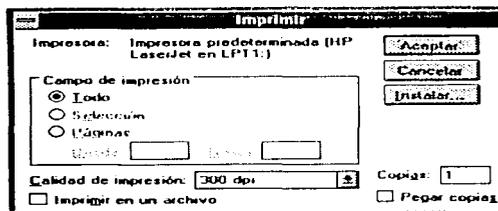


Fig. 5.4

El cuadro presenta la opción de mandar los datos a la impresora o a un archivo para su posterior impresión, este archivo tendrá la extensión **.PRN**.

Dentro de esta ventana también es posible configurar la impresora y el tipo de papel que se desee utilizar, así como la resolución de impresión. Estos parámetros dependerán de la impresora que esté especificada en el Panel de Control de Windows (fig. 5.5)

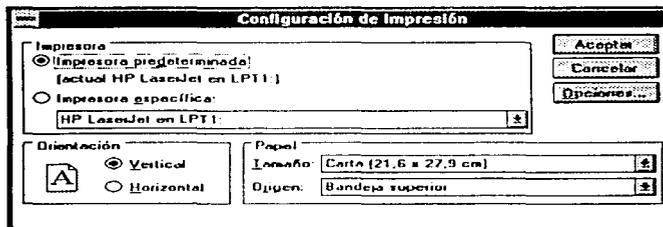


Fig. 5.5

Las opciones del Menú de **DATOS** son **Puntos Nodales**, **Elementos**, **Condiciones de Frontera** y **Materiales** que son los datos que se capturan en el sistema para el análisis de la estructura.

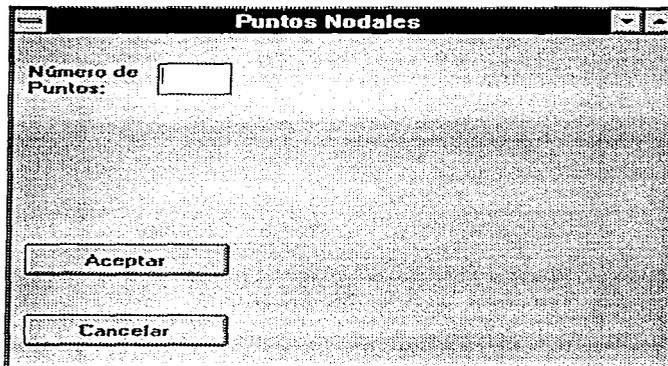
Cada opción puede ser accedida colocando el puntero del mouse y haciendo click en la opción, o también con las teclas ALT y la letra subrayada de la opción. Otra forma de acceder a las opciones es mediante la barra de botones en la cual se incluyen los botones correspondientes a Puntos Nodales, Elementos, Condiciones de Frontera y Materiales (fig. 5.6)



Fig. 5.6

Cuando se crea un archivo nuevo de datos, al acceder a las opciones el sistema solicita el número de Puntos Nodales, el número de Elementos y el número de Materiales que se utilizarán para el análisis de la estructura. El número de Condiciones de Frontera será igual al número de Puntos Nodales por lo que el sistema los genera automáticamente (fig. 5.7)

Una vez dado el valor inicial, el sistema despliega una tabla con columnas y renglones en donde se introducirán los datos de cada opción (figuras 5.8, 5.9, 5.10 y 5.11)



The image shows a graphical user interface window titled "Puntos Nodales". It features a text input field labeled "Número de Puntos:" with a small rectangular box next to it. Below the input field are two buttons: "Aceptar" and "Cancelar". The window has a standard title bar with a close button on the right.

Fig. 5.7

Puntos Nodales

Número de Puntos:

No.	X	Y
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		

Fig. 5.8

Elementos

Número de Elementos:

Elem.	Nodo I	Nodo J	Nodo K	Nodo L	Nodo M	Nodo N	Nodo O	Nodo P	Material
1									
2									
3									
4									
5									

Fig. 5.9

Condiciones de Frontera

Número de Condiciones:

No.	DX	DY
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0

Fig. 5.10

Materiales

Materiales:

No.	E	V	ESPESOR	P.V.
1				
2				
3				
4				

Fig 5.11

El botón de **Aceptar** se utiliza para cerrar la ventana cuando se termina de introducir datos y el botón de **Cancelar** para borrar los datos contenidos en la tabla. Cada ventana puede ser accedada individualmente pero solo se visualizará una a la vez, y en caso de ver un archivo existente las ventanas se desplegarán con los datos contenidos en el archivo.

Por último se presenta la opción **Acerca de** la cual presenta una ventana con información de los autores del programa (fig. 5.12)

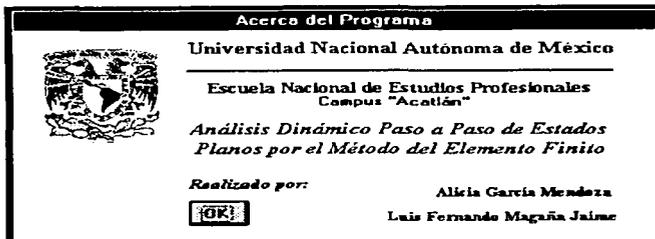


Fig. 5.12

Para terminar de utilizar el programa Preprocesador se debe de seleccionar la opción del **Salir** del menú principal o seleccionando el botón de salir de la barra de botones.

Existen algunas consideraciones que deben tomarse en cuenta antes de salir del sistema. Debe haberse grabado la información, de lo contrario envía un mensaje de advertencia (fig. 5.13)



Fig. 5.13

En caso de seleccionar la opción **Sí**, se presenta la ventana para guardar archivos y después sale del sistema, en caso de seleccionar la opción **No**, inmediatamente sale del sistema y la información que existía en memoria se pierde, la opción **Cancelar**, cancela el mandato de salir del programa y continúa su ejecución.

6. PROCESADOR 1

El procesador es el segundo programa del sistema, encargado de procesar o manipular datos por medio de cálculos matemáticos.

6.1 GENERALIDADES

El procesador 1, denominado "PROCMEF8" fue realizado en lenguaje FORTRAN ya que maneja los cálculos matemáticos por medio del Método del Elemento Finito, basándose en los datos contenidos en el archivo .DAT generado por el preprocesador.

Este sistema calcula las matrices de masas y de rigideces mediante el procedimiento de ensamble por lo que se genera un archivo que contiene las matrices anteriormente indicadas. En general el procedimiento de ensamble requiere de varios cálculos matemáticos que, dependiendo del tamaño de la estructura, en ocasiones se requiere memoria para poder almacenar esa información.

6.2 ORGANIZACIÓN

El procesador fue realizado en el lenguaje FORTRAN, por su capacidad y aproximación en el procesamiento de grandes cantidades de datos, el manejo de subrutinas que hacen más fácil la programación así como la rapidez que maneja para compilar y generar archivos ejecutables.

El programa en general consta de 20 subrutinas propias definidas por el programador, las cuales se describen a continuación:

Subrutina	Función
PROCMEF8	Es el programa principal donde se realiza la llamada a las demás subrutinas y calcula las direcciones del arreglo de memoria dinámica.
PRESENTA	Muestra en la pantalla el nombre de los autores y la versión del programa.
ABRIRA	Pide nombre del archivo y unidad donde se localiza para abrir o guardar según sea el caso.

CD Se encarga de calcular la matriz " D " la cual hace referencia a la matriz de estados planos de esfuerzos.

CROUT Resuelve el sistema de ecuaciones de orden $N*N$ por el método de Gauss-Crout.

DATINIC Lee el archivo proporcionado por el procesador *.dat, el número de puntos nodales, número de elementos, número de materiales, número de nodos cargados y número de las cargas.

ENSAM Ensambla la matriz de rigideces de la estructura, mediante las matrices de rigideces de cada elemento.

ESFUER8 Calcula los esfuerzos de cada elemento y de cada combinación de $c_{i,2}$ y de $n_{i,2}$.

FINAR Cierra el archivo.

FUERZA Genera el vector de cargas de la estructura

IMATR1 Imprime los datos de la matriz " A " (que es la que se le asigna como parámetro) en la pantalla.

IND Calcula los indicadores de ecuación del sistema, guardándolos en una matriz resultante.

LECAR Lee el valor del vector de datos del archivo especificado desde NI hasta NF-1

LECMAT Lee una matriz " A " con la dimensión NR número de renglones por NC número de columnas, pidiendo cada uno de los datos que la componen.

MATB8 Contiene las fórmulas de las funciones de forma, las fórmulas de los puntos gaussianos y demás para calcular la matriz B.

MATKES Calcula la matriz de rigideces de cada elemento.

MATK8 Con base en las subrutinas ENSAM y MATKES calcula y ensambla las matrices de rigideces de cada elemento para obtener la matriz de rigideces de la estructura.

MATRAN Calcula la transpuesta de la matriz que se le pasa como argumento.

MULMAT Realiza la multiplicación de las matrices que le son pasadas como argumentos, guardando el resultado en una tercera matriz.

REVISAR Revisa el dimensionamiento de la memoria dinámica, en caso de sobrepasar el dimensionamiento máximo, el programa se detiene.

6.3 MANUAL DEL USUARIO

El manejo del procesador 1 es sencillo por los pocos datos que usa, ya que el usuario solamente debe saber el nombre del archivo de datos con extensión .DAT el cual contiene los datos generados en el Preprocesador para que éste los tome y lleve a cabo sus cálculos.

El usuario debe teclear A:>PROCMEF8 y presionar ENTER para que en la primera pantalla aparezca el nombre de la Escuela, el nombre del programa y los nombres de los autores.

Finalmente pedirá en secuencia el drive donde se encuentra el archivo de datos *.DAT para poder abrirlo. En caso de algún error el programa lo marcará y detendrá su ejecución regresándose al sistema operativo (fig. 6.1)

```

C:\TESIS>PROCMEF8
PROGRAMA DEL METODO DEL ELEMENTO FINITO DE 8 PUNTOS
ELABORADO POR:
ALICIA GARCIA MENDOZA
LUIS FERNANDO MAGAÑA JAIME
VERSION MARZO-97
(Calcula la matriz de rigideces y de masas)

DRIVE PARA LEER LOS DATOS (A,B,C)
C

ARCHIVO DE DATOS
DAR EL NOMBRE DEL ARCHIVO DE DATOS SIN EXTENSION
(MAXIMO 8 CARACTERES)
CASO1

INDICADOR DE EJECUCION (0=NORMAL, 1=COMPLETO)
1

```

Fig. 6.1

Posteriormente el procesador indicará al usuario que introduzca la unidad donde será almacenado el archivo de resultados así como el nombre que a éste se le dará, después de introducir tanto la letra de la unidad como el nombre del archivo sin extensión hay que teclear "enter" para indicarle al programa que lo procese. De igual manera necesitará que se indique el drive y el nombre del archivo auxiliar donde guardará los resultados de las multiplicaciones de matrices para su empleo posterior.

En la figura 6.2, se muestra el procesador y el dimensionamiento usado. Dado que emplea un dimensionamiento dinámico es preciso mencionar al usuario la cantidad empleada en los cálculos pidiendo que presione "enter" para poder continuar.

```
NC =          40

INFORMACION :

DIMENSIONAMIENTO DE A=      18000
DIMENSIONAMIENTO USADO =    3332

PRESIONE RETURN PARA PODER CONTINUAR ...

MATRIZ DE RIGIDECES
CALCULA ELEMENTO NO.      1
CALCULA ELEMENTO NO.      2
CALCULA ELEMENTO NO.      3
CALCULA ELEMENTO NO.      4
MATRIZ DE MASAS
CALCULA ELEMENTO NO.      1
CALCULA ELEMENTO NO.      2
CALCULA ELEMENTO NO.      3
CALCULA ELEMENTO NO.      4

P R O C E S O   T E R M I N A D O . . .

C:>
```

Fig. 6.2

Finalmente el procesador realiza todos los cálculos necesarios para elaborar el archivo de resultados que el usuario le especificó que generará. Es importante mencionar que el tiempo de ejecución del programa depende de la magnitud del problema.

Por último el procesador terminará su ejecución al regresar al sistema operativo para que, si es deseo del usuario, se pueda consultar el archivo de resultados y el archivo auxiliar el cual contiene los resultados en forma de matrices para su mejor lectura.

7. PROCESADOR 2

7.1. GENERALIDADES

El sismógrafo es un aparato que registra la frecuencia y amplitud de los movimientos sísmicos en determinados puntos de la Tierra.

Los sismógrafos de movimiento fuerte están especialmente diseñados para registrar movimientos fuertes del suelo, de tal forma que los registros puedan ser leídos directamente como aceleración, velocidad o desplazamiento del suelo. Los sismógrafos más comunes son los que registran directamente la aceleración del suelo y son llamados acelerómetros o acelerógrafos.

Estos instrumentos normalmente se colocan de tal modo que registran directamente los movimientos de Tierra durante los sismos. Los sismógrafos medirán movimientos a lo largo de dos ejes horizontales y uno vertical. Las medidas más importantes son tres: aceleración, velocidad y desplazamiento. La aceleración cuando se multiplica por la masa, da la fuerza de inercia que debe resistir el medio continuo.

La velocidad se refiere al cambio del movimiento del suelo. El desplazamiento indica la distancia de una partícula que se mueve de su posición de reposo.

El término aceleración es fundamental cuando se trata de medir el movimiento variable del suelo, por lo que se puede pensar que cada parte de las ondas de un terremoto está asociada con una cierta aceleración del suelo. La aceleración debe ser expresada en función de un valor familiar. Esta es la aceleración debida a la gravedad, es decir, la aceleración con la que cae un objeto que es abandonado en el vacío ($g = 9.80 \text{ m/s}^2$; donde $g =$ aceleración de la gravedad). Esta es una porción bastante grande del incremento de la velocidad.

El concepto de la aceleración del movimiento sísmico es importante, de igual manera el de los efectos vibratorios que requiere un conocimiento de la velocidad, desplazamiento del suelo y las propiedades de las ondas. Cuando más lejos viajan las ondas, mayor es la atenuación de las ondas de alta frecuencia en comparación con las de largo periodo. Los acelerógrafos suministran, actualmente, cientos de registros de movimientos sísmicos, fuera y en el interior de los medios continuos.

El acelerograma proporciona una imagen del movimiento del suelo; la interpretación exacta de esta imagen requiere conocimientos y experiencia. En el acelerograma, el arribo de una onda "P" inicia el movimiento. A esta onda le sigue otra que se denomina "S". El tiempo que transcurre entre los dos movimientos permite calcular la distancia desde el instrumento hasta el foco del sismo, por lo que se puede ver la duración del movimiento fuerte y se puede medir la amplitud de onda, mientras que la aceleración del suelo se puede calcular relacionando la amplitud con el tiempo. La velocidad y el desplazamiento se obtienen matemáticamente integrando una y dos veces el registro respectivamente.

7.2. ORGANIZACIÓN

El programa de computadora fue realizado por el Lic. en Matemáticas Aplicadas y Computación Oscar Alejandro González Bustamante como trabajo de investigación, por lo tanto el enfoque será exclusivamente el manejo del sistema por que lo que interesan los valores convertidos en código ASCII, los cuales se necesitan para realizar el proceso respectivo en el siguiente Procesador.

El sistema consta de un programa principal llamado ACEGRAF y cinco subprogramas que son:

Subrutina	Función
ARCHIVO	En este módulo se realiza todo lo referente a los archivos como es almacenamiento en disco, desplegar directorio, cargar archivos, empaclar y desempacar archivos.
GRAFICA	Aquí se producen las gráficas de los acelerogramas con las escalas y la información del archivo, máximos y mínimos.
AGRANDA	Se producen las gráficas en pantalla de los acelerogramas pero con duración acotada en una escala mayor.
IMPRIME	Manda a la impresora la gráfica generada.
SALIDA	Termina la ejecución del programa.

7.3 MANUAL DEL USUARIO

Para ejecutar el programa del acelerograma se debe teclear desde la línea de comandos la instrucción ACEGRAF. La primera pantalla presenta un cuadro de texto conteniendo el nombre del programa, la versión y el año de creación; presionando cualquier tecla desaparece el cuadro y presenta el menú de opciones el cual siempre está presente en la pantalla y consta de cinco opciones que a su vez cada una presenta subopciones en forma de ventanas colgantes. (fig. 7.1). Abajo del menú principal se encuentra el área donde se grafican los acelerogramas (ventana de graficación) y en la parte inferior de la pantalla aparece una barra de ayuda la cual contiene información de diferentes órdenes asociadas a las teclas de función o combinación de teclas.



Fig. 7.1

Para acceder a las opciones del sistema basta con posicionar el cursor en la opción deseada usando las flechas de navegación y después presionar la tecla <ENTER>, o también se puede acceder presionando la primera letra que describe dicha tarea (por ejemplo, en el menú principal, para entrar a la opción Imprime sólo se debe presionar la tecla I).

También el programa proporciona una ayuda en línea la cual puede ser accesada colocándose en la opción de la cual se desea información y se presiona la tecla F1. (fig. 7.2)

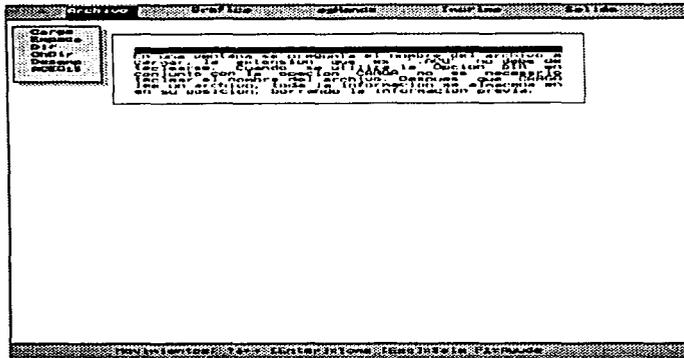


Fig. 7.2

Es importante saber que cuando se está dentro de un determinado menú y se desea salir de él, es posible hacerlo presionando la tecla <ESC>.

El sistema también proporciona teclas de referencia rápida, las cuales permitirán realizar un salto directo de un menú a otro sin necesidad de utilizar las flechas de navegación. Las combinaciones de teclas son las siguientes:

Tecla o Combinación	Función
F1	Ayuda de comando.
Alt + F1	Ayuda teclas disponibles para menús.
Alt + A	Menú de Archivo.
Alt + G	Menú de Gráfica.
Alt + R	Agranda.
Alt + P	Imprime.
Alt + X	Salir del sistema.
F3	Cargar un archivo existente.

7.3.1 ENTRADA DE DATOS

Como ya se explicó al principio, el programa tiene como fin, el graficar los registros de un acelerógrafo y producir los acelerogramas, un acelerógrafo capta el movimiento de aceleración del suelo en tres direcciones que son: vertical, Norte-Sur y Este-Oeste. Este programa utiliza los datos aportados por el CIRES (Centro de Instrumentación y Registro Sísmico) captados en su red de estaciones con acelerógrafos digitales con registro magnético en cassette. Estos registros son grabados en diskette en archivos ASCII.

El menú ARCHIVO contiene las opciones para alimentar datos al sistema, así como para la conversión y manejo de archivos. Dado que los archivos proporcionados por el CIRES contienen los datos en un formato ASCII o texto, es necesario realizar una conversión de estos datos a formato de tipo entero, para así reducir el espacio de memoria que ocuparía el formato ASCII, esto se realiza mediante la opción EMPACA. Estos registros convertidos de formato ASCII a enteros, son almacenados en archivos con extensión .ACG, los cuales están listos para su graficación. Estos archivos con extensión .ACG se denominan archivos empacados, los cuales se cargarán en la memoria principal, con la opción CARGA. En ambos casos aparecen ventanas con mensajes al usuario, indicándole si el proceso va efectuándose correctamente.

Dado que el siguiente Procesador lee los datos en formato ASCII, es necesario restablecer los archivos a su formato original, esta acción se realiza ejecutando la opción DESEMP, en donde se convertirá un archivo con datos de tipo entero a un archivo con formato ASCII.

La opción DIR sirve para visualizar los archivos con extensión .ACG en el directorio en uso. Para cambiar de directorio deberá utilizarse la opción CHDIR.

7.3.2 GRAFICACION

Antes de realizar la graficación de un acelerograma es necesario cargar en memoria RAM el archivo que contiene los registros del acelerógrafo, al cual se le aplicó el proceso de conversión de formato ASCII a formato de tipo entero y que tiene la extensión .ACG.

El menú GRAFICA contiene las opciones para la graficación de los acelerogramas en pantalla, las opciones son las siguientes:

3-COMP: Produce en pantalla los acelerógramas correspondientes a las tres componentes del movimiento; vertical, Norte-Sur y Este-Oeste. Al pie de los acelerogramas se muestra la información relacionada a estos; el nombre del archivo, la fecha y hora del evento, los máximos y mínimos registrados en cada componente ortogonal y el máximo absoluto de los acelerogramas. Cada componente tiene su escala de aceleración en gals en el eje Y y en el eje X la escala del tiempo en segundos. (fig. 7.3)

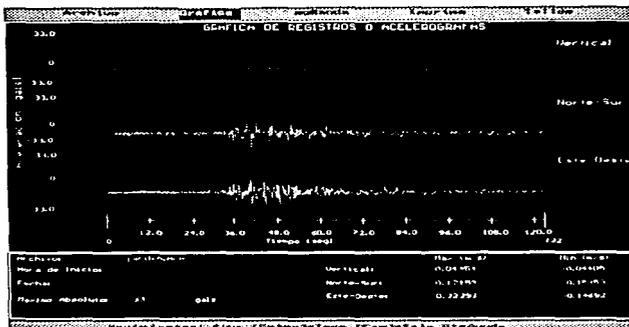


Fig. 7.3

VERT, N-S, E-O: El propósito de estas opciones es producir el despliegue por separado de cada acelerograma: VERT para componente vertical, N-S para la componente Norte-Sur y E-O para la componente Este-Oeste. El área de trabajo o despliegue es ocupada por el acelerograma correspondiente y en la parte inferior aparece la misma información de la opción anterior.

Existe otra opción del menú principal llamada **AGRANDA** que despliega en pantalla los acelerogramas de cada componente, pero además hace un escalamiento en el eje Y, o de aceleración y permite ajustar el rango de graficación en el eje X, o tiempo, con un incremento definido por el usuario. Esta opción sirve para visualizar la parte crítica donde se registraron las aceleraciones más violentas, proporcionándole al usuario una visión en detalle de la gráfica.

7.3.3 IMPRESION

Esta opción se utiliza para imprimir la gráfica del acelerógrama que esté en pantalla en ese momento, se debe asegurar que la impresora esté en línea antes de ejecutar esta opción.

7.3.4 SALIDA

Esta opción da por terminada la sesión con el programa y regresa al indicador del sistema operativo, antes de salir aparece un cuadro de diálogo que permite confirmar el término de la sesión al oprimir S o regresar con una N.

8. PROCESADOR 3

8.1. GENERALIDADES

El Procesador 3 será el encargado de aplicar el Método Beta de Newmark, este programa fue realizado en lenguaje FORTRAN por ser un compilador que maneja grandes cantidades de datos y por su exactitud al realizar los cálculos.

El programa lee el archivo donde se encuentran las matrices de masas y rigideces, calculadas por el Procesador 1, lee el archivo donde se encuentra la información del acelerógrafo y crea el archivo de resultados, posteriormente lee las constantes α , β , γ y μ para realizar los cálculos.

El proceso comienza calculando las constantes $E0$, $E1$, $EK1$, $EK2$, $EA1$ y $EB1$, luego se calcula la matriz K , multiplicando la constante $EK1$ por la matriz de masas y sumándola a la multiplicación de la constante $EK2$ por la matriz de rigideces. Se inicializan los vectores de velocidad y desplazamiento con cero y se comienza con el cálculo para cada valor del acelerograma para calcular el desplazamiento, velocidad y aceleración de los nodos del medio continuo. Se inicializa un contador hasta el número de valores del acelerógrafo y se calculan los vectores \underline{a} y \underline{b} para cada aceleración, estos vectores se utilizarán para el cálculo del vector \underline{x} .

Se resuelve la ecuación (3.29) usando el Método de Gauss-Crout, obteniendo las aceleraciones de los nodos del medio continuo y por último se calculan las velocidades y desplazamientos de los nodos; este proceso se repite para el número de puntos que se tenga en la lectura del acelerógrafo. Para finalizar el programa se calculan las velocidades, aceleraciones y desplazamientos promedio de cada nodos del medio continuo.

El programa cuenta con tres indicadores de ejecución los cuales determinarán el contenido del archivo de resultados; el primero realiza la impresión de todo el proceso de cálculo en el cual se incluyen los datos de las matrices de masas y rigideces, así como la matriz resultante K , también se incluyen los valores de los los vectores \underline{a} , \underline{b} y auxiliares para cada punto del acelerograma, la historia de los desplazamientos, velocidades y aceleraciones en cada iteración y por último los desplazamientos, velocidades y aceleraciones promedio. El segundo incluye solamente las historias de los desplazamientos, velocidades y aceleraciones de cada iteración y promedios de cada uno. La opción tres

unicamente imprime los resultados de los desplazamientos, velocidades y aceleraciones promedio.

8.2 ORGANIZACIÓN

El Procesador 3 fue realizado en lenguaje FORTRAN y está constituido por un programa principal y 6 subrutinas. El programa principal tiene como objetivo llamar a las demás subrutinas, leer los archivos que contienen los datos de las matrices de masas y rigideces, leer el archivo con los registros del acelerograma y genera el archivo secuencial de salida con los resultados finales.

Las subrutinas y sus funciones son las siguientes:

Subrutina	Función
PROCBETA	Es el programa principal y es el que realiza la llamada a las demás subrutinas y calcula las direcciones del arreglo de memoria dinámica, además lee el valor de las constantes α , β , γ y μ .
PRESENTA	Muestra en la pantalla el nombre de los autores y la versión del programa.
ABRIRA	Abre el archivo con las matrices de masas y rigideces generado en Procesador 1 y el archivo del acelerograma; también crea el archivo de resultados del Método beta de Newmark.
REVISA	Revisa el dimensionamiento de la memoria dinámica, en caso de sobrepasar el dimensionamiento máximo el programa se detiene.
BETAN3	Lleva a cabo la aplicación del Método auxiliándose de las subrutinas IMAAR, TGCSIM, MULTMAT y SGCSIM, y con éstas se imprimen los resultados de las matrices y los vectores auxiliares.
IMAAR	Manda a impresión en el archivo de resultados los valores calculados en cada paso del método.
TGCSIM	Realiza la triangulación de la matriz cuadrada por el método de Gauss-Crout.

SGCSIM	Realiza las sustituciones en el método de Gauss-Crout.
MULMAT	Realiza las multiplicaciones de las matrices de masas y rigideces por el vector de carga \underline{a} y la multiplicación de la matriz de rigideces por la suma de los vectores \underline{a} y \underline{b} para obtener el vector \underline{x} .
IMPRES1	Imprime en el archivo de resultados el valor de las constantes utilizadas para el método.
IMPRES2	Imprime en el archivo de resultados las matrices de masas y rigideces y la matriz de solución \underline{K} .
IMPRES3	Imprime en el archivo de resultados el valor de los vectores \underline{a} , \underline{b} y auxiliares para cada punto del acelerógrama.
IMPRES4	Imprime en el archivo de resultados la historia de los desplazamientos, velocidades y aceleraciones en cada iteración así como los vectores promedios de cada uno.

8.3 MANUAL DEL USUARIO

Para ejecutar el programa del Procesador 3 es necesario escribir PROCBETA desde la unidad donde se encuentre (A: o C:). Al cargar el programa aparece el nombre del programa y la versión del mismo, así como el nombre de los autores; inmediatamente solicita el nombre del archivo de datos con extensión .MRI donde están almacenadas las matrices de rigideces y de masas producidas por el Procesador 1 y el drive donde se encuentra; luego solicita el nombre del archivo que contiene la información del acelerograma y su ubicación, además debe de especificarse el nombre del archivo donde se guardaran los resultados del programa. (fig. 8.1)

El programa tiene 3 niveles de ejecución dependiendo de los resultados que deseen generarse: la opción 1 genera en un archivo todo el proceso de cálculo del método, la opción 2 genera sólo la historia de cada paso, así como los promedios de aceleración, velocidad y desplazamiento y la opción 3 únicamente genera los promedios finales antes mencionados; luego revisa el dimensionamiento de la memoria dinámica varias veces para verificar que no exceda el tamaño máximo

del arreglo y en caso contrario detener el programa. (fig. 8.2). Por último solicita el valor de las constantes α , β , y μ para comenzar a calcular los vectores de desplazamiento, velocidad y aceleración para cada punto del acelerógrama. (fig.8.3).

El procesamiento de los datos depende del número de ecuaciones que se tengan que resolver y el número de puntos del acelerograma. Los datos obtenidos se almacenan en el archivo de resultados para que posteriormente el usuario pueda interpretarlos.

```
C:\TESIS>PROCBETA
PROGRAMA GENERALIZADO BETA DE NEWMARK
ELABORADO POR:
ALICIA GARCIA MENDOZA
LUIS FERNANDO MAGAÑA JAIME
VERSION MARZO-97

ARCHIVO DE DATOS
DAR EL NOMBRE DEL ARCHIVO
DE LA MATRICES DE RIGIDECES Y MASAS
(MAXIMO 8 CARACTERES PUNTO EXTENSION)
ARCHIVO.MRI
UNIDAD PARA LEER LOS DATOS (A,B,C,)
C
DAR EL NOMBRE DEL ARCHIVO DEL ACELEROGRAMA
(MAXIMO 8 CARACTERES PUNTO EXTENSION)
ACELERO.ACG
UNIDAD PARA LEER LOS DATOS (A,B,C,)
C
DAR EL NOMBRE DEL ARCHIVO DE RESULTADOS
(MAXIMO 8 CARACTERES PUNTO EXTENSION)
RESULTA.DVA
UNIDAD PARA LEER LOS DATOS (A,B,C,)
C
```

Fig. 8.1

INDICADOR DE EJECUCION
OPCIONES

1=IMPRESION TODO
2=IMPRESION HISTORIA DESPLAZAMIENTOS, VELOCIDAD Y
ACELERACION
3=IMPRESION PROMEDIO DESPLAZAMIENTOS, VELOCIDAD Y
ACELERACION
1

INFORMACION :

DIMENSIONAMIENTO DE A= 50000
DIMENSIONAMIENTO USADO = 3200

PRESIONE RETURN PARA PODER CONTINUAR ...

INFORMACION :

DIMENSIONAMIENTO DE A= 50000
DIMENSIONAMIENTO USADO = 3210

PRESIONE RETURN PARA PODER CONTINUAR ...

Fig. 8.2

DAR LA CONSTANTE ALFA =
0.0

DAR LA CONSTANTE MIU =
0.0

DAR LA CONSTANTE GAMA =
0.5

DAR LA CONSTANTE BETA =
0.25

CALCULO DE CONSTANTES

CALCULO DE MATRICES

TRIANGULACION

NO. DE PUNTO	1
NO. DE PUNTO	2
NO. DE PUNTO	3
NO. DE PUNTO	4
NO. DE PUNTO	5
NO. DE PUNTO	6
NO. DE PUNTO	7
NO. DE PUNTO	8
NO. DE PUNTO	9
NO. DE PUNTO	10

PROCESO TERMINADO
C:>

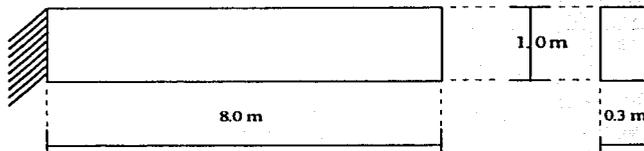
Fig. 8.3

9. EJEMPLO DE APLICACION

Con la finalidad de mostrar el uso del sistema se presenta un problema específico.

9.1 ESTRUCTURA EJEMPLO

Sea una viga empotrada con las siguientes características como se indica en la figura 9.1



(Fig. 9.1) Viga empotrada en la pared

Los datos de la estructura son los siguientes:

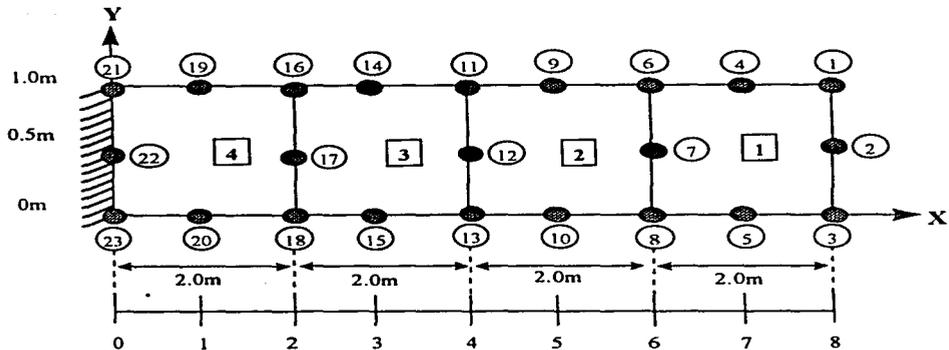
$$\begin{aligned} E &= 2.0 \times 10^6 \text{ t/m}^2 \\ \gamma &= 2.4 \text{ ton/m}^3 \\ \text{Espesor} &= 0.3 \text{ m} \\ \nu &= 0.2 \end{aligned}$$

Se realizan análisis de medios continuos para resolver las incógnitas de equilibrio entre esfuerzo-deformación, por tal motivo se recurre al uso del método del elemento finito. El siguiente paso a seguir es analizar la estructura por medio de una discretización de la misma en elementos cuadráticos (8 puntos nodales) para poder introducir los datos que el preprocesador ha de requerir.

9.2 PROCESO DE CALCULO

Con base en los datos indicados se efectúa la construcción de la malla, se numeran los puntos nodales y los elementos como se indica en la figura 9.2, se obtienen las coordenadas de los puntos nodales posteriormente y se asocian los elementos a los puntos nodales correspondientes, de esta manera se le

dan los datos al PREPROCESADOR como se indica en las siguientes figuras:



(Fig. 9.2) Estructuración en puntos nodales de la viga empotrada en la pared

Puntos Nodales					
Numero de Puntos:	23	No.	X	Y	
		1	8.00	1.00	
		2	8.00	0.50	
		3	8.00	0.00	
		4	7.00	1.00	
		5	7.00	0.00	
		6	6.00	1.00	
		7	6.00	0.50	
		8	6.00	0.00	
		9	5.00	1.00	
		10	5.00	0.00	
		11	4.00	1.00	
		12	4.00	0.50	

Buttons: Aceptar, Cancelar

(Fig. 9.39) A Puntos Nodales del 1 al 12 de la viga empotrada

Puntos Nodales				
Número de Puntos:	23	No.	X	Y
		12	4.00	0.50
		13	4.00	0.00
		14	3.00	1.00
		15	3.00	0.00
		16	2.00	1.00
		17	2.00	0.50
		18	2.00	0.00
		19	1.00	1.00
		20	1.00	0.00
		21	0.00	1.00
		22	0.00	0.50
		23	0.00	0.00

(Fig. 9.3 B) Puntos Nodales del 12 al 23 de la viga empotrada

Elementos									
Número de Elementos:	4								
Elem.	Nodo I	Nodo J	Nodo K	Nodo L	Nodo M	Nodo N	Nodo O	Nodo P	Material
1	8	5	3	2	1	4	6	7	1
2	13	10	8	7	6	9	11	12	1
3	18	15	13	12	11	14	16	17	1
4	23	20	18	17	16	19	21	22	1

(Fig. 9.4) Elementos de la viga empotrada

Condiciones de frontera

Número de Condiciones:

No.	DX	DY
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0

(Fig. 9.5) Condiciones Frontera de la viga empotrada

Condiciones de frontera

Número de Condiciones:

No.	DX	DY
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	1	1
22	1	1
23	1	1

(Fig. 9.6) Condiciones Frontera del 12 al 23 de la viga empotrada

Materiales				
No.	E	V	ESPESOR	P.V.
1	2000000.00	0.20	0.30	2.4

Número de Materiales:

(Fig. 9.7) Materiales de la viga empotrada

El preprocesador genera un archivo secuencial de datos en este caso se llama CAS01.DAT

A continuación se ejecuta el procesador 1 y se obtiene un archivo CAS01.MRI con las matrices de rigideces y masas.

Se ejecuta el procesador 2 y se discretiza un acelerograma de acuerdo al incremento de tiempo considerado, generando el archivo CAS01.ACG con los siguientes datos:

Donde los valores son dados en m/s^2

No. de Puntos =	10.00
Incrementos de tiempo =	0.02
1.-	0.00
2.-	0.4340
3.-	0.5930
4.-	0.0830
5.-	-0.8030
6.-	0.0150
7.-	0.0840
8.-	-0.4150
9.-	-0.3980
10.-	0.1750

Con los archivos generados del procesador 1 y 2 se ejecuta el procesador 3 utilizando las constantes $\alpha=0$, $\nu=0$, $\gamma=0.5$ y $\beta=0.25$, generando el archivo CAS01.DVA el cual muestra los desplazamientos, velocidades y aceleraciones promedio.

9.3 RESULTADOS

En la tabla se muestran los desplazamientos, velocidades y aceleraciones promedio obtenido de acuerdo a los datos considerados:

NODO	COORD	DESPLAZAMIENTOS	VELOCIDADES	ACELERACIONES
1	X	3.343372E-04	1.373873E-02	1.603741E+00
	Y	3.474865E-03	5.076024E-02	3.522164E+00
2	X	8.620819E-05	1.039040E-02	1.712874E+00
	Y	3.474998E-03	5.073907E-02	3.526117E+00
3	X	3.387297E-04	1.524014E-02	2.375552E+00
	Y	3.474816E-03	5.070785E-02	3.526138E+00
4	X	3.313176E-04	1.334304E-02	1.553983E+00
	Y	2.856073E-03	3.884194E-02	1.995344E+00
5	X	3.351191E-04	1.471539E-02	2.301337E+00
	Y	2.856052E-03	3.882180E-02	1.994044E+00
6	X	3.187493E-04	1.236657E-02	1.484373E+00
	Y	2.260592E-03	3.472138E-02	2.049078E+00
7	X	7.965187E-05	9.584738E-03	1.586069E+00
	Y	2.260034E-03	3.473154E-02	2.052482E+00
8	X	3.197234E-04	1.289625E-02	2.061174E+00
	Y	2.260478E-03	3.461813E-02	2.034514E+00
9	X	2.941370E-04	1.107795E-02	1.442327E+00
	Y	1.705155E-03	3.571142E-02	2.894124E+00
10	X	2.891108E-04	9.369841E-03	1.571274E+00
	Y	1.704963E-03	3.559007E-02	2.876284E+00
11	X	2.611080E-04	1.011778E-02	1.419783E+00
	Y	1.201751E-03	3.464887E-02	3.147621E+00
12	X	6.187068E-05	7.414956E-03	1.231199E+00
	Y	1.201171E-03	3.481199E-02	3.166541E+00
13	X	2.508960E-04	5.973547E-03	1.069047E+00
	Y	1.201518E-03	3.454455E-02	3.132635E+00
14	X	2.210708E-04	9.533278E-03	1.319115E+00
	Y	7.557349E-04	2.784585E-02	2.572376E+00
15	X	2.091243E-04	4.860373E-03	7.422763E-01
	Y	7.553598E-04	2.771360E-02	2.553431E+00
16	X	1.688954E-04	8.186400E-03	1.020943E+00
	Y	3.830178E-04	1.686771E-02	1.526488E+00
17	X	3.478656E-05	4.142995E-03	6.794280E-01
	Y	3.789952E-04	1.677356E-02	1.517866E+00
18	X	1.600215E-04	5.307325E-03	6.486201E-01
	Y	3.828122E-04	1.680963E-02	1.521978E+00
19	X	9.728112E-05	5.145475E-03	5.617324E-01
	Y	1.137342E-04	5.799003E-03	5.146389E-01
20	X	9.331691E-05	4.016739E-03	4.479893E-01
	Y	1.133766E-04	5.699678E-03	5.121045E-01

9.4 REFINADO DE MALLA (CASO 2)

Tomando como base la misma estructura se realizó otro caso, en el cual se dividió cada elemento en dos de tal forma que la malla fuese más fina como se muestra en la figura 9.9.

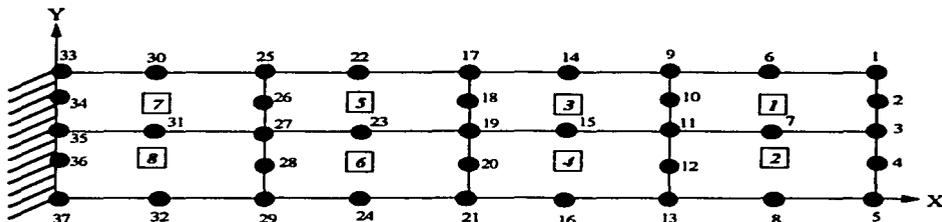


Fig. 9.9

Utilizando los mismos datos del acelerograma y las mismas constantes para el Método β de Newmark (Capítulo 3) del primer ejemplo, obtenemos la siguiente tabla donde se muestran los desplazamientos, velocidades y aceleraciones promedio para el caso 2.

NODO	COORD	DESPLAZAMIENTOS	VELOCIDADES	ACELERACIONES
1	X	3.373110E-04	1.342389E-02	1.573320E+00
	Y	3.538735E-03	5.027810E-02	3.458243E+00
2	X	1.832662E-04	1.092976E-02	1.562551E+00
	Y	3.539232E-03	5.025841E-02	3.456228E+00
3	X	8.679923E-05	1.046283E-02	1.723049E+00
	Y	3.539644E-03	5.023574E-02	3.453219E+00
4	X	1.884841E-04	1.219079E-02	2.006796E+00
	Y	3.539210E-03	5.024443E-02	3.457314E+00
5	X	3.430936E-04	1.547026E-02	2.379574E+00
	Y	3.538691E-03	5.024987E-02	3.460452E+00
6	X	3.346246E-04	1.308557E-02	1.532180E+00
	Y	2.912146E-03	3.850125E-02	1.968542E+00
7	X	8.443156E-05	1.017179E-02	1.677256E+00
	Y	2.911405E-03	3.848207E-02	1.967328E+00
8	X	3.395736E-04	1.491912E-02	2.302874E+00
	Y	2.912125E-03	3.847121E-02	1.966812E+00

9	X	3.227507E-04	1.221619E-02	1.476081E+00
	Y	2.306435E-03	3.447349E-02	2.035515E+00
10	X	1.747910E-04	1.020755E-02	1.482292E+00
	Y	2.306653E-03	3.451966E-02	2.042083E+00
11	X	8.005941E-05	9.634259E-03	1.593427E+00
	Y	2.307124E-03	3.453455E-02	2.043367E+00
12	X	1.761954E-04	1.067373E-02	1.788751E+00
	Y	2.306597E-03	3.447064E-02	2.035018E+00
13	X	3.245526E-04	1.307624E-02	2.063130E+00
	Y	2.306322E-03	3.437566E-02	2.021423E+00
14	X	2.983664E-04	1.099668E-02	1.440679E+00
	Y	1.742005E-03	3.563001E-02	2.874618E+00
15	X	7.207401E-05	8.658266E-03	1.434559E+00
	Y	1.740443E-03	3.566967E-02	2.881837E+00
16	X	2.939076E-04	9.518448E-03	1.578190E+00
	Y	1.741824E-03	3.551109E-02	2.857507E+00
17	X	2.654456E-04	1.012938E-02	1.427650E+00
	Y	1.227641E-03	3.462250E-02	3.122096E+00
18	X	1.450261E-04	8.673440E-03	1.325494E+00
	Y	1.227940E-03	3.478394E-02	3.139355E+00
19	X	6.246148E-05	7.484905E-03	1.242417E+00
	Y	1.228486E-03	3.483773E-02	3.143594E+00
20	X	1.359251E-04	6.538372E-03	1.155614E+00
	Y	1.227817E-03	3.472796E-02	3.131282E+00
21	X	2.553707E-04	5.975334E-03	1.062562E+00
	Y	1.227396E-03	3.451015E-02	3.105880E+00
22	X	2.250295E-04	9.568714E-03	1.321701E+00
	Y	7.741015E-04	2.797872E-02	2.562105E+00
23	X	4.936067E-05	5.900853E-03	9.770874E-01
	Y	7.710650E-04	2.795213E-02	2.558889E+00
24	X	2.130616E-04	4.800451E-03	7.432028E-01
	Y	7.737232E-04	2.784165E-02	2.542891E+00
25	X	1.732293E-04	8.293251E-03	1.026778E+00
	Y	3.904857E-04	1.687166E-02	1.513650E+00
26	X	9.282718E-05	5.881311E-03	8.288979E-01
	Y	3.898268E-04	1.692480E-02	1.518040E+00
27	X	3.519905E-05	4.191520E-03	6.871666E-01
	Y	3.911605E-04	1.700963E-02	1.525585E+00
28	X	8.425001E-05	3.803085E-03	6.090918E-01
	Y	3.897182E-04	1.689358E-02	1.515704E+00
29	X	1.641325E-04	5.293442E-03	6.450704E-01
	Y	3.902686E-04	1.680881E-02	1.508791E+00
30	X	1.000826E-04	5.228901E-03	5.672977E-01
	Y	1.183798E-04	5.921437E-03	5.204638E-01
31	X	1.808963E-05	2.146395E-03	3.475012E-01
	Y	1.123518E-04	5.642399E-03	4.979923E-01
32	X	9.590032E-05	4.026345E-03	4.489692E-01
	Y	1.179835E-04	5.811247E-03	5.174150E-01

CONCLUSIONES Y RECOMENDACIONES

En relación al desarrollo de software técnico, es decir, sistemas que resuelvan modelos matemáticos, se puede obtener la solución del mismo, pero en términos generales se tienen muchos caminos para elaborar el sistema, debido a las características del trabajo, se opta por trabajar en forma interdisciplinaria con ingenieros civiles y no limitar el sistema a un compilador específico, sino utilizar el compilador adecuado según las características de cada programa.

En un intento de presentar en forma clara las conclusiones de este trabajo, se van a presentar en dos grupos: el primer grupo se refiere a los aspectos de relacionados con el desarrollo del sistema donde se presentan los juicios y recomendaciones para las personas que pretenden desarrollar este tipo de programas en forma adecuada y el segundo grupo con los resultados obtenidos de los ejemplos de aplicación para ser analizados por los especialistas en el área.

1.1 DESARROLLO DEL SOFTWARE

En relación al sistema:

a) Para el desarrollo de un sistema de tipo técnico se propone el siguiente plan de trabajo:

- Estudiar el modelo matemático y resolverlo a partir de un ejemplo.
- Conocer las necesidades operativas del usuario.
- Elaborar el esquema general del sistema, hacer el plan de trabajo.
- En caso de requerirse un lenguaje no conocido se efectúa la capacitación del mismo.
- El desarrollo propiamente del sistema.
- Documentación del paquete y manual del usuario.

b) Durante la etapa de desarrollo del sistema (número 5 del inciso a) se propone este plan de trabajo:

- El sistema se debe dividir en programas de acuerdo al compilador a utilizar, cada programa se divide por lo menos en 4 partes u objetos alcanzar.
 - De cada programa, después de llegar a la etapa considerada se debe presentar al usuario potencial para su evaluación y posibles cambios, no se debe pasar a otra etapa hasta que esté totalmente aceptada la que este en proceso.
- c) Al trabajar en equipo con un programa específico se deben considerar los siguientes aspectos.
- En caso de no conocer algún compilador el entrenamiento debe ser en forma conjunta.
 - En cada etapa se debe discutir el objetivo y respetar el tiempo de trabajo y el calendario asignado.
 - Una constante retroalimentación de información, se debe tener el mismo nivel de conocimientos, ante la opción de un cambio se debe primero discutir y luego tomar la decisión en forma conjunta.
 - Se debe tener un lugar adecuado de trabajo, no se recomienda en casa debido a que existen factores de distracción.
- d) De acuerdo a los requerimientos el sistema se dividió en 4 programas que son:

PREPROCESADOR

Es el programa que sirve para la captura de datos para generar el archivo de datos que serán utilizados para el procesador 1. Este programa fue diseñado en lenguaje Visual BASIC versión 3.0, el cual generó 1,480 líneas de código y se desarrolló en un período de 4 meses.

PROCESADOR 1

Es el programa que lee el archivo de datos generado por el Preprocesador y calcula las matrices de masas y rigideces además de generar el archivo de datos que será utilizado en el Procesador 3. Este programa fue diseñado en lenguaje FORTRAN, el cual generó 791 líneas de código y se desarrolló en un período de 2 meses.

PROCESADOR 2

Es el programa que genera el archivo con los datos del acelerograma para ser utilizados en el Procesador 3. Este programa fue diseñado en lenguaje PASCAL, el cual generó 4,296 líneas de código y se desarrolló en un periodo de 8 meses.

PROCESADOR 3

Es el programa que realiza el cálculo de los desplazamientos, velocidades y aceleraciones utilizando el método β de Newmark con la información generada en los procesadores 1 y 2. Este programa fue diseñado en lenguaje FORTRAN, el cual generó 389 líneas de código y se desarrolló en un periodo de 2 meses.

e) En relación al Visual BASIC se recomienda a quien desee desarrollar aplicaciones y desconozca su uso tomar un curso de introducción o en su defecto contar con un manual de referencia como material de apoyo y practicar al menos dos horas diarias en un máximo de 5 semanas.

f) El FORTRAN sigue siendo el paquete por excelencia para procesos matemáticos ; además sigue siendo compatible para cualquier tamaño de computadoras lo que permite utilizar varias subrutinas de apoyo y analizar casi cualquier tipo de estructura.

g) De acuerdo al cuadro comparativo y tomando en consideración que el Procesador 2 fue desarrollado como trabajo de investigación independientemente de este proyecto, el programa que se llevo más tiempo es el Preprocesador, debido a que se desconocía el lenguaje Visual BASIC, además de que contiene más líneas de código. El Procesador 1 y Procesador 3 fueron desarrollados en el mismo periodo de tiempo generando el primero más líneas de código dada la complejidad del método.

CUADRO COMPARATIVO

PROGRAMA	LENGUAJE	No. LÍNEAS	MEMORIA	TIEMPO DE DESARROLLO
PREPROCESADOR	VISUAL BASIC	1,480	49,726	4 meses
PROCESADOR-1	FORTRAN	791	130,204	2 meses
PROCESADOR-2	PASCAL	4,296	67,568	8 meses
PROCESADOR-3	FORTRAN	389	253,318	2 meses

1.2 RESULTADOS DE LOS EJEMPLOS NUMERICOS

Tomando como referencia los ejemplos utilizados para la demostración del sistema, se puede observar en la tabla comparativa los resultados obtenidos en cuanto a los desplazamientos en Y para los puntos nodales comunes, en cada una de las mallas propuestas se encontró que cada vez que se hace la malla más fina, se obtienen datos más aproximados a los desplazamientos reales de una estructura.

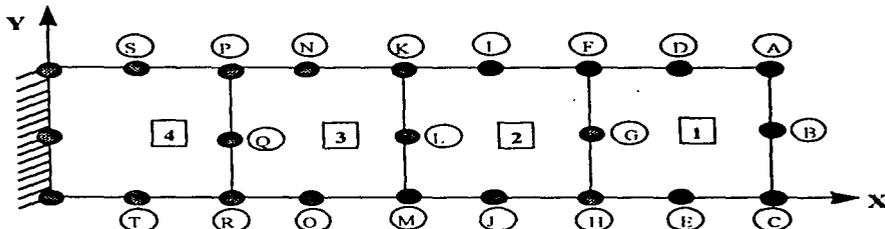


TABLA COMPARATIVA

NODO	CASO1	CASO2
A	3.474865E-03	3.538735E-03
B	3.474998E-03	3.539644E-03
C	3.474816E-03	3.538691E-03
D	2.856073E-03	2.912146E-03
E	2.856052E-03	2.912125E-03
F	2.260592E-03	2.306435E-03
G	2.260034E-03	2.307124E-03
H	2.260478E-03	2.306322E-03
I	1.705155E-03	1.742005E-03
J	1.704963E-03	1.741824E-03
K	1.201751E-03	1.227641E-03
L	1.201171E-03	1.228486E-03
M	1.201518E-03	1.227396E-03
N	7.557349E-04	7.741015E-04
O	7.553598E-04	7.737232E-04
P	3.830178E-04	3.904857E-04
Q	3.789952E-04	3.911605E-04
R	3.828122E-04	3.902686E-04
S	1.137342E-04	1.183798E-04
T	1.133766E-04	1.179835E-04

BIBLIOGRAFIA

1. C. Zienkiewicz, THE FINITE ELEMENT METHOD, Fourth Edition, Mc Graw-Hill, vol. I, II, 1989.
2. J. Bathe y E.L. Wilson, NUMERICAL METHOD IN FINITE ELEMENT ANALYSIS, Prentice-Hall, 1976.
3. L. Segerlind, APPLIED FINITE ELEMENT ANALYSIS, John Wiley, 1976.
4. S. Timoshenko y J. N. Goodier, TEORIA DE LA ELASTICIDAD, Segunda Edición, Ediciones Urmo, 1975.
5. Salazar Polanco, ESTABILIDAD DE LAS CONSTRUCCIONES, Representaciones y Servicios de Ingeniería, 1980.
6. W. Clough y J. Penzien, DYNAMICS OF STRUCTURES, Mc Graw Hill, 1975.
7. A. Ghali y A. Neville, ANALISIS ESTRUCTURAL, Primera Edición, Diana, 1980.
8. Robert H. Hammond y Willian B. Rogers, INTRODUCCION AL FORTRAN 77 Y LA PC, Mc Graw-Hill, 1988.
9. Oscar Alejandro González Bustamante, APLICACION DE LA COMPUTADORA PARA EL CONTROL DE REGISTRO, GRAFICACION, DISCRETIZACION E INTERPRETACION DE ACELEROGRAMAS, Universidad Nacional Autónoma de México, E.N.E.P. Acatlán, Tesis Profesional, 1991.
10. Ross P. Nelson, GUIA COMPLETA DE VISUAL BASIC PARA WINDOWS, Segunda Edición, Mc Graw Hill, 1993.
11. Grossman, I. Stanley, ALGEBRA LINEAL, Grupo Editorial Iberoamericano, 1990

APENDICE A

LISTADO DEL PREPROCESADOR

ARCHIVO MENU.FRM

```
Sub Abrir ()
  Dim Archivo As String
  Dim cPuntos As String
  Dim cElementos As String
  Dim cCondicion As String
  Dim cMaterial As String

  Puntos = 0
  nElementos = 0
  Condicion = 0
  nMateriales = 0

  Dlogo.CancelError = True
  Dlogo.Filename = ""
  Dlogo.DefaultExt = ".dat"
  Dlogo.Filter = "Archivos de datos (*.dat)|*.dat|Todos los archivos
(*.*)|*.*"
  Dlogo.FilterIndex = 1
  Dlogo.Flags = OFN_FILEMUSTEXIST Or OFN_PATHMUSTEXIST Or
OFN_HIDEREADONLY
  On Error GoTo cancela
  Dlogo.Action = 1

  If Dlogo.Filename = "" Then MsgBox "No se ha seleccionado ningun
archivo."

  Open Dlogo.Filename For Input As 1

  Do While Not EOF(1)
    Line Input #1, Archivo
    Line Input #1, cPuntos
    Line Input #1, cCondicion
    cCondicion = cPuntos
    Line Input #1, cMaterial
    Line Input #1, cElementos

    Puntos = Val(cPuntos)
    ReDim MPuntos(Puntos, 2)
    For j = 1 To 2
      For i = 1 To Puntos
        Line Input #1, MPuntos(i, j)
      Next i
    Next j

    Condicion = Val(cCondicion)
    ReDim MCondicion(Condicion, 2)
    For i = 1 To Condicion
      For j = 1 To 2
        Line Input #1, MCondicion(i, j)
      Next j
    Next i

    nMateriales = Val(cMaterial)
    ReDim MMaterial(nMateriales, 4)
    For i = 1 To nMateriales
      For j = 1 To 4
        Line Input #1, MMaterial(i, j)
      Next j
    Next i
  End While
End Sub
```

```

Next i
nElementos = Val(cElementos)
ReDim MEElementos(nElementos, 9)
For j = 1 To 9
For i = 1 To nElementos
Line Input #1, MEElementos(i, j)
Next i
Next j

Loop

Close 1

LeerPuntos (Puntos)
DiseñoPuntos (Puntos)
LeerElementos (nElementos)
DiseñoElementos (nElementos)
LeerCondiciones (Condicion)
DiseñoCondiciones (Condicion)
LeerMateriales (nMateriales)
DiseñoMateriales (nMateriales)

btnCondiciones.Enabled = True
mnCondiciones.Enabled = True
Dlogo.FileName = Archivo
Principal.Caption = "Análisis Dinámico de Estados Planos" + " [" +
Dlogo.FileName + "]"

cancela:
If Err = 62 Or Err = 30009 Then MsgBox "El archivo no corresponde al
de Preprocesador", 16, "Advertencia"
If Err = 30009 Then Nuevo
Close 1

Exit Sub

End Sub

Sub Acercade ()

Load acerca
acerca.Show

End Sub

Sub Borrar ()

actual = Dlogo.FileName
Dlogo.DialogTitle = "Borrar"
Dlogo.CancelError = True
Dlogo.FileName = " "
Dlogo.Filter = "Archivos de datos (*.dat)|*.dat|Todos los archivos
(*.*)|*.*"
Dlogo.FilterIndex = 1
Dlogo.Flags = OFN_FILEMUSTEXIST Or OFN_PATHMUSTEXIST Or
OFN_HIDEREADONLY
On Error GoTo cancelar
Dlogo.Action = 1

```

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

```

On Error GoTo 0
msg = "No se puede borrar el archivo " + Chr(10)
msg = msg + Dlogo.FileName + Chr(10)
msg = msg + "Acceso Denegado"
If Dlogo.FileName = actual Then MsgBox msg, 16, "Advertencia": Exit
Sub
    MsgBox "Borrando el archivo " + Dlogo.FileName, 0, "Borrar"
    Kill Dlogo.FileName
cancelar:
    Exit Sub
End Sub
Sub btnAbrir_Click ()
Abrir
End Sub
Sub btnAbrir_MouseMove (Button As Integer, Shift As Integer, X As Single,
Y As Single)
lblnuevo.Visible = False
lblAbrir.Visible = True
lblGuardar.Visible = False
lblBorrar.Visible = False
lblImprimir.Visible = False
lblSalir.Visible = False
lblAcerca.Visible = False
lblPuntos.Visible = False
lblCondiciones.Visible = False
lblElementos.Visible = False
lblMateriales.Visible = False
Panel3d2.Caption = "Abrir un archivo existente en disco"
End Sub
Sub btnAcerca_Click ()
Acercade
End Sub
Sub btnAcerca_MouseMove (Button As Integer, Shift As Integer, X As
Single, Y As Single)
lblnuevo.Visible = False
lblAbrir.Visible = False
lblGuardar.Visible = False
lblBorrar.Visible = False
lblImprimir.Visible = False
lblSalir.Visible = False
lblAcerca.Visible = True
Panel3d2.Caption = "Muestra a los autores del programa"
End Sub
Sub btnBorrar_Click ()

```

```
Borrar  
End Sub
```

```
Sub btnBorrar_MouseMove (Button As Integer, Shift As Integer, X As  
Single, Y As Single)
```

```
lblnuevo.Visible = False  
lblAbrir.Visible = False  
lblGuardar.Visible = False  
lblBorrar.Visible = True  
lblImprimir.Visible = False  
lblSalir.Visible = False  
lblAcerca.Visible = False  
lblPuntos.Visible = False  
lblCondiciones.Visible = False  
lblElementos.Visible = False  
lblMateriales.Visible = False
```

```
Panel3d2.Caption = "Borra un archivo existente en disco"
```

```
End Sub
```

```
Sub btnCondiciones_Click ()  
CargaCondiciones  
End Sub
```

```
Sub btnCondiciones_MouseMove (Button As Integer, Shift As Integer, X As  
Single, Y As Single)
```

```
lblnuevo.Visible = False  
lblAbrir.Visible = False  
lblGuardar.Visible = False  
lblBorrar.Visible = False  
lblImprimir.Visible = False  
lblSalir.Visible = False  
lblAcerca.Visible = False  
lblPuntos.Visible = False  
lblCondiciones.Visible = True  
lblElementos.Visible = False  
lblMateriales.Visible = False
```

```
Panel3d2.Caption = "Muestra tabla para captura de Condiciones Frontera"  
End Sub
```

```
Sub btnElementos_Click ()  
CargaElementos  
End Sub
```

```
Sub btnElementos_MouseMove (Button As Integer, Shift As Integer, X As  
Single, Y As Single)
```

```
lblnuevo.Visible = False  
lblAbrir.Visible = False  
lblGuardar.Visible = False  
lblBorrar.Visible = False  
lblImprimir.Visible = False  
lblSalir.Visible = False  
lblAcerca.Visible = False  
lblPuntos.Visible = False
```

```
lblCondiciones.Visible = False
lblElementos.Visible = True
lblMateriales.Visible = False
```

```
Panel3d2.Caption = "Muestra tabla para captura de Elementos"
End Sub
```

```
Sub btnGuardar_Click ()
Guardar
End Sub
```

```
Sub btnGuardar_MouseMove (Button As Integer, Shift As Integer, X As
Single, Y As Single)
```

```
lblnuevo.Visible = False
lblAbrir.Visible = False
lblGuardar.Visible = True
lblBorrar.Visible = False
lblImprimir.Visible = False
lblSalir.Visible = False
lblAcerca.Visible = False
lblPuntos.Visible = False
lblCondiciones.Visible = False
lblElementos.Visible = False
lblMateriales.Visible = False
```

```
Panel3d2.Caption = "Guarda los datos en un archivo .dat"
```

```
End Sub
```

```
Sub btnImprimir_Click ()
Imprimir
End Sub
```

```
Sub btnImprimir_MouseMove (Button As Integer, Shift As Integer, X As
Single, Y As Single)
```

```
lblnuevo.Visible = False
lblAbrir.Visible = False
lblGuardar.Visible = False
lblBorrar.Visible = False
lblImprimir.Visible = True
lblSalir.Visible = False
lblAcerca.Visible = False
lblPuntos.Visible = False
lblCondiciones.Visible = False
lblElementos.Visible = False
lblMateriales.Visible = False
```

```
Panel3d2.Caption = "Envía los datos capturados a la impresora"
```

```
End Sub
```

```
Sub btnMateriales_Click ()
CargaMateriales
End Sub
```

```
Sub btnMateriales_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
lblnuevo.Visible = False  
lblAbrir.Visible = False  
lblGuardar.Visible = False  
lblBorrar.Visible = False  
lblImprimir.Visible = False  
lblSalir.Visible = False  
lblAcerca.Visible = False  
lblPuntos.Visible = False  
lblCondiciones.Visible = False  
lblElementos.Visible = False  
lblMateriales.Visible = True
```

```
Panel3d2.Caption = "Muestra tabla para captura de Elementos"  
End Sub
```

```
Sub btnNuevo_Click ()  
Nuevo
```

```
End Sub
```

```
Sub btnNuevo_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
lblnuevo.Visible = True  
lblAbrir.Visible = False  
lblGuardar.Visible = False  
lblBorrar.Visible = False  
lblImprimir.Visible = False  
lblSalir.Visible = False  
lblAcerca.Visible = False  
lblPuntos.Visible = False  
lblCondiciones.Visible = False  
lblElementos.Visible = False  
lblMateriales.Visible = False
```

```
Panel3d2.Caption = "Crea un archivo nuevo para captura de datos"
```

```
End Sub
```

```
Sub btnPuntos_Click ()  
CargaPuntos  
End Sub
```

```
Sub btnPuntos_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
lblnuevo.Visible = False  
lblAbrir.Visible = False  
lblGuardar.Visible = False  
lblBorrar.Visible = False  
lblImprimir.Visible = False  
lblSalir.Visible = False  
lblAcerca.Visible = False  
lblPuntos.Visible = True  
lblCondiciones.Visible = False  
lblElementos.Visible = False  
lblMateriales.Visible = False
```

```

Panel3d2.Caption = "Muestra tabla para captura de Puntos Nodales"
End Sub

Sub btnSalir_Click ()
Salir
End Sub

Sub btnSalir_MouseMove (Button As Integer, Shift As Integer, X As Single,
Y As Single)
lblnuevo.Visible = False
lblAbrir.Visible = False
lblGuardar.Visible = False
lblBorrar.Visible = False
lblImprimir.Visible = False
lblSalir.Visible = True
lblAcerca.Visible = False
lblPuntos.Visible = False
lblCondiciones.Visible = False
lblElementos.Visible = False
lblMateriales.Visible = False

Panel3d2.Caption = "Salir del programa a Windows"

End Sub

Sub CargaCondiciones ()

Load Frontera
Frontera.Show
If Condicion = 0 Then
    Frontera.Text1.SetFocus
Else
    Frontera.TablaFrontera.SetFocus
End If

End Sub

Sub CargaElementos ()

Load Elementos
Elementos.Show
If nElementos = 0 Then
    Elementos.Text1.SetFocus
Else
    Elementos.TablaElementos.SetFocus
End If

End Sub

Sub CargaMateriales ()

Load Material
Material.Show
If nMateriales = 0 Then
    Material.Text1.SetFocus
Else
    Material.TablaMateriales.SetFocus
End If

```

```

End Sub

Sub CargaPuntos ()
Load PuntosNodales
PuntosNodales.Show
If Puntos = 0 Then
    PuntosNodales.Text1.SetFocus
Else
    PuntosNodales.TablaPuntos.SetFocus
End If
End Sub

Sub Form_Load ()

Dlogo.FileName = "Nuevo.dat"
Principal.Caption = "Análisis Dinámico de Estados Planos" + " (" +
Dlogo.FileName + ")"

Salvado = True
End Sub

Sub Form_MouseMove (Button As Integer, Shift As Integer, X As Single, Y
As Single)
Panel3d2.Caption = ""
lblnuevo.Visible = False
lblAbrir.Visible = False
lblGuardar.Visible = False
lblBorrar.Visible = False
lblImprimir.Visible = False
lblSalir.Visible = False
lblAcerca.Visible = False
lblPuntos.Visible = False
lblCondiciones.Visible = False
lblElementos.Visible = False
lblMateriales.Visible = False

End Sub

Sub Guardar ()

If Puntos = 0 Or nElementos = 0 Or Condicion = 0 Or nMateriales = 0
Then
    MsgBox "No se han completado los datos Iniciales", 16,
"Advertencia"
Exit Sub
End If
Dlogo.DialogTitle = "Guardar"
Dlogo.CancelError = True
Dlogo.DefaultExt = ".dat"
Dlogo.Filter = "Archivos de datos (*.dat)|*.dat|Todos los archivos
(*.*)|*.*"
Dlogo.FilterIndex = 1
Dlogo.Flags = OFN_OVERWRITEPROMPT Or OFN_PATHMUSTEXIST Or
OFN_HIDEREADONLY
On Error GoTo Cancelal
Dlogo.Action = 2
On Error GoTo 0

```

```

BackSlash = 0
For i = Len(Dlogo.FileName) To 1 Step -1
BackSlash = InStr(i, Dlogo.FileName, "\") ' Encuentra diagonal
invertida
If BackSlash <> 0 Then i = 1
Next i

Archivo = Right(Dlogo.FileName, Len(Dlogo.FileName) - BackSlash)
'Obtiene palabras a la derecha

Dlogo.FileName = Archivo

MsgBox "Escribiendo el archivo " + Archivo, 0, "Guardar"

Open Dlogo.FileName For Output Lock Write As 1

Print #1, Archivo
Print #1, Puntos
Print #1, Condicion
Print #1, nMateriales
Print #1, nElementos

For j = 1 To 2
For i = 1 To Puntos
PuntosNodales.TablaPuntos.Row = i
PuntosNodales.TablaPuntos.Col = j
Print #1, PuntosNodales.TablaPuntos.Text
Next i
Next j

For i = 1 To Condicion
For j = 1 To 2
Frontera.TablaFrontera.Row = i
Frontera.TablaFrontera.Col = j
Frontera.TablaFrontera.Text =
Format$(Val(Frontera.TablaFrontera.Text), "0.00")
Print #1, Frontera.TablaFrontera.Text
Frontera.TablaFrontera.Text =
Format$(Val(Frontera.TablaFrontera.Text), "0")
Next j
Next i

For i = 1 To nMateriales
For j = 1 To 4
Material.TablaMateriales.Row = i
Material.TablaMateriales.Col = j
Print #1, Material.TablaMateriales.Text
Next j
Next i

For j = 1 To 9
For i = 1 To nElementos
Elementos.TablaElementos.Row = i
Elementos.TablaElementos.Col = j
Elementos.TablaElementos.Text =
Format$(Val(Elementos.TablaElementos.Text), "0.00")
Print #1, Elementos.TablaElementos.Text
Elementos.TablaElementos.Text =
Format$(Val(Elementos.TablaElementos.Text), "0")

```

```

Next i
Next j

Close 1
Salvado = True
Principal.Caption = "Análisis Dinámico de Estados Planos" + " [" +
Dlogo.FileName + "]"

Cancelal:
Exit Sub

End Sub

Sub Imparchivo ()

Dim Columna As Integer
CrLf = Chr(10) + Chr(13)

DefExt = ".PRN"
PosExt = InStr(1, Dlogo.FileName, ".")
Dlogo.FileName = Mid(Dlogo.FileName, 1, PosExt) + DefExt
MsgBox "Creando el archivo " + Dlogo.FileName, 0, "Imprimir"

Open Dlogo.FileName For Output Lock Write As 2

Print #2, Spc(22); "Análisis Dinámico de Estados Planos"
Print #2,
Print #2, Dlogo.FileName
Print #2,
Print #2, "Número de Puntos Nodales: "; Puntos
Print #2,
Print #2, " No. "; " X "; " Y "
For i = 1 To Puntos
Columna = 2
For j = 0 To 2
PuntosNodales.TablaPuntos.Row = i
PuntosNodales.TablaPuntos.Col = j
Print #2, Tab(Columna); PuntosNodales.TablaPuntos.Text;
Columna = Columna + 7
Next j
Print #2,
Next i
Print #2, CrLf

Print #2, "Número de Elementos: "; nElementos
Print #2,
For i = 0 To nElementos
Columna = 2
For j = 0 To 9
Elementos.TablaElementos.Row = i
Elementos.TablaElementos.Col = j
Print #2, Tab(Columna); Elementos.TablaElementos.Text;
Columna = Columna + 7
Next j
Print #2,
Next i
Print #2, CrLf

Print #2, "Número de Condiciones Frontera"; Condicion
Print #2,

```

```

Print #2, " No. "; " DX "; " DY "
For i = 1 To Condicion
Columna = 2
For j = 0 To 2
Frontera.TablaFrontera.Row = i
Frontera.TablaFrontera.Col = j
Print #2, Tab(Columna); Frontera.TablaFrontera.Text;
Columna = Columna + 7
Next j
Print #2,
Next i
Print #2, crlf

Print #2, "Número de Materiales: "; nMateriales
Print #2,
Print #2, " No. "; " E "; " V "; "ESPESOR"; " P.V. "
For i = 1 To nMateriales
Columna = 2
For j = 0 To 4
Material.TablaMateriales.Row = i
Material.TablaMateriales.Col = j
Print #2, Tab(Columna); Material.TablaMateriales.Text;
Columna = Columna + 7
Next j
Print #2,
Next i
Print #2, crlf
Close 2

End Sub

Sub Impresora ()
Dim Columna As Integer
crlf = Chr(10) + Chr(13)

Printer.Print Spc(22); "Análisis Dinámico de Estados Planos"
Printer.Print
Printer.Print Dlogo.FileName
Printer.Print

Printer.Print "Número de Puntos Nodales: "; Puntos
Printer.Print
Printer.Print " No. "; " X "; " Y "
For i = 1 To Puntos
Columna = 2
For j = 0 To 2
PuntosNodales.TablaPuntos.Row = i
PuntosNodales.TablaPuntos.Col = j
Printer.Print Tab(Columna); PuntosNodales.TablaPuntos.Text;
Columna = Columna + 7
Next j
Printer.Print
Next i
Printer.Print crlf

Printer.Print "Número de Elementos: "; nElementos
Printer.Print
For i = 0 To nElementos
Columna = 2
For j = 0 To 9

```

```

Elementos.TablaElementos.Row = i
Elementos.TablaElementos.Col = j
Printer.Print Tab(Columna); Elementos.TablaElementos.Text;
Columna = Columna + 7
Next j
Printer.Print
Next i
Printer.Print crlf

Printer.Print "Número de Condiciones Frontera"; Condicion
Printer.Print
Printer.Print " No. "; " DX "; " DY "
For i = 1 To Condicion
Columna = 2
For j = 0 To 2
Frontera.TablaFrontera.Row = i
Frontera.TablaFrontera.Col = j
Printer.Print Tab(Columna); Frontera.TablaFrontera.Text;
Columna = Columna + 7
Next j
Printer.Print
Next i
Printer.Print crlf

Printer.Print "Número de Materiales: "; nMateriales
Printer.Print
Printer.Print " No. "; " E "; " V "; "ESPESOR"; " P.V. "
For i = 1 To nMateriales
Columna = 2
For j = 0 To 4
Material.TablaMateriales.Row = i
Material.TablaMateriales.Col = j
Printer.Print Tab(Columna); Material.TablaMateriales.Text;
Columna = Columna + 7
Next j
Printer.Print
Next i
Printer.EndDoc

End Sub

Sub Imprimir ()

If Dlogo.FileName = "" Then MsgBox "Guardar el archivo antes de
imprimir", 48, "Advertencia": Exit Sub

Dlogo.CancelError = True 'Abortar si se cancela el cuadro de dialogo
Dlogo.Copies = 1 'Ajustes por omision
Dlogo.Flags = PD_NOSELECTION Or PD_NOPAGENUMS 'Or
PD_USEDEVMODECOPIES 'Permitir mas de una copia
On Error GoTo cancela2
Dlogo.Action = 5
On Error GoTo 0
imprimirarchivo = Dlogo.Flags And PD_PRINTTOFILE

If imprimirarchivo Then
Imparchivo
Else
Impresora
End If

```

```

cancela2:
  Exit Sub

End Sub

Sub LeerCondiciones (Condicion As Integer)

  Frontera.TablaFrontera.Rows = Condicion + 1
  Frontera.TablaFrontera.Cols = 3

  For i = 1 To Condicion
    For j = 1 To 2
      Frontera.TablaFrontera.Row = i
      Frontera.TablaFrontera.Col = j
      Frontera.TablaFrontera.Text = MCondicion(i, j)
      Frontera.TablaFrontera.Text =
Format$(Val(Frontera.TablaFrontera.Text), "0")
    Next j
  Next i

End Sub

Sub LeerElementos (nElementos As Integer)

  Elementos.TablaElementos.Rows = nElementos + 1
  Elementos.TablaElementos.Cols = 10

  For i = 1 To nElementos
    For j = 1 To 9
      Elementos.TablaElementos.Row = i
      Elementos.TablaElementos.Col = j
      Elementos.TablaElementos.Text = MElementos(i, j)
      Elementos.TablaElementos.Text =
Format$(Val(Elementos.TablaElementos.Text), "0")
    Next j
  Next i

End Sub

Sub LeerMateriales (nMateriales As Integer)

  Material.TablaMateriales.Rows = nMateriales + 1
  Material.TablaMateriales.Cols = 5

  For i = 1 To nMateriales
    For j = 1 To 4
      Material.TablaMateriales.Row = i
      Material.TablaMateriales.Col = j
      Material.TablaMateriales.Text = MMaterial(i, j)
    Next j
  Next i

End Sub

Sub LeerPuntos (Puntos As Integer)

  PuntosNodales.TablaPuntos.Rows = Puntos + 1
  PuntosNodales.TablaPuntos.Cols = 3

```

```

For i = 1 To Puntos
  For j = 1 To 2
    PuntosNodales.TablaPuntos.Row = i
    PuntosNodales.TablaPuntos.Col = j
    PuntosNodales.TablaPuntos.Text = MPuntos(i, j)
  Next j
Next i

End Sub

Sub mnAbrir_Click ()
Abrir
End Sub

Sub mnAcerca_Click ()
Acercade
End Sub

Sub mnBorrar_Click ()
Borrar
End Sub

Sub mnCondiciones_Click ()
CargaCondiciones
End Sub

Sub mnElementos_Click ()
CargaElementos
End Sub

Sub mnGuardar_click ()
Guardar
End Sub

Sub mnImprimir_Click ()
Imprimir
End Sub

Sub mnMateriales_Click ()
CargaMateriales
End Sub

Sub mnNuevo_Click ()
Nuevo
'Load datos
'datos.Show

End Sub

Sub mnPuntos_Click ()
CargaPuntos
End Sub

```

```

Sub mnSalir_Click ()
Salir
End Sub

Sub Nuevo ()
If Salvado = False Then
    respuesta = MsgBox("Los cambios realizados se perderan." + Chr(13) +
        "¿Desea continuar?", MB_YESNO + MB_ICONSTOP + MB_DEFBOTTOM1,
        "Advertencia")
    If respuesta = IDNO Then Exit Sub
End If

PuntosNodales.TablaPuntos.Rows = 2
PuntosNodales.TablaPuntos.Cols = 2
PuntosNodales.TablaPuntos.Text = ""
Puntos = 0
PuntosNodales.Text1.Text = ""
PuntosNodales.TablaPuntos.Visible = False

Elementos.TablaElementos.Rows = 2
Elementos.TablaElementos.Cols = 2
Elementos.TablaElementos.Text = ""
nElementos = 0
Elementos.Text1.Text = ""
Elementos.TablaElementos.Visible = False

Frontera.TablaFrontera.Rows = 2
Frontera.TablaFrontera.Cols = 2
Frontera.TablaFrontera.Text = ""
Condicion = 0
Frontera.Text1.Text = ""
Frontera.TablaFrontera.Visible = False
btnCondiciones.Enabled = False
mnCondiciones.Enabled = False

Material.TablaMateriales.Rows = 2
Material.TablaMateriales.Cols = 2
Material.TablaMateriales.Text = ""
nMateriales = 0
Material.Text1.Text = ""
Material.TablaMateriales.Visible = False

Dlogo.FileName = "Nuevo.dat"
Principal.Caption = "Análisis Dinámico de Estados Planos" + " [" +
Dlogo.FileName + "]"
Salvado = True

End Sub

Sub Panel3D1_MouseMove (Button As Integer, Shift As Integer, X As Single,
Y As Single)
Panel3d2.Caption = ""
lblnuevo.Visible = False
lblAbrir.Visible = False
lblGuardar.Visible = False
lblBorrar.Visible = False

```

```
lblImprimir.Visible = False
lblSalir.Visible = False
lblAcerca.Visible = False
lblPuntos.Visible = False
lblMateriales.Visible = False
lblPuntos.Visible = False
lblCondiciones.Visible = False
lblElementos.Visible = False
lblMateriales.Visible = False
```

```
End Sub
```

```
Sub Salir ()
```

```
If Salvado = False Then
    respuesta = MsgBox("Los cambios realizados se perderán." + Chr(13) +
        "¿Desea guardar el archivo?", MB_YESNOCANCEL + MB_ICONSTOP +
        MB_DEFBUTTON1, "Advertencia")
    Select Case respuesta
        Case IDYES
            Guardar
        Case IDNO
            End
        Case IDCANCEL
            Exit Sub
    End Select
End If
```

```
If Salvado = False Then
    Exit Sub
Else
    End
End If
End Sub
```

ARCHIVO PUNTOS.FRM

```
Sub Cancelar_Click ()

If Puntos = 0 Then Hide : Exit Sub
Respuesta = MsgBox("La información contenida se perderá," + Chr(13) +
"¿Desea continuar?", MB_YESNO + MB_ICONSTOP + MB_DEFBUTTON1,
"Advertencia")
    If Respuesta = idyes Then
        TablaPuntos.Rows = 2
        TablaPuntos.Cols = 2
        TablaPuntos.Text = ""
        Puntos = 0
        text1.Text = ""
        TablaPuntos.Visible = False
        Hide
    Else
        Exit Sub
    End If

End Sub

Sub Command1_Click ()
Hide
End Sub

Sub TablaPuntos_KeyPress (Ka As Integer)

If Ka = 13 Then
    TablaPuntos.Text = Format$(Val(TablaPuntos.Text), "0.00")
    If TablaPuntos.Col < 2 Then
        SendKeys ("{right}")
    Else
        TablaPuntos.Col = 1
        SendKeys ("{down}")
    End If
    Exit Sub
End If

If cambio = True And Ka <> 8 Then TablaPuntos.Text = ""

If Not ((Ka >= 48 And Ka <= 57) Or Ka = 46 Or Ka = 8) Then Exit Sub

If Ka = 8 Then
    If Len(TablaPuntos.Text) <> 0 Then TablaPuntos.Text =
Left$(TablaPuntos.Text, Len(TablaPuntos.Text) - 1)
Else
    dato = Chr(Ka)
    TablaPuntos.Text = TablaPuntos.Text + dato
End If
cambio = False
salvado = False
End Sub

Sub TablaPuntos_RowColChange ()
cambio = True
End Sub

Sub Text1_KeyPress (Ka As Integer)
```

```

Puntos = 0
If Not ((Ka >= 48 And Ka <= 57) Or Ka = 13 Or Ka = 8) Then Ka = 0
If Ka = 13 Then
    Puntos = Val(text1.Text)
    If Puntos < 10 Then
        MsgBox "El valor minimo de Nodos es 10", 16, "Advertencia"
    Exit Sub
End If
TablaPuntos.Cols = 3
TablaPuntos.Rows = Puntos + 1
DiseñoPuntos (Puntos)
TablaPuntos.SetFocus

Condicion = Puntos
Frontera.TablaFrontera.Cols = 3
Frontera.TablaFrontera.Rows = Condicion + 1
DiseñoCondiciones (Condicion)

For i = 1 To Condicion
    For j = 1 To 2
        Frontera.TablaFrontera.Row = i
        Frontera.TablaFrontera.Col = j
        Frontera.TablaFrontera.Text = 0
    Next j
Next i

Principal.btnCondiciones.Enabled = True
Principal.mnCondiciones.Enabled = True
salvado = False

End If

End Sub

```

ARCHIVO ELEMENTO.FRM

```
Sub Command1_Click ()
Hide
End Sub

Sub Command2_Click ()

If nelementos = 0 Then Hide : Exit Sub
Respuesta = MsgBox("La información contenida se perderá," + Chr(13) +
";Desea continuar?", MB_YESNO + MB_ICONSTOP + MB_DEFBOTTOM1,
"Advertencia")
  If Respuesta = idyes Then
    TablaElementos.Rows = 2
    TablaElementos.Cols = 2
    TablaElementos.Text = ""
    nelementos = 0
    text1.Text = ""
    TablaElementos.Visible = False
  Hide
  Else
    Exit Sub
  End If
End Sub

Sub TablaElementos_KeyPress (Ka As Integer)

If Ka = 13 Then
  If TablaElementos.Col < 9 Then
    SendKeys ("(right)")
  Else
    TablaElementos.Col = 1
    SendKeys ("(down)")
  End If
  Exit Sub
End If

If cambio = True And Ka <> 8 Then TablaElementos.Text = ""
"

If Not ((Ka >= 48 And Ka <= 57) Or Ka = 8) Then Exit Sub

If Ka = 8 Then
  If Len(TablaElementos.Text) <> 0 Then TablaElementos.Text =
Left$(TablaElementos.Text, Len(TablaElementos.Text) - 1)
  Else
    dato = Chr(Ka)
    TablaElementos.Text = TablaElementos.Text + dato
  End If

  cambio = False
  salvado = False
End Sub

Sub TablaElementos_RowColChange ()
  cambio = True
End Sub

Sub Text1_KeyPress (Ka As Integer)
```

```
nelementos = 0
If Not ((Ka >= 48 And Ka <= 57) Or Ka = 13 Or Ka = 8) Then Ka = 0
If Ka = 13 Then
    nelementos = Val(text1.Text)
    If nelementos < 1 Then
        MsgBox "El valor mínimo de Elementos es 1", 16, "Advertencia"
        Exit Sub
    End If
    TablaElementos.Cols = 10
    TablaElementos.Rows = nelementos + 1
    DiseñoElementos (nelementos)
    TablaElementos.SetFocus
End If
salvado = False
End Sub
```

ARCHIVO FRONTERA.FRM

```
Sub Command1_Click ()
Hide
End Sub

Sub Command2_Click ()

If Condicion = 0 Then Hide : Exit Sub
Respuesta = MsgBox("La información contenida se perderá," + Chr(13) +
"¿Desea continuar?", MB_YESNO + MB_ICONSTOP + MB_DEFBUTTON1,
"Advertencia")
    If Respuesta = idyes Then
        For i = 1 To Condicion
            For j = 1 To 2
                TablaFrontera.Row = i
                TablaFrontera.Col = j
                TablaFrontera.Text = 0
            Next j
        Next i
    Else
        Exit Sub
    End If

End Sub

Sub TablaFrontera_KeyPress (Ka As Integer)

If Ka = 13 Then
    If TablaFrontera.Col < 2 Then
        SendKeys ("{right}")
    Else
        TablaFrontera.Col = 1
        SendKeys ("{down}")
    End If
    Exit Sub
End If

If cambio = True And Ka <> 8 Then TablaFrontera.Text = ""

If Not ((Ka >= 48 And Ka <= 57) Or Ka = 46 Or Ka = 8) Then Exit Sub

If Ka = 8 Then
    If Len(TablaFrontera.Text) <> 0 Then TablaFrontera.Text =
Left$(TablaFrontera.Text, Len(TablaFrontera.Text) - 1)
    Else
        dato = Chr(Ka)
        TablaFrontera.Text = TablaFrontera.Text + dato
    End If

    cambio = False
    salvado = False
End Sub

Sub TablaFrontera_RowColChange ()
cambio = True
End Sub
```

ARCHIVO MATERIAL.FRM

```
Sub Command1_Click ()
Hide
End Sub
```

```
Sub Command2_Click ()
```

```
If nMateriales = 0 Then Hide : Exit Sub
Respuesta = MsgBox("La información contenida se perderá," + Chr(13) +
"¿Desea continuar?", MB_YESNO + MB_ICONSTOP + MB_DEFBUTTON1,
"Advertencia")
```

```
    If Respuesta = idyes Then
        TablaMateriales.Rows = 2
        TablaMateriales.Cols = 2
        TablaMateriales.Text = ""
        nMateriales = 0
        text1.Text = ""
        TablaMateriales.Visible = False
        Hide
```

```
    Else
        Exit Sub
    End If
```

```
End Sub
```

```
Sub TablaMateriales_KeyPress (Ka As Integer)
```

```
If Ka = 13 Then
    TablaMateriales.Text = Format$(Val(TablaMateriales.Text), "0.00")
    If TablaMateriales.Col < 4 Then
        SendKeys "{right}")
    Else
        TablaMateriales.Col = 1
        SendKeys "{down}")
    End If
    Exit Sub
End If
```

```
If cambio = True And Ka <> 8 Then TablaMateriales.Text = ""
```

```
If Not ((Ka >= 48 And Ka <= 57) Or Ka = 46 Or Ka = 8) Then Exit Sub
```

```
If Ka = 8 Then
    If Len(TablaMateriales.Text) <> 0 Then TablaMateriales.Text =
Left$(TablaMateriales.Text, Len(TablaMateriales.Text) - 1)
    Else
        dato = Chr(Ka)
        TablaMateriales.Text = TablaMateriales.Text + dato
    End If
```

```
cambio = False
salvado = False
End Sub
```

```
Sub TablaMateriales_RowColChange ()
cambio = True
End Sub
```

```
Sub Text1_KeyPress (Ka As Integer)
```

```

nMateriales = 0
If Not ((Ka >= 48 And Ka <= 57) Or Ka = 13 Or Ka = 8) Then Ka = 0
If Ka = 13 Then
    nMateriales = Val(text1.Text)
    If nMateriales < 1 Then
        MsgBox "El valor minimo de materiales es 1", 16, "Advertencia"
    Exit Sub
    End If
    TablaMateriales.Cols = 5
    TablaMateriales.Rows = nMateriales + 1
    DiseñoMateriales (nMateriales)
    TablaMateriales.SetFocus
End If
salvado = False
End Sub

```

ARCHIVO ACERCA.FRM

```

Sub Command1_Click ()
Hide
End Sub

```

ARCHIVO DISEÑOS.BAS

```
Global Puntos As Integer
Global nElementos As Integer
Global Condicion As Integer
Global nMateriales As Integer
Global MPuntos() As String
Global MElementos() As String
Global MCondicion() As String
Global MMaterial() As String
Global cambio As Integer
Global Salvado As Integer
Global Dlogo() As String

' MsgBox parameters
Global Const MB_OK = 0
Global Const MB_OKCANCEL = 1
Global Const MB_ABORTRETRYIGNORE = 2
Global Const MB_YESNOCANCEL = 3
Global Const MB_YESNO = 4
Global Const MB_RETRYCANCEL = 5

Global Const MB_ICONSTOP = 16
Global Const MB_ICONQUESTION = 32
Global Const MB_ICONEXCLAMATION = 48
Global Const MB_ICONINFORMATION = 64

Global Const MB_APPLMODAL = 0
Global Const MB_DEFBUTTON1 = 0
Global Const MB_DEFBUTTON2 = 256
Global Const MB_DEFBUTTON3 = 512
Global Const MB_SYSTEMMODAL = 4096

' MsgBox return values
Global Const IDOK = 1
Global Const IDCANCEL = 2
Global Const IDABORT = 3
Global Const IDRETRY = 4
Global Const IDIGNORE = 5
Global Const IDYES = 6
Global Const IDNO = 7

' Common Dialog Control
' Action Property
Global Const DLG_FILE_OPEN = 1
Global Const DLG_FILE_SAVE = 2
Global Const DLG_COLOR = 3
Global Const DLG_FONT = 4
Global Const DLG_PRINT = 5
Global Const DLG_HELP = 6

' File Open/Save Dialog Flags
Global Const OFN_READONLY = &H1&
Global Const OFN_OVERWRITEPROMPT = &H2&
Global Const OFN_HIDEREADONLY = &H4&
Global Const OFN_NOCHANGEDIR = &H8&
Global Const OFN_SHOWHELP = &H10&
Global Const OFN_NOVALIDATE = &H100&
Global Const OFN_ALLOWMULTISELECT = &H200&
Global Const OFN_EXTENSIONDIFFERENT = &H400&
```

```
Global Const OFN_PATHMUSTEXIST = &H800&
Global Const OFN_FILEMUSTEXIST = &H1000&
Global Const OFN_CREATEPROMPT = &H2000&
Global Const OFN_SHAREAWARE = &H4000&
Global Const OFN_NOREADONLYRETURN = &H8000&
```

'Color Dialog Flags

```
Global Const CC_RGBINIT = &H1&
Global Const CC_FULLOPEN = &H2&
Global Const CC_PREVENTFULLOPEN = &H4&
Global Const CC_SHOWHELP = &H8&
```

'Fonts Dialog Flags

```
Global Const CF_SCREENFONTS = &H1&
Global Const CF_PRINTERFONTS = &H2&
Global Const CF_BOTH = &H3&
Global Const CF_SHOWHELP = &H4&
Global Const CF_INITTOLGOFONTSTRUCT = &H40&
Global Const CF_USESTYLE = &H80&
Global Const CF_EFFECTS = &H100&
Global Const CF_APPLY = &H200&
Global Const CF_ANSIONLY = &H400&
Global Const CF_NOVECTORFONTS = &H800&
Global Const CF_NOSIMULATIONS = &H1000&
Global Const CF_LIMITSIZE = &H2000&
Global Const CF_FIXEDPITCHONLY = &H4000&
Global Const CF_WYSIWYG = &H8000&
```

'must also have CF_SCREENFONTS

```
& CF_PRINTERFONTS
Global Const CF_FORCEFONTEXIST = &H10000
Global Const CF_SCALABLEONLY = &H20000
Global Const CF_TTONLY = &H40000
Global Const CF_NOFACESEL = &H80000
Global Const CF_NOSTYLESEL = &H100000
Global Const CF_NOSIZESEL = &H200000
```

'Printer Dialog Flags

```
Global Const PD_ALLPAGES = &H0&
Global Const PD_SELECTION = &H1&
Global Const PD_PAGENUMS = &H2&
Global Const PD_NOSELECTION = &H4&
Global Const PD_NOPAGENUMS = &H8&
Global Const PD_COLLATE = &H10&
Global Const PD_PRINTTOFILE = &H20&
Global Const PD_PRINTSETUP = &H40&
Global Const PD_NOWARNING = &H80&
Global Const PD_RETURNDC = &H100&
Global Const PD_RETURNIC = &H200&
Global Const PD_RETURNDEFAULT = &H400&
Global Const PD_SHOWHELP = &H800&
Global Const PD_USEDEVMODECOPIES = &H40000
Global Const PD_DISABLEPRINTTOFILE = &H80000
Global Const PD_HIDEPRINTTOFILE = &H100000
```

'Help Constants

```
Global Const HELP_CONTEXT = &H1
Global Const HELP_QUIT = &H2
Global Const HELP_INDEX = &H3
Global Const HELP_CONTENTS = &H3
Global Const HELP_HELPONHELP = &H4
```

'Display topic in ulTopic

'Terminate help

'Display index

'Display help on using help

```

Global Const HELP_SETINDEX = &H5           'Set the current Index for
multi index help
Global Const HELP_SETCONTENTS = &H5
Global Const HELP_CONTEXTPOPUP = &H8
Global Const HELP_FORCEFILE = &H9
Global Const HELP_KEY = &H101             'Display topic for keyword in
offabData
Global Const HELP_COMMAND = &H102
Global Const HELP_PARTIALKEY = &H105      'call the search engine in
winhelp

'Error Constants
Global Const CDERR_DIALOGFAILURE = -32768

Global Const CDERR_GENERALCODES = &H7FFF
Global Const CDERR_STRUCTSIZE = &H7FFE
Global Const CDERR_INITIALIZATION = &H7FFD
Global Const CDERR_NOTEMPLATE = &H7FFC
Global Const CDERR_NOINSTANCE = &H7FFB
Global Const CDERR_LOADSTRFAILURE = &H7FFA
Global Const CDERR_FINDRESFAILURE = &H7FF9
Global Const CDERR_LOADRESFAILURE = &H7FF8
Global Const CDERR_LOCKRESFAILURE = &H7FF7
Global Const CDERR_MEMALLOCFailure = &H7FF6
Global Const CDERR_MEMLOCKFAILURE = &H7FF5
Global Const CDERR_NOHOOK = &H7FF4

'Added for CMDIALOG.VBX
Global Const CDERR_CANCEL = &H7FF3
Global Const CDERR_NODLL = &H7FF2
Global Const CDERR_ERRPROC = &H7FF1
Global Const CDERR_ALLOC = &H7FF0
Global Const CDERR_HELP = &H7FEF

Global Const PDERR_PRINTERCODES = &H6FFF
Global Const PDERR_SETUPFAILURE = &H6FFE
Global Const PDERR_PARSEFAILURE = &H6FFD
Global Const PDERR_RETDEFFAILURE = &H6FFC
Global Const PDERR_LOADDRVFAILURE = &H6FFB
Global Const PDERR_GETDEVMODEFAIL = &H6FFA
Global Const PDERR_INITFAILURE = &H6FF9
Global Const PDERR_NODEVICES = &H6FF8
Global Const PDERR_NODEFAULTPRN = &H6FF7
Global Const PDERR_DNDMMISMATCH = &H6FF6
Global Const PDERR_CREATEICFAILURE = &H6FF5
Global Const PDERR_PRINTERNOTFOUND = &H6FF4

Global Const CFERR_CHOOSEFONTCODES = &H5FFF
Global Const CFERR_NOFONTS = &H5FFE

Global Const FNERR_FILENAMECODES = &H4FFF
Global Const FNERR_SUBCLASSFAILURE = &H4FFE
Global Const FNERR_INVALIDFILENAME = &H4FFD
Global Const FNERR_BUFFERTOOSMALL = &H4FFC

Global Const FRERR_FINDREPLACECODES = &H3FFF
Global Const CCERR_CHOOSECOLORCODES = &H2FFF

Sub DiseñoCondiciones (Condicion As Integer)
    Frontera.TablaFrontera.ColWidth(1) = 1000

```

```

Frontera.TablaFrontera.ColWidth(2) = 1000
Frontera.TablaFrontera.ColAlignment(1) = 1
Frontera.TablaFrontera.ColAlignment(2) = 1
Frontera.TablaFrontera.Col = 0
Frontera.TablaFrontera.Row = 0
Frontera.TablaFrontera.Text = "No."
Frontera.TablaFrontera.Col = 1
Frontera.TablaFrontera.Text = "    DX"
Frontera.TablaFrontera.Col = 2
Frontera.TablaFrontera.Text = "    DY"
Frontera.TablaFrontera.Col = 0
For i = 1 To Condicion
    Frontera.TablaFrontera.Row = i
    Frontera.TablaFrontera.Text = i
Next i

Frontera.TablaFrontera.Row = 1
Frontera.TablaFrontera.Col = 1
Frontera.TablaFrontera.Visible = True
Frontera.Text1.Text = Condicion

```

End Sub

```

Sub DiseñoElementos (nElementos As Integer)
    Elementos.TablaElementos.ColWidth(0) = 620
    Elementos.TablaElementos.ColWidth(1) = 870
    Elementos.TablaElementos.ColWidth(2) = 870
    Elementos.TablaElementos.ColWidth(3) = 870
    Elementos.TablaElementos.ColWidth(4) = 870
    Elementos.TablaElementos.ColWidth(5) = 870
    Elementos.TablaElementos.ColWidth(6) = 870
    Elementos.TablaElementos.ColWidth(7) = 870
    Elementos.TablaElementos.ColWidth(8) = 870
    Elementos.TablaElementos.ColWidth(9) = 950

    Elementos.TablaElementos.ColAlignment(1) = 1
    Elementos.TablaElementos.ColAlignment(2) = 1
    Elementos.TablaElementos.ColAlignment(3) = 1
    Elementos.TablaElementos.ColAlignment(4) = 1
    Elementos.TablaElementos.ColAlignment(5) = 1
    Elementos.TablaElementos.ColAlignment(6) = 1
    Elementos.TablaElementos.ColAlignment(7) = 1
    Elementos.TablaElementos.ColAlignment(8) = 1
    Elementos.TablaElementos.ColAlignment(9) = 1

    Elementos.TablaElementos.Col = 0
    Elementos.TablaElementos.Row = 0
    Elementos.TablaElementos.Text = "Elem."
    Elementos.TablaElementos.Col = 1
    Elementos.TablaElementos.Text = "Nodo I"
    Elementos.TablaElementos.Col = 2
    Elementos.TablaElementos.Text = "Nodo J"
    Elementos.TablaElementos.Col = 3
    Elementos.TablaElementos.Text = "Nodo K"
    Elementos.TablaElementos.Col = 4
    Elementos.TablaElementos.Text = "Nodo L"
    Elementos.TablaElementos.Col = 5
    Elementos.TablaElementos.Text = "Nodo M"
    Elementos.TablaElementos.Col = 6

```

```

Elementos.TablaElementos.Text = "Nodo N"
Elementos.TablaElementos.Col = 7
Elementos.TablaElementos.Text = "Nodo O"
Elementos.TablaElementos.Col = 8
Elementos.TablaElementos.Text = "Nodo P"
Elementos.TablaElementos.Col = 9
Elementos.TablaElementos.Text = "Material"

```

```

Elementos.TablaElementos.Col = 0
For i = 1 To nElementos
    Elementos.TablaElementos.Row = i
    Elementos.TablaElementos.Text = i
Next i
Elementos.TablaElementos.Row = 1
Elementos.TablaElementos.Col = 1
Elementos.TablaElementos.Visible = True
Elementos.Text1.Text = nElementos

```

End Sub

Sub DiseñoMateriales (nMateriales As Integer)

```

Material.TablaMateriales.ColWidth(1) = 1200
Material.TablaMateriales.ColWidth(2) = 1200
Material.TablaMateriales.ColWidth(3) = 1200
Material.TablaMateriales.ColWidth(4) = 1200
Material.TablaMateriales.ColAlignment(1) = 1
Material.TablaMateriales.ColAlignment(2) = 1
Material.TablaMateriales.ColAlignment(3) = 1
Material.TablaMateriales.ColAlignment(4) = 1
Material.TablaMateriales.Col = 0
Material.TablaMateriales.Row = 0
Material.TablaMateriales.Text = "No."
Material.TablaMateriales.Col = 1
Material.TablaMateriales.Text = "E"
Material.TablaMateriales.Col = 2
Material.TablaMateriales.Text = "V"
Material.TablaMateriales.Col = 3
Material.TablaMateriales.Text = "ESPESOR"
Material.TablaMateriales.Col = 4
Material.TablaMateriales.Text = "P. V."

```

```

Material.TablaMateriales.Col = 0
For i = 1 To nMateriales
    Material.TablaMateriales.Row = i
    Material.TablaMateriales.Text = i
Next i
Material.TablaMateriales.Row = 1
Material.TablaMateriales.Col = 1
Material.TablaMateriales.Visible = True
Material.Text1.Text = nMateriales

```

End Sub

Sub DiseñoPuntos (Puntos As Integer)

```

PuntosNodales.TablaPuntos.ColWidth(1) = 1000
PuntosNodales.TablaPuntos.ColWidth(2) = 1000
PuntosNodales.TablaPuntos.ColAlignment(1) = 1
PuntosNodales.TablaPuntos.ColAlignment(2) = 1

```

```
PuntosNodales.TablaPuntos.Col = 0
PuntosNodales.TablaPuntos.Row = 0
PuntosNodales.TablaPuntos.Text = "No."
PuntosNodales.TablaPuntos.Col = 1
PuntosNodales.TablaPuntos.Text = "      X"
PuntosNodales.TablaPuntos.Col = 2
PuntosNodales.TablaPuntos.Text = "      Y"
PuntosNodales.TablaPuntos.Col = 0
For i = 1 To Puntos
    PuntosNodales.TablaPuntos.Row = i
    PuntosNodales.TablaPuntos.Text = i
Next i
PuntosNodales.TablaPuntos.Row = 1
PuntosNodales.TablaPuntos.Col = 1
PuntosNodales.TablaPuntos.Visible = True
PuntosNodales.Text1.Text = Puntos
```

```
End Sub
```

APENDICE B

**LISTADO DEL
PROCESADOR 1**

```

C   PROGRAMA DEL METODO DE ELEMENTO FINITO DE 8 PUNTOS
C   ELABORADO POR:
C   ALICIA GARCIA MENDOZA
C   LUIS FERNANDO MAGAÑA JAIME
C   VERSION MARZO-97
C   (Calcula la matriz de rigideces y de masas)
C   PROGRAM PROCMEF8
C   DIMENSION A(18000)
C   COMMON INDI
C   NA=18000
C   CALL PRESENTA
C   MANEJO DE ARCHIVOS
C   ABRIR EL ARCHIVO DE DATOS, RESULTADOS, AUXILIAR
C   CALL ABRIRA
C   LECTURA DE DATOS INICIALES
C   CALL DATINIC(NPUN, NELEM, NMAT, 5)
C   DIMENSIONAMIENTO PARA MEMORIA DINAMICA
C   COORDENADAS 2-X, 3-Y PARA PUNTOS NODALES
C   N1=1
C   N2=N1+NPUN
C   INDICADOR DE ECUACION (CONDICIONES FRONTERA)
C   N3=N2+NPUN
C   MATERIALES
C   N4=N3+2*NPUN
C   NODOS (5-I, 6-J, 7-K, 8-L, 9-M, 10-N, 11-O, 12-P)
C   N5=N4+4*NMAT
C   N6=N5+NELEM
C   N7=N6+NELEM
C   N8=N7+NELEM
C   N9=N8+NELEM
C   N10=N9+NELEM
C   N11=N10+NELEM
C   N12=N11+NELEM
C   TIPO DE MATERIAL
C   N13=N12+NELEM
C   MATRIZ DE RIGIDEZ
C   N14=N13+NELEM
C   TAMAÑO MAXIMO DEL ARREGLO
C   NT=N14-1
C   REvisa EL DIMENSIONAMIENTO
C   CALL REvisa (NA, NT)
C   LECTURA DE DATOS
C   CALL LECAR(A, N1, N14, NT, 5)
C   CERRAR ARCHIVO
C   CALL FINAR(5)
C   CALCULO DEL INDICADOR DE ECUACION
C   CALL IND (A(N3), NPUN, 2, NC, NP, NN)
C   N15=N14+NC*NC
C   N16=N15+NC*NC
C   NT=N16-1
C   WRITE (*, 6100)NC
C   REvisa EL DIMENSIONAMIENTO
C   CALL REvisa (NA, NT)
C   CALCULO DE LA MATRIZ DE RIGIDECES DE LA ESTRUCTURA
C   CALL MATK8 (A(N1), A(N2), A(N3), A(N4), A(N5), A(N6), A(N7)
C   *, A(N8), A(N9), A(N10), A(N11), A(N12), A(N13), NPUN, NELEM, NMAT
C   *, NC, A(N14))
C   CALCULO DE LA MATRIZ DE MASA
C   CALL MMSA8 (A(N1), A(N2), A(N3), A(N4), A(N5), A(N6), A(N7)
C   *, A(N8), A(N9), A(N10), A(N11), A(N12), A(N13), NPUN, NELEM, NMAT

```

```

*,NC,A(N15))
C GUARDA LA MATRIZ DE RIGIDECES Y DE MASA DE LA ESTRUCTURA
CALL GUARMK(A(N14),A(N15),NC,7)
C CIERRA EL ARCHIVO DE IMPRESION DE RESULTADOS
CALL FINAR(6)
C CIERRA EL ARCHIVO QUE GUARDA LAMATRIZ DE RIGIDECES Y MASAS
CALL FINAR(7)
6100 FORMAT(/,2X,'NC = ',I10)
WRITE(*,*)
WRITE(*,*) ' P R O C E S O T E R M I N A D O . . . '
END

SUBROUTINE PRESENTA
WRITE(*,*) 'PROGRAMA DEL METODO DEL ELEMENTO FINITO DE 8 PUNTOS'
WRITE(*,*) 'ELABORADO POR:'
WRITE(*,*) 'ALICIA GARCIA MENDOZA'
WRITE(*,*) 'LUIS FERNANDO MAGAÑA JAIME'
WRITE(*,*) 'VERSION MARZO-97'
WRITE(*,*) '(Calcula la matriz de rigideces y de masas)'
WRITE(*,*)
RETURN
END

SUBROUTINE ABRIRA
COMMON INDI
CHARACTER NOMBRE*8,NOMBRE1*15,NOMBRE2*15,NOMBRE3*15,DRI*1
WRITE(*,*)
WRITE(*,*) ' DRIVE PARA LEER LOS DATOS (A,B,C)'
READ(*,5010)DRI
WRITE(*,*)
WRITE(*,*) ' ARCHIVO DE DATOS'
WRITE(*,*) ' DAR EL NOMBRE DEL ARCHIVO DE DATOS SIN EXTENSION'
WRITE(*,*) ' (MAXIMO 8 CARACTERES)'
READ(*,5000)NOMBRE
WRITE(*,*)
WRITE(*,*) ' INDICADOR DE EJECUCION (0=NORMAL, 1=COMPLETO)'
READ(*,5020)INDI
NOMBRE1=' '//DRI//':'//NOMBRE//'.DAT'
NOMBRE2=' '//DRI//':'//NOMBRE//'.RMR'
NOMBRE3=' '//DRI//':'//NOMBRE//'.MRI'
OPEN(5,FILE=NOMBRE1,STATUS='OLD')
OPEN(6,FILE=NOMBRE2,STATUS='NEW')
OPEN(7,FILE=NOMBRE3,STATUS='NEW')
5000 FORMAT(A8)
5010 FORMAT(A1)
5020 FORMAT(I1)
RETURN
END

SUBROUTINE DATINIC(NPUN,NELEM,NMAT,NAR)
CHARACTER*15 ARCHIVO
READ(NAR,5000)ARCHIVO
READ(NAR,*)NPUN
READ(NAR,*)NMAT
READ(NAR,*)NELEM
5000 FORMAT(A15)
RETURN
END

```

```

SUBROUTINE REVISA(NA,NT)
IF (NA.LT.NT)THEN
WRITE(*,6000)NA,NT
STOP
ELSE
WRITE(*,*)
WRITE(*,*) 'INFORMACION : '
WRITE(*,6000)NA,NT
WRITE(*,*)
WRITE(*,*) ' PRESIONE RETURN PARA PODER CONTINUAR ... '
READ(*,5000)NE
ENDIF
5000 FORMAT(I1)
6000 FORMAT(/,2X,'DIMENSIONAMIENTO DE A= ',I10,
* /,2X,'DIMENSIONAMIENTO USADO = ',I10)
RETURN
END

SUBROUTINE LECAR(A,NI,NF,NT,NAR)
DIMENSION A(NT)
DO 100 I=NI,NF-1
READ(NAR,*)A(I)
100 CONTINUE
RETURN
END

SUBROUTINE FINAR(NAR)
CLOSE(NAR)
RETURN
END

SUBROUTINE IND(XIND,NNOD,NGL,NC,NP,NN)
DIMENSION XIND(NGL,NNOD)
NC=0
NN=0
NP=0
DO 500 J=1, NNOD
DO 400 I=1, NGL
IN= INT(XIND(I,J))
IF (IN) 100,200,300
100 CONTINUE
NP=NP+1
XIND(I,J)=NP
GOTO 400
200 CONTINUE
NC=NC+1
XIND(I,J)=NC
GOTO 400
300 CONTINUE
NN=NN+1
XIND(I,J)=0
400 CONTINUE
500 CONTINUE
RETURN
END

SUBROUTINE MATKS (XX,YY,XIND,XMAT1,XIEL
* ,XJEL,XKEL,XLEL,XMEL,XNEL,XOEL,XPEL,XIMAT,NPUN,NELEM,NMAT,NC,RK)
C MATRIZ DE RIGIDECES GLOBAL
DIMENSION XX(NPUN),YY(NPUN),XIND(2,NPUN)

```

```

*,XMAT1(NMAT,4),
*XIEL(NELEM),XJEL(NELEM),XKEL(NELEM),XLEL(NELEM),XMEL(NELEM),
*XNEL(NELEM),XOEL(NELEM),XPEL(NELEM)
*,XIMAT(NELEM),RG(16,16),X(8),Y(8),IE(16),RK(NC,NC)
COMMON INDI
WRITE(*,*)'MATRIZ DE RIGIDECES'
DO L=1,NELEM
WRITE(*,*)'CALCULA ELEMENTO NO.',L
X(1)=XX(INT(XIEL(L)))
X(2)=XX(INT(XJEL(L)))
X(3)=XX(INT(XKEL(L)))
X(4)=XX(INT(XLEL(L)))
X(5)=XX(INT(XMEL(L)))
X(6)=XX(INT(XNEL(L)))
X(7)=XX(INT(XOEL(L)))
X(8)=XX(INT(XPEL(L)))
Y(1)=YY(INT(XIEL(L)))
Y(2)=YY(INT(XJEL(L)))
Y(3)=YY(INT(XKEL(L)))
Y(4)=YY(INT(XLEL(L)))
Y(5)=YY(INT(XMEL(L)))
Y(6)=YY(INT(XNEL(L)))
Y(7)=YY(INT(XOEL(L)))
Y(8)=YY(INT(XPEL(L)))
C
MATERIALES
E=XMAT1(INT(XIMAT(L)),1)
XNU=XMAT1(INT(XIMAT(L)),2)
T=XMAT1(INT(XIMAT(L)),3)
PV=XMAT1(INT(XIMAT(L)),4)
C
INDICADOR DE ECUACION
IE(1)=INT(XIND(1,INT(XIEL(L))))
IE(2)=INT(XIND(2,INT(XIEL(L))))
IE(3)=INT(XIND(1,INT(XJEL(L))))
IE(4)=INT(XIND(2,INT(XJEL(L))))
IE(5)=INT(XIND(1,INT(XKEL(L))))
IE(6)=INT(XIND(2,INT(XKEL(L))))
IE(7)=INT(XIND(1,INT(XLEL(L))))
IE(8)=INT(XIND(2,INT(XLEL(L))))
IE(9)=INT(XIND(1,INT(XMEL(L))))
IE(10)=INT(XIND(2,INT(XMEL(L))))
IE(11)=INT(XIND(1,INT(XNEL(L))))
IE(12)=INT(XIND(2,INT(XNEL(L))))
IE(13)=INT(XIND(1,INT(XOEL(L))))
IE(14)=INT(XIND(2,INT(XOEL(L))))
IE(15)=INT(XIND(1,INT(XPEL(L))))
IE(16)=INT(XIND(2,INT(XPEL(L))))
C
IMPRESION DE RESULTADOS (IE)
C
CALL IRESU1 (IE,B,D,DB,RG,RK,NC,L,6,1)
C
CALCULA LA MATRIZ DE RIGIDECES DE CADA ELEMENTO
CALL MATKES (E,XNU,X,Y,RG,T)
C
IMPRESION DE RESULTADOS (RG)
CALL IRESU1 (IE,B,D,DB,RG,RK,NC,L,6,3)
C
SE ENSAMBLA LA MATRIZ DE RIGIDECES DEL ELEMENTO
C
CON LA MATRIZ DE RIGIDECES DE LA ESTRUCTURA
CALL ENSAM (RK,RG,IE,NC,16)
END DO
C
IMPRESION DE RESULTADOS (RK)
CALL IRESU1 (IE,B,D,DB,RG,RK,NC,L,6,4)
RETURN
END

```

```

SUBROUTINE ENSAM(RK,RKE,IE,NC,NGLEL)
DIMENSION RK(NC,NC), RKE(NGLEL,NGLEL), IE(NGLEL)
DO 200 I=1, NGLEL
IF (IE(I).LE.0) GOTO 200
DO 100 J=1, NGLEL
IF (IE(J).LE.0) GOTO 100
IR=IE(I)
JR=IE(J)
RK(IR,JR)=RK(IR,JR)+RKE(I,J)
100 CONTINUE
200 CONTINUE
RETURN
END

C SUBROUTINE MATKES (E,XNU,X,Y,RG,T)
MATRIZ DE RIGIDECEDES DE CADA ELEMENTO
DIMENSION H(2),SI(4),ANU(4),D(3,3),B(3,16)
,BT(16,3),RIG(16,16),DB(3,16),X(8),Y(8),RG(16,16)
COMMON INDI
H(1)=1.0
H(2)=1.0
SI(1)=-0.57735026918963
SI(2)=-SI(1)
SI(3)=-SI(1)
SI(4)=SI(1)
ANU(1)=SI(1)
ANU(2)=SI(1)
ANU(3)=-SI(1)
ANU(4)=-SI(1)
IG=0
M=2
N=2

C INICIALIZANDO
DO I=1,16
DO J=1,16
RIG(I,J)=0.0
RG(I,J)=0.0
END DO
END DO
DO I=1,M
DO J=1,N
IG=IG+1
SA=SI(IG)
EA=ANU(IG)
CALL MATB8 (SA,EA,X,Y,DET,B,PGX,PGY)
CALL CD (D,E,XNU)
CALL MULMAT (D,B,DB,3,3,16)
CALL MATRAN (B,BT,3,16)
CALL MULMAT (BT,DB,RIG,16,3,3,16)
DO I1=1,16
DO J1=1,16
RG(I1,J1)=RG(I1,J1)+(T*DET*RIG(I1,J1))
END DO
END DO

C IMPRESION DE RESULTADOS
CALL TRESU1 (IE,B,D,DB,RG,RK,NC,L,6,2)
END DO
END DO
RETURN

```

```

END

SUBROUTINE IMAAR(A, NR, NC, NAR)
DIMENSION A(NR, NC)
DO 100 L=1,NC,5
M = L + 4
IF (M.GT.NC) M=NC
WRITE(NAR,6000)(K,K=L,M)
DO 100 I=1,NR
WRITE(NAR,6010)I,(A(I,J),J=L,M)
CONTINUE
100 RETURN
6000 FORMAT(/,9X,I4,4(10X,I4))
6010 FORMAT(1X,I4,5(1PE14.6))
RETURN
END

SUBROUTINE MULMAT(A,B,C,NRA,NCA,NRB,NCB)
DIMENSION A(NRA,NCA),B(NRB,NCB),C(NRA,NCB)
DO 300 I=1,NRA
DO 200 J=1,NCB
XX=0.0
DO 100 K=1,NCA
XX=XX+A(I,K)*B(K,J)
CONTINUE
100 C(I,J)=XX
200 CONTINUE
300 CONTINUE
RETURN
END

SUBROUTINE MATBS(SA,EA,X,Y,DET,B,PGX,PGY)
DIMENSION FN(8),DFNS(8),DFNN(8),X(8),Y(8),DFNX(8),
*DFNY(8),B(3,16)
C FUNCIONES DE FORMA
FN(1)=0.25*(1-SA)*(1-EA)*(-SA-EA-1)
FN(2)=0.50*(1-SA**2)*(1-EA)
FN(3)=0.25*(1+SA)*(1-EA)*(SA-EA-1)
FN(4)=0.50*(1+SA)*(1-EA**2)
FN(5)=0.25*(1+SA)*(1+EA)*(SA+EA-1)
FN(6)=0.50*(1-SA**2)*(1+EA)
FN(7)=0.25*(1-SA)*(1+EA)*(-SA+EA-1)
FN(8)=0.5*(1-SA)*(1-EA**2)
C DERIVADAS DE LA FUNCIONES DE FORMA EN REF. LOCAL
DFNS(1)=0.25*(1-EA)*(EA+2*SA)
DFNS(2)=SA*(EA-1)
DFNS(3)=0.25*(1-EA)*(2*SA-EA)
DFNS(4)=0.5*(1-EA**2)
DFNS(5)=0.25*(1+EA)*(2*SA+EA)
DFNS(6)=-SA*(EA+1)
DFNS(7)=0.25*(1+EA)*(2*SA-EA)
DFNS(8)=0.5*(EA**2-1)
DFNN(1)=0.25*(1-SA)*(SA+2*EA)
DFNN(2)=0.5*(SA**2-1)
DFNN(3)=0.25*(1+SA)*(2*EA-SA)
DFNN(4)=-EA*(1+SA)
DFNN(5)=0.25*(1+SA)*(2*EA+SA)
DFNN(6)=0.5*(1-SA**2)
DFNN(7)=0.25*(1-SA)*(2*EA-SA)
DFNN(8)=-EA*(1-SA)

```

```

C      PUNTOS GAUSSIANOS
      PGX=FN (1)*X (1)+FN (2)*X (2)+FN (3)*X (3)+FN (4)*X (4)
      +FN (5)*X (5)+FN (6)*X (6)+FN (7)*X (7)+FN (8)*X (8)
      PGY=FN (1)*Y (1)+FN (2)*Y (2)+FN (3)*Y (3)+FN (4)*Y (4)
      +FN (5)*Y (5)+FN (6)*Y (6)+FN (7)*Y (7)+FN (8)*Y (8)
C      DERIVADAS DE LAS CORD. GLOBALES EN TERMINOS DE LAS
C      LOCALES.
      DXS=DFNS (1)*X (1)+DFNS (2)*X (2)+DFNS (3)*X (3)+DFNS (4)*X (4)
      +DFNS (5)*X (5)+DFNS (6)*X (6)+DFNS (7)*X (7)+DFNS (8)*X (8)
      DXN=DFNN (1)*X (1)+DFNN (2)*X (2)+DFNN (3)*X (3)+DFNN (4)*X (4)
      +DFNN (5)*X (5)+DFNN (6)*X (6)+DFNN (7)*X (7)+DFNN (8)*X (8)
      DYS=DFNS (1)*Y (1)+DFNS (2)*Y (2)+DFNS (3)*Y (3)+DFNS (4)*Y (4)
      +DFNS (5)*Y (5)+DFNS (6)*Y (6)+DFNS (7)*Y (7)+DFNS (8)*Y (8)
      DYN=DFNN (1)*Y (1)+DFNN (2)*Y (2)+DFNN (3)*Y (3)+DFNN (4)*Y (4)
      +DFNN (5)*Y (5)+DFNN (6)*Y (6)+DFNN (7)*Y (7)+DFNN (8)*Y (8)
      DET=DXS*DYN-DYS*DXN
C      DERIVADAS DE LAS FUNCIONES CON RESPECTO A X y Y
      DFNX (1)=(DYN*DFNS (1)-DYS*DFNN (1))/DET
      DFNX (2)=(DYN*DFNS (2)-DYS*DFNN (2))/DET
      DFNX (3)=(DYN*DFNS (3)-DYS*DFNN (3))/DET
      DFNX (4)=(DYN*DFNS (4)-DYS*DFNN (4))/DET
      DFNX (5)=(DYN*DFNS (5)-DYS*DFNN (5))/DET
      DFNX (6)=(DYN*DFNS (6)-DYS*DFNN (6))/DET
      DFNX (7)=(DYN*DFNS (7)-DYS*DFNN (7))/DET
      DFNX (8)=(DYN*DFNS (8)-DYS*DFNN (8))/DET
      DFNy (1)=(-DXN*DFNS (1)+DXS*DFNN (1))/DET
      DFNy (2)=(-DXN*DFNS (2)+DXS*DFNN (2))/DET
      DFNy (3)=(-DXN*DFNS (3)+DXS*DFNN (3))/DET
      DFNy (4)=(-DXN*DFNS (4)+DXS*DFNN (4))/DET
      DFNy (5)=(-DXN*DFNS (5)+DXS*DFNN (5))/DET
      DFNy (6)=(-DXN*DFNS (6)+DXS*DFNN (6))/DET
      DFNy (7)=(-DXN*DFNS (7)+DXS*DFNN (7))/DET
      DFNy (8)=(-DXN*DFNS (8)+DXS*DFNN (8))/DET
C      CALCULO DE B
      B (1,1)=DFNX (1)
      B (1,3)=DFNX (2)
      B (1,5)=DFNX (3)
      B (1,7)=DFNX (4)
      B (1,9)=DFNX (5)
      B (1,11)=DFNX (6)
      B (1,13)=DFNX (7)
      B (1,15)=DFNX (8)
      B (2,2)=DFNy (1)
      B (2,4)=DFNy (2)
      B (2,6)=DFNy (3)
      B (2,8)=DFNy (4)
      B (2,10)=DFNy (5)
      B (2,12)=DFNy (6)
      B (2,14)=DFNy (7)
      B (2,16)=DFNy (8)
      B (3,1)=DFNy (1)
      B (3,2)=DFNX (1)
      B (3,3)=DFNy (2)
      B (3,4)=DFNX (2)
      B (3,5)=DFNy (3)
      B (3,6)=DFNX (3)
      B (3,7)=DFNy (4)
      B (3,8)=DFNX (4)
      B (3,9)=DFNy (5)
      B (3,10)=DFNX (5)

```

```

B(3,11)=DFNY(6)
B(3,12)=DFNX(6)
B(3,13)=DFNY(7)
B(3,14)=DFNX(7)
B(3,15)=DFNY(8)
B(3,16)=DFNX(8)
RETURN
END

```

```

SUBROUTINE CD(D,E,XNU)
DIMENSION D(3,3)
T=E/(1-XNU**2)
D(1,1)=T
D(1,2)=XNU*T
D(1,3)=0
D(2,1)=XNU*T
D(2,2)=T
D(2,3)=0
D(3,1)=0
D(3,2)=0
D(3,3)=(1-XNU)*T/2
RETURN
END

```

```

SUBROUTINE MATRAN (A,AT,NR,NC)
DIMENSION A(NR,NC),AT(NC,NR)
DO I=1,NR
DO J=1,NC
AT(J,I)=A(I,J)
END DO
END DO
RETURN
END

```

```

SUBROUTINE IRESU1(IE,B,D,DB,RG,RK,NC,L,NAR,INDICA)
DIMENSION IE(16),B(3,16),DB(3,16),D(3,3),RG(16,16),
*RK(NC,NC)
COMMON INDI
IF (INDI.NE.0) THEN
IF (INDICA.EQ.1) THEN
WRITE(NAR,6000)L
WRITE(NAR,*)
WRITE(NAR,*) 'INDICADOR DE ECUACION'
WRITE(NAR,6010) (IE(IY),IY=1,16)
ENDIF
IF (INDICA.EQ.2) THEN
WRITE(NAR,*)
WRITE(NAR,*) 'MATRIZ B'
CALL IMAAR(D,3,3,NAR)
WRITE(NAR,*)
WRITE(NAR,*) 'MATRIZ B'
CALL IMAAR(B,3,16,NAR)
WRITE(NAR,*)
WRITE(NAR,*) 'MATRIZ DB'
CALL IMAAR(DB,3,16,NAR)
ENDIF
IF (INDICA.EQ.3) THEN
WRITE(NAR,*)
WRITE(NAR,*) 'MATRIZ DE RIGIDEZES'

```

```

CALL IMAAR(RG,16,16,NAR)
ENDIF
ENDIF
IF (INDICA.EQ.4) THEN
WRITE(NAR,*)
WRITE(NAR,*) 'MATRIZ DE RIGIDECE DE LA ESTRUCTURA'
CALL IMAAR(RK,NC,NC,NAR)
ENDIF
6000 FORMAT(//,2X,'NUMERO DE ELEMENTO = ',I5)
6010 FORMAT(16I3)
RETURN
END

SUBROUTINE MMASAS(XX,YY,XIND,XMAT1,XIEL
*,XJEL,XKEL,XLEL,XMEL,XNEL,XOEL,XPEL,XIMAT,NPUN,NELEM,NMAT,NC
*,RMASA)
C MATRIZ DE MASAS GLOBAL
DIMENSION XX(NPUN),YY(NPUN),XIND(2,NPUN),XMAT1(NMAT,4),
*XIEL(NELEM),XJEL(NELEM),XKEL(NELEM),XLEL(NELEM),XMEL(NELEM),
*XNEL(NELEM),XOEL(NELEM),XPEL(NELEM),XIMAT(NELEM)
*,RM(16,16),X(8),Y(8),IE(16),RMASA(NC,NC)
COMMON INDI
WRITE(*,*) 'MATRIZ DE MASAS'
DO L=1,NELEM
WRITE(*,*) 'CALCULA ELEMENTO NO.',L
X(1)=XX(INT(XIEL(L)))
X(2)=XX(INT(XJEL(L)))
X(3)=XX(INT(XKEL(L)))
X(4)=XX(INT(XLEL(L)))
X(5)=XX(INT(XMEL(L)))
X(6)=XX(INT(XNEL(L)))
X(7)=XX(INT(XOEL(L)))
X(8)=XX(INT(XPEL(L)))
Y(1)=YY(INT(XIEL(L)))
Y(2)=YY(INT(XJEL(L)))
Y(3)=YY(INT(XKEL(L)))
Y(4)=YY(INT(XLEL(L)))
Y(5)=YY(INT(XMEL(L)))
Y(6)=YY(INT(XNEL(L)))
Y(7)=YY(INT(XOEL(L)))
Y(8)=YY(INT(XPEL(L)))
C MATERIALES
E=XMAT1(INT(XIMAT(L)),1)
XNU=XMAT1(INT(XIMAT(L)),2)
T=XMAT1(INT(XIMAT(L)),3)
PV=XMAT1(INT(XIMAT(L)),4)
C INDICADOR DE ECUACION
IE(1)=INT(XIND(1,INT(XIEL(L))))
IE(2)=INT(XIND(2,INT(XIEL(L))))
IE(3)=INT(XIND(1,INT(XJEL(L))))
IE(4)=INT(XIND(2,INT(XJEL(L))))
IE(5)=INT(XIND(1,INT(XKEL(L))))
IE(6)=INT(XIND(2,INT(XKEL(L))))
IE(7)=INT(XIND(1,INT(XLEL(L))))
IE(8)=INT(XIND(2,INT(XLEL(L))))
IE(9)=INT(XIND(1,INT(XMEL(L))))
IE(10)=INT(XIND(2,INT(XMEL(L))))
IE(11)=INT(XIND(1,INT(XNEL(L))))
IE(12)=INT(XIND(2,INT(XNEL(L))))
IE(13)=INT(XIND(1,INT(XOEL(L))))

```

```

IE(14)=INT(XIND(2,INT(XOEL(L))))
IE(15)=INT(XIND(1,INT(XPEL(L))))
IE(16)=INT(XIND(2,INT(XPEL(L))))
C IMPRESION DE RESULTADOS (IE)
CALL IRESU2 (IE, FN, RM, RMASA, NC, L, 6, 1)
C CALCULA LA MATRIZ DE MASAS DE CADA ELEMENTO
CALL MASAS (T, PV, X, Y, RM)
C IMPRESION DE RESULTADOS (RM)
CALL IRESU2 (IE, FN, RM, RMASA, NC, L, 6, 3)
C SE ENSAMBLA LA MATRIZ DE MASAS DEL ELEMENTO
C CON LA MATRIZ DE MASAS DE LA ESTRUCTURA
CALL ENSAM (RMASA, RM, IE, NC, 16)
END DO
C IMPRESION DE RESULTADOS (RMASA)
CALL IRESU2 (IE, FN, RM, RMASA, NC, L, 6, 4)
RETURN
END

SUBROUTINE MASAS (T, PV, X, Y, RM)
C MATRIZ DE MASAS DE CADA ELEMENTO
DIMENSION H(2), SI(4), ANU(4), RM(16, 16), X(8), Y(8)
*, RIM(16, 16), FN(8)
COMMON INDI
H(1)=1.0
H(2)=1.0
SI(1)=-0.57735026918963
SI(2)=-SI(1)
SI(3)=-SI(1)
SI(4)=SI(1)
ANU(1)=SI(1)
ANU(2)=SI(1)
ANU(3)=-SI(1)
ANU(4)=-SI(1)
IG=0
M=2
N=2
C INICIALIZANDO
DO I=1, 16
DO J=1, 16
RIM(I, J)=0.0
RM(I, J)=0.0
END DO
END DO
DO I=1, M
DO J=1, N
IG=IG+1
SA=SI(IG)
EA=ANU(IG)
CALL MASFN8 (SA, EA, X, Y, FN, DET)
CALL MMAS8 (RIM, FN)
DO I1=1, 16
DO J1=1, 16
RM(I1, J1)=RM(I1, J1) + (PV*T*DET*RIM(I1, J1))
END DO
END DO
END DO
C IMPRESION DE RESULTADOS
CALL IRESU2 (IE, FN, RM, RMASA, NC, L, 6, 2)
END DO
END DO
RETURN

```

END

SUBROUTINE MMAS8(RIM, FN)
DIMENSION RIM(16, 16), FN(16)

C

RIM(1, 1)=FN(1)*FN(1)
RIM(2, 2)=RIM(1, 1)
RIM(3, 3)=FN(2)*FN(2)
RIM(4, 4)=RIM(3, 3)
RIM(5, 5)=FN(3)*FN(3)
RIM(6, 6)=RIM(5, 5)
RIM(7, 7)=FN(4)*FN(4)
RIM(8, 8)=RIM(7, 7)
RIM(9, 9)=FN(5)*FN(5)
RIM(10, 10)=RIM(9, 9)
RIM(11, 11)=FN(6)*FN(6)
RIM(12, 12)=RIM(11, 11)
RIM(13, 13)=FN(7)*FN(7)
RIM(14, 14)=RIM(13, 13)
RIM(15, 15)=FN(8)*FN(8)
RIM(16, 16)=RIM(15, 15)

C

RIM(1, 3)=FN(1)*FN(2)
RIM(1, 5)=FN(1)*FN(3)
RIM(1, 7)=FN(1)*FN(4)
RIM(1, 9)=FN(1)*FN(5)
RIM(1, 11)=FN(1)*FN(6)
RIM(1, 13)=FN(1)*FN(7)
RIM(1, 15)=FN(1)*FN(8)

C

RIM(2, 4)=FN(1)*FN(2)
RIM(2, 6)=FN(1)*FN(3)
RIM(2, 8)=FN(1)*FN(4)
RIM(2, 10)=FN(1)*FN(5)
RIM(2, 12)=FN(1)*FN(6)
RIM(2, 14)=FN(1)*FN(7)
RIM(2, 16)=FN(1)*FN(8)

C

RIM(3, 3)=FN(2)*FN(3)
RIM(3, 5)=FN(2)*FN(4)
RIM(3, 7)=FN(2)*FN(5)
RIM(3, 9)=FN(2)*FN(6)
RIM(3, 11)=FN(2)*FN(7)
RIM(3, 13)=FN(2)*FN(8)

C

RIM(4, 6)=FN(2)*FN(3)
RIM(4, 8)=FN(2)*FN(4)
RIM(4, 10)=FN(2)*FN(5)
RIM(4, 12)=FN(2)*FN(6)
RIM(4, 14)=FN(2)*FN(7)
RIM(4, 16)=FN(2)*FN(8)

C

RIM(5, 7)=FN(3)*FN(4)
RIM(5, 9)=FN(3)*FN(5)
RIM(5, 11)=FN(3)*FN(6)
RIM(5, 13)=FN(3)*FN(7)
RIM(5, 15)=FN(3)*FN(8)

C

RIM(6, 8)=FN(3)*FN(4)
RIM(6, 10)=FN(3)*FN(5)

C RIM(6, 12)=FN(3)*FN(6)
RIM(6, 14)=FN(3)*FN(7)
RIM(6, 16)=FN(3)*FN(8)

C RIM(7, 9)=FN(4)*FN(5)
RIM(7, 11)=FN(4)*FN(6)
RIM(7, 13)=FN(4)*FN(7)
RIM(7, 15)=FN(4)*FN(8)

C RIM(8, 10)=FN(4)*FN(5)
RIM(8, 12)=FN(4)*FN(6)
RIM(8, 14)=FN(4)*FN(7)
RIM(8, 16)=FN(4)*FN(8)

C RIM(9, 11)=FN(5)*FN(6)
RIM(9, 13)=FN(5)*FN(7)
RIM(9, 15)=FN(5)*FN(8)

C RIM(10, 12)=FN(5)*FN(6)
RIM(10, 14)=FN(5)*FN(7)
RIM(10, 16)=FN(5)*FN(8)

C RIM(11, 13)=FN(6)*FN(7)
RIM(11, 15)=FN(6)*FN(8)

C RIM(12, 14)=FN(6)*FN(7)
RIM(12, 16)=FN(6)*FN(8)

C RIM(13, 15)=FN(7)*FN(8)

RIM(14, 16)=FN(7)*FN(8)

DO I=1,16

DO J=I,16

RIM(J, I)=RIM(I, J)

ENDDO

ENDDO

RETURN

END

C SUBROUTINE MASFNS (SA, EA, X, Y, FN, DET)
DIMENSION FN(8), DFNS(8), DFNN(8), X(8), Y(8)
FUNCIONES DE FORMA

C FN(1)=0.25*(1-SA)*(1-EA)*(-SA-EA-1)
FN(2)=0.50*(1-SA**2)*(1-EA)
FN(3)=0.25*(1+SA)*(1-EA)*(SA-EA-1)
FN(4)=0.50*(1+SA)*(1-EA**2)
FN(5)=0.25*(1+SA)*(1+EA)*(SA+EA-1)
FN(6)=0.50*(1-SA**2)*(1+EA)
FN(7)=0.25*(1-SA)*(1+EA)*(-SA+EA-1)
FN(8)=0.5*(1-SA)*(1-EA**2)

C DERIVADAS DE LA FUNCIONES DE FORMA EN REF. LOCAL
DFNS(1)=0.25*(1-EA)*(EA+2*SA)
DFNS(2)=SA*(EA-1)

DFNS(3)=0.25*(1-EA)*(2*SA-EA)

DFNS(4)=0.5*(1-EA**2)

DFNS(5)=0.25*(1+EA)*(2*SA+EA)

DFNS(6)=-SA*(EA+1)

DFNS(7)=0.25*(1+EA)*(2*SA-EA)

DFNS(8)=0.5*(EA**2-1)

DFNN(1)=0.25*(1-SA)*(SA+2*EA)

```

DFNN (2)=0.5*(SA**2-1)
DFNN (3)=0.25*(1+SA)*(2*EA-SA)
DFNN (4)=-EA*(1+SA)
DFNN (5)=0.25*(1+SA)*(2*EA+SA)
DFNN (6)=0.5*(1-SA**2)
DFNN (7)=0.25*(1-SA)*(2*EA-SA)
DFNN (8)=-EA*(1-SA)
C
C
DERIVADAS DE LAS CORD. GLOBALES EN TERMINOS DE LAS
LOCALES.
DKS=DFNS (1)*X (1)+DFNS (2)*X (2)+DFNS (3)*X (3)+DFNS (4)*X (4)
*+DFNS (5)*X (5)+DFNS (6)*X (6)+DFNS (7)*X (7)+DFNS (8)*X (8)
DXN=DFNN (1)*X (1)+DFNN (2)*X (2)+DFNN (3)*X (3)+DFNN (4)*X (4)
*+DFNN (5)*X (5)+DFNN (6)*X (6)+DFNN (7)*X (7)+DFNN (8)*X (8)
DYS=DFNS (1)*Y (1)+DFNS (2)*Y (2)+DFNS (3)*Y (3)+DFNS (4)*Y (4)
*+DFNS (5)*Y (5)+DFNS (6)*Y (6)+DFNS (7)*Y (7)+DFNS (8)*Y (8)
DYN=DFNN (1)*Y (1)+DFNN (2)*Y (2)+DFNN (3)*Y (3)+DFNN (4)*Y (4)
*+DFNN (5)*Y (5)+DFNN (6)*Y (6)+DFNN (7)*Y (7)+DFNN (8)*Y (8)
DET=DKS*DYN-DYS*DXN
RETURN
END

```

```

SUBROUTINE IRESU2 (IE, FN, RM, RMASA, NC, L, NAR, INDIR)
DIMENSION IE (16), RM (16, 16), RMASA (NC, NC), FN (8)
COMMON INDI
IF (INDI.NE.0) THEN
IF (INDIR.EQ.1) THEN
WRITE (NAR, 6000) L
WRITE (NAR, *)
WRITE (NAR, *) 'INDICADOR DE ECUACION'
WRITE (NAR, 6010) (IE (IY), IY=1, 16)
ENDIF
IF (INDIR.EQ.2) THEN
WRITE (NAR, *)
WRITE (NAR, *) 'VECTOR DE FUNCIONES'
CALL IMAAR (FN, 8, 1, NAR)
ENDIF
IF (INDIR.EQ.3) THEN
WRITE (NAR, *)
WRITE (NAR, *) 'MATRIZ DE MASAS'
CALL IMAAR (RM, 16, 16, NAR)
ENDIF
ENDIF
IF (INDIR.EQ.4) THEN
WRITE (NAR, *)
WRITE (NAR, *) 'MATRIZ DE MASAS DE LA ESTRUCTURA'
CALL IMAAR (RMASA, NC, NC, NAR)
ENDIF
6000 FORMAT (//, 2X, 'NUMERO DE ELEMENTO = ', I5)
6010 FORMAT (16I3)
RETURN
END

```

```

SUBROUTINE GUARMK (RK, RMASA, NC, NAR)
DIMENSION RK (NC, NC), RMASA (NC, NC)
WRITE (NAR, 6000) NC
DO I=1, NC
DO J=1, NC
WRITE (NAR, *) RK (I, J)
ENDDO
ENDDO

```

```
DO I=1,NC
DO J=1,NC
WRITE(NAR,*)RMASA(I,J)
ENDDO
6000  ENDDO
      FORMAT(I5)
      RETURN
      END
```

APENDICE C

LISTADO DEL PROCESADOR 3

```

PROGRAM BETA397
DIMENSION A(50000)
PROGRAMA DEL METODO GENERALIZADO BETA NEWMARK
ELABORADO POR:
ALICIA GARCIA MENDOZA
LUIS FERNANDO MAGAÑA JAIME
VERSION MARZO-97
NA=50000
CALL PRESENTA
LECTURA DE LOS ARCHIVOS
MATRIZ DE RIGIDECEZ Y DE MASAS
CALL ABRIRA (INDI)
READ(5,*) NC
N1 = 1
N2 = N1+(NC*NC)
N3 = N2+(NC*NC)
DO I = N1, N3-1
  READ(5,*) A(I)
END DO
CLOSE(5)
NT=N3-1
CALL REvisa (NA,NT)
LECTURA DEL ACELEROGRAMA
READ(6,*) NP
READ(6,*) DT
N4 = N3+NP
DO I = N3, N4-1
  READ(6,*) A(I)
END DO
CLOSE(6)
NT=N4-1
CALL REvisa (NA,NT)
DIMENSIONAMIENTO MEMORIA DINAMICA
N5=N4+NC*NC
N6=N5+NC
N7=N6+NC
N8=N7+NC
N9=N8+NC
N10=N9+NC
N11=N10+NC
N12=N11+NC
N13=N12+NC
N14=N13+NC
N15=N14+NC
N16=N15+NC*NP
N17=N16+NC*NP
N18=N17+NC*NP
N19=N18+NC*3
NT=N19-1
CALL REvisa (NA,NT)
DATOS INICIALES
WRITE(+,6000)
READ(+,5000)ALFA
WRITE(+,6010)
READ(+,5000)ANU
WRITE(+,6020)
READ(+,5000)GAMA
WRITE(+,6030)
READ(+,5000)BETA
CALCULO POR EL METODO BETA DE NEWMARK

```

```

CALL BETAN3 (ALFA, ANU, GAMA, BETA, DT, NP, NC, INDI,
*A(N1), A(N2), A(N3), A(N4), A(N5), A(N6), A(N7),
*A(N8), A(N9), A(N10), A(N11), A(N12), A(N13), A(N14),
*A(N15), A(N16), A(N17), A(N18))
FORMAT (F10.2)
5000 FORMAT (/, 3X, 'DAR LA CONSTANTE ALFA = ')
6000 FORMAT (/, 3X, 'DAR LA CONSTANTE NU = ')
6010 FORMAT (/, 3X, 'DAR LA CONSTANTE GAMA = ')
6020 FORMAT (/, 3X, 'DAR LA CONSTANTE BETA = ')
6030
END

```

```

SUBROUTINE PRESENTA
WRITE(*,*) 'PROGRAMA GENERALIZADO BETA DE NEWMARK'
WRITE(*,*) 'ELABORADO POR:'
WRITE(*,*) 'ALICIA GARCIA MENDOZA'
WRITE(*,*) 'LUIS FERNANDO MAGAÑA JAIME'
WRITE(*,*) 'VERSION MARZO-97'
WRITE(*,*)
RETURN
END

```

```

SUBROUTINE ABRIRA (INDI)
CHARACTER*12, NOMBREA, NOMBREB, NOMBREC
CHARACTER*15, NOMBRE1, NOMBRE2, NOMBRE3
CHARACTER*1, DR11, DR12, DR13
WRITE(*,*) 'ARCHIVO DE DATOS'
WRITE(*,*) 'DAR EL NOMBRE DEL ARCHIVO'
WRITE(*,*) 'DE LA MATRICES DE RIGIDECES Y MASAS'
WRITE(*,*) '(MAXIMO 8 CARACTERES PUNTO EXTENSION)'
READ(*, 5000) NOMBREA
WRITE(*,*) 'UNIDAD PARA LEER LOS DATOS (A,B,C,)'
READ(*, 5010) DR11
WRITE(*,*) 'DAR EL NOMBRE DEL ARCHIVO DEL ACELEROGRAMA'
WRITE(*,*) '(MAXIMO 8 CARACTERES PUNTO EXTENSION)'
READ(*, 5000) NOMBREB
WRITE(*,*) 'UNIDAD PARA LEER LOS DATOS (A,B,C,)'
READ(*, 5010) DR12
WRITE(*,*) 'DAR EL NOMBRE DEL ARCHIVO DE RESULTADOS'
WRITE(*,*) '(MAXIMO 8 CARACTERES PUNTO EXTENSION)'
READ(*, 5000) NOMBREC
WRITE(*,*) 'UNIDAD PARA LEER LOS DATOS (A,B,C,)'
READ(*, 5010) DR13
WRITE(*, 6000)
READ(*, 5020) INDI
NOMBRE1=' //DR11//': '//NOMBREA
NOMBRE2=' //DR12//': '//NOMBREB
NOMBRE3=' //DR13//': '//NOMBREC
OPEN(5, FILE=NOMBRE1, STATUS='OLD')
OPEN(6, FILE=NOMBRE2, STATUS='OLD')
OPEN(7, FILE=NOMBRE3, STATUS='NEW')
5000 FORMAT (A12)
5010 FORMAT (A1)
5020 FORMAT (I1)
6000 FORMAT (/, 2X, 'INDICADOR DE EJECUCION',
* /, 2X, 'OPCIONES',
* /, 2X, '1=IMPRESION TODO',
* /, 2X, '2=IMPRESION HISTORIA DESPLAZAMIENTOS',
* ' VELOCIDAD Y ACELERACION',
* /, 2X, '3=IMPRESION PROMEDIO DESPLAZAMIENTOS',

```

```

* ' VELOCIDAD Y ACELERACION' )
RETURN
END

SUBROUTINE REVISA (NA, NT)
IF (NA. LT. NT) THEN
WRITE (*, 6000) NA, NT
STOP
ELSE
WRITE (*, *)
WRITE (*, *) ' INFORMACION : '
WRITE (*, 6000) NA, NT
WRITE (*, *)
WRITE (*, *) ' PRESIONE RETURN PARA PODER CONTINUAR ... '
READ (*, 5000) NE
ENDIF
5000 FORMAT (I1)
6000 FORMAT (/ , 2X, ' DIMENSIONAMIENTO DE A = ', I10,
* / , 2X, ' DIMENSIONAMIENTO USADO = ', I10)
RETURN
END

SUBROUTINE BETAN3 (ALFA, ANU, GAMA, BETA, DT, NPUNT, N, IMPRI,
* AK, AM, ACEL, AMK, DESO, VELO, ACEO, VA, VB, VR,
* VAUX1, VAUX2, VAUX3, VAUX4, DES, VEL, ACE,
* PROME)
DIMENSION AK (N, N), AM (N, N), ACEL (NPUNT), AMK (N, N),
* DESO (N), VELO (N), ACEO (N), VA (N), VB (N), VR (N),
* VAUX1 (N), VAUX2 (N), VAUX3 (N), VAUX4 (N),
* DES (N, NPUNT), VEL (N, NPUNT), ACE (N, NPUNT),
* PROME (N, 3)
C CALCULO DE CONSTANTES
WRITE (*, *) ' CALCULO DE CONSTANTES '
EO=GAMA*DT
DT2=DT*DT
E1=BETA*DT2
EK1=1.+EO*ALFA
EK2=EO*ANU+E1
EA1=DT-EO
C EB1=(0.5-BETA)*DT2
IMPRESION DE RESULTADOS PARCIALES
CALL IMPRE1 (ALFA, ANU, GAMA, BETA, DT, DT2,
* EO, E1, EK1, EK2, EA1, EB1, 7, IMPRI)
C CALCULO DE MATRICES
WRITE (*, *) ' CALCULO DE MATRICES '
DO I=1, N
DO J=1, N
AMK (I, J)=EK1*AM (I, J)+EK2*AK (I, J)
ENDDO
C ENDDO
IMPRESION DE RESULTADOS PARCIALES
CALL IMPRE2 (AK, AM, AMK, N, 7, IMPRI)
C TRIANGULACION
WRITE (*, *) ' TRIANGULACION '
CALL TGCSIM (AMK, N)
C CALCULO DE VECTORES
DO I=1, N
VA (I)=0.
VB (I)=0.
DESO (I)=0.0

```

```

      VELO(I)=0.0
      ACEO(I)=0.0
      VAUX4(I)=1.0
ENDDO
DT1=DT
DO K=1,NPUNT
  WRITE(*,*)'NO. DE PUNTO',K
  DO I=1,N
    VA(I)=VELO(I)+EA1*ACEO(I)
    VB(I)=DESO(I)+DT*VELO(I)+EB1*ACEO(I)
  ENDDO
  DO I=1,N
    VAUX1(I)=ANU*VA(I)+VB(I)
  ENDDO
  CALL MULMAT (AK,VAUX1,VAUX2,N,N,N,1)
  CALL MULMAT (AM,VA,VAUX3,N,N,N,1)
  DO I=1,N
    VAUX1(I)=ACEL(K)
  ENDDO
  CALL MULMAT (AM,VAUX1,VAUX4,N,N,N,1)
  DO I=1,N
    VR(I)=-VAUX4(I)-ALFA*VAUX3(I)-VAUX2(I)
  ENDDO
C   IMPRESION DE RESULTADOS PARCIALES
  CALL IMPRE3(K,VA,VB,VR,VAUX2,VAUX3,N,7,
*IMPRI)
C   SOLUCION DEL SISTEMA
  CALL SGCSIM(AMK,VR,N)
  DO I=1,N
    ACEO(I)=VR(I)
    VELO(I)=VA(I)+EO*ACEO(I)
    DESO(I)=VB(I)+E1*ACEO(I)
  ENDDO
  DO I=1,N
    DES(I,K)=DESO(I)
    VEL(I,K)=VELO(I)
    ACE(I,K)=ACEO(I)
    PROME(I,1)=PROME(I,1)+(DESO(I)*DESO(I))
    PROME(I,2)=PROME(I,2)+(VELO(I)*VELO(I))
    PROME(I,3)=PROME(I,3)+(ACEO(I)*ACEO(I))
  ENDDO
  DT=DT+DT1
C   ENDDO
  DO I=1,N
    PROME(I,1)=SQRT(PROME(I,1))
    PROME(I,2)=SQRT(PROME(I,2))
    PROME(I,3)=SQRT(PROME(I,3))
  ENDDO
C   IMPRESION DE RESULTADOS
  CALL IMPRE4(DES,VEL,ACE,PROME,N,NPUNT,7,IMPRI)
  RETURN
END

SUBROUTINE IMPRE1(ALFA,ANU,GAMA,BETA,DT,DT2,
*EO,E1,EK1,EK2,EA1,EB1,NAR,IMPRI)
  IF (IMPRI.EQ.1)THEN
    WRITE(NAR,6000)ALFA,ANU,GAMA,BETA,DT,DT2,
*EO,E1,EK1,EK2,EA1,EB1
  ENDF
6000  FORMAT(/,2X,'VALORES DE LAS CONSTANTES',

```

```

*      /, 2X, 'ALFA=' , F15.8,
*      /, 2X, 'ANU  =' , F15.8,
*      /, 2X, 'GAMA=' , F15.8,
*      /, 2X, 'BETA=' , F15.8,
*      /, 2X, 'DAT='  , F15.8,
*      /, 2X, 'DAT2=' , F15.8,
*      /, 2X, 'EO='   , F15.8,
*      /, 2X, 'E1='   , F15.8,
*      /, 2X, 'EK1='  , F15.8,
*      /, 2X, 'EK2='  , F15.8,
*      /, 2X, 'EA1='  , F15.8,
*      /, 2X, 'EB1='  , F15.8, )
RETURN
END

SUBROUTINE IMPRE2(AK,AM,AMK,N,NAR,IMPRI)
DIMENSION AK(N,N),AM(N,N),AMK(N,N)
IF (IMPRI.EQ.1) THEN
WRITE (NAR, 6000)
CALL IMAAR (AK,N,N,NAR)
WRITE (NAR, 6010)
CALL IMAAR (AM,N,N,NAR)
WRITE (NAR, 6020)
CALL IMAAR (AMK,N,N,NAR)
ENDIF
6000 FORMAT (/, 2X, 'MATRIZ DE RIGIDECES')
6010 FORMAT (/, 2X, 'MATRIZ DE MASAS')
6020 FORMAT (/, 2X, 'MATRIZ AMK')
RETURN
END

SUBROUTINE IMPRE3(K,VA,VB,VR,VAUX2,VAUX3,N,
*NAR,IMPRI)
DIMENSION VA(N),VB(N),VR(N),VAUX2(N),VAUX3(N)
IF (IMPRI.EQ.1) THEN
WRITE (NAR, 6000) K
WRITE (NAR, 6010)
CALL IMAAR (VA,N,1,NAR)
WRITE (NAR, 6020)
CALL IMAAR (VB,N,1,NAR)
WRITE (NAR, 6030)
CALL IMAAR (VAUX2,N,1,NAR)
WRITE (NAR, 6040)
CALL IMAAR (VAUX3,N,1,NAR)
WRITE (NAR, 6050)
CALL IMAAR (VR,N,1,NAR)
ENDIF
6000 FORMAT (/, 2X, 'NO. DE ITERACION', I2)
6010 FORMAT (/, 2X, 'VECTOR VA')
6020 FORMAT (/, 2X, 'VECTOR VB')
6030 FORMAT (/, 2X, 'VECTOR VAUX2')
6040 FORMAT (/, 2X, 'VECTOR VAUX3')
6050 FORMAT (/, 2X, 'VECTOR VR')
RETURN
END

SUBROUTINE IMPRE4 (DES,VEL,ACE,PROME,N,NPUNT,NAR,IMPRI)
DIMENSION DES(N,NPUNT),VEL(N,NPUNT),ACE(N,NPUNT),
*PROME(N,3)
IF (IMPRI.EQ.2) THEN
WRITE (NAR, 6000)

```

```

CALL IMAAR(DES,N,NPUNT,NAR)
WRITE(NAR,6010)
CALL IMAAR(VEL,N,NPUNT,NAR)
WRITE(NAR,6020)
CALL IMAAR(ACE,N,NPUNT,NAR)
ENDIF
IF(IMPRI.LE.3)THEN
WRITE(NAR,6030)
CALL IMAAR(PROME,N,3,NAR)
ENDIF
6000 FORMAT(/,2X,'HISTORIA DE DESPLAZAMIENTOS')
6010 FORMAT(/,2X,'HISTORIA DE VELOCIDADES')
6020 FORMAT(/,2X,'HISTORIA DE ACELERACIONES')
6030 FORMAT(/,2X,'PROMEDIO DE DESPLAZAMIENTOS, VELOCIDADES'
*' Y ACELERACIONES')
RETURN
END

```

```

SUBROUTINE TGCSIM(A,N)
DIMENSION A(N,N)
A(1,2)=A(1,2)/A(1,1)
A(2,2)=A(2,2)-A(1,1)*A(1,2)**2
IF(N.EQ.2) GOTO 50
DO J=3,N
IS=J-1
A(1,J)=A(1,J)/A(1,1)
DO I=2,IS
S=0
KS=I-1
DO K=1,KS
S=S+A(K,I)*A(K,K)*A(K,J)
ENDDO
A(I,J)=(A(I,J)-S)/A(I,I)
ENDDO
S1=0
DO K=1,IS
S1=S1+A(K,K)*A(K,J)**2
ENDDO
A(J,J)=A(J,J)-S1
ENDDO
RETURN
END
50

```

```

SUBROUTINE SGCSIM(A,B,N)
DIMENSION A(N,N),B(N)
DO I=2,N
KS=I-1
S=0
DO K=1,KS
S=S+A(K,I)*B(K)
ENDDO
B(I)=B(I)-S
ENDDO
B(N)=B(N)/A(N,N)
IS=N-1
DO I=1,IS
IA=N-I
KI=IA+1
S=0
DO K=KI,N

```

```

      S=S+A(IA,K)*B(K)
    ENDDO
    B(IA)=B(IA)/A(IA,IA)-S
  ENDDO
  RETURN
END

```

```

SUBROUTINE MULMAT(A,B,C,N1,N2,M1,M2)
DIMENSION A(N1,N2),B(M1,M2),C(N1,M2)
DO K=1,N1
  DO I=1,M2
    C(K,I)=0.0
    DO J=1,N2
      C(K,I)=C(K,I)+A(K,J)*B(J,I)
    ENDDO
  ENDDO
ENDDO
RETURN
END

```

```

SUBROUTINE IMAAR(A, NR, NC, NAR)
DIMENSION A(NR, NC)
DO 100 L=1,NC,5
  M = L + 4
  IF (M.GT.NC) M=NC
  WRITE(NAR,6000)(K,K=L,M)
  DO 100 I=1,NR
    WRITE(NAR,6010)I,(A(I,J),J=L,M)
100  CONTINUE
  RETURN
6000  FORMAT(/,9X,I4,4(10X,I4))
6010  FORMAT(1X,I4,5(1PE14.6))
  RETURN
END

```