

15
24



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ACATLAN

SISTEMA PARA EL ANALISIS DE PLACAS POR
MEDIO DEL 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 :

María Guadalupe Vera Chávez

Director de Tesis:
M. en I. Fernando Vera Badillo

**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 está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

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

INDICE

	pág.
INTRODUCCION	1
CAPITULO I ANALISIS DE PLACAS CON LA TEORIA CLASICA.	4
1.1 <i>Hipótesis de la teoría clásica.</i>	5
1.2 <i>Esfuerzos y deformaciones.</i>	6
1.3 <i>Relaciones momentos-esfuerzos y relaciones momentos-deformaciones.</i>	8
1.4 <i>Equilibrio del elemento placa.</i>	10
1.5 <i>Relaciones cortantes-deformaciones y relaciones cortantes-esfuerzos.</i>	11
1.6 <i>Ecuaciones de equilibrio y ecuaciones de deflexión de la placa.</i>	12
CAPITULO II EL METODO DEL ELEMENTO FINITO.	13
2.1 <i>Hipótesis básicas.</i>	14
2.2 <i>Ecuaciones generales de Hooke y Cauchy.</i>	15
2.3 <i>Ecuaciones de Hooke para placas.</i>	18
2.4 <i>Aproximación de la solución con el método del elemento finito.</i>	20
2.5 <i>Matriz de deformaciones, matriz de rigideces y vector de fuerzas de cuerpo.</i>	23
2.6 <i>Proceso de solución por el método del elemento finito.</i>	26

	pág.	
CAPITULO III	ELEMENTOS QUE INTERVIENEN EN LA GRAFICACION.....	28
3.1	<i>Uso de las gráficas.....</i>	29
3.1.1	<i>Verificación de datos.....</i>	30
3.1.2	<i>Interpretación de resultados.....</i>	31
3.2	<i>Conceptos generales de graficación.....</i>	32
3.3	<i>Gráficas en modo texto.....</i>	33
3.3.1	<i>Requerimiento de memoria.....</i>	34
3.3.2	<i>Mapeo de memoria.....</i>	35
3.4	<i>Gráficas en modo gráfico.....</i>	36
3.4.1	<i>Requerimiento de memoria.....</i>	36
3.4.2	<i>Mapeo de memoria.....</i>	36
3.5	<i>Instrucciones de graficación del lenguaje C.....</i>	37
3.6	<i>Análisis de regresión.....</i>	41
CAPITULO IV	DESCRIPCION Y REQUERIMIENTOS DEL SISTEMA.....	44
4.1	<i>Desarrollo del sistema.....</i>	45
4.2	<i>Organización del sistema.....</i>	49
4.3	<i>Módulo de recepción de datos.....</i>	51
4.4	<i>Módulo de análisis de placas.....</i>	55
4.5	<i>Manual del usuario.....</i>	57
4.5.1	<i>Instructivo de operación del módulo de Recepción de datos.....</i>	61
4.5.2	<i>Instructivo de operación del módulo de Análisis de placas.....</i>	66

	pág.
CAPITULO V EJEMPLOS DE APLICACION.	67
5.1 <i>Análisis de una placa de concreto de geometría regular, características del problema</i>	67
5.2 <i>Discretización de la placa</i>	68
5.3 <i>Verificación de los datos a través de gráficas</i>	71
5.4 <i>Comparación de resultados</i>	72
5.5 <i>Placa de concreto, de geometría irregular</i> ..	74
 CONCLUSIONES.....	 75
 GLOSARIO DE TERMINOS.....	 80
 APENDICE A FIGURAS.....	 84
 APENDICE B CUADROS.....	 99
 APENDICE C LISTADO DE LOS PROGRAMAS.....	 118
 BIBLIOGRAFIA.....	 151

INTRODUCCION

Uno de los problemas más frecuentes en el ramo de la ingeniería civil, es el diseño de estructuras en la construcción, ya que éstas se deben analizar bajo diferentes condiciones de carga.

De las estructuras que se estudian en la ingeniería civil nos centraremos sólo en el análisis de aquellas que tienen el espesor pequeño comparado con sus otras dos dimensiones y la carga es perpendicular a su plano, a este tipo de estructuras se les conoce como placas. El modelo matemático de las placas resulta sencillo de resolver cuando éstas tienen geometría regular, condiciones de apoyo simple y están formadas de un sólo material. Pero cuando no se cumple lo anterior, la teoría de placas no cuenta con los recursos suficientes para resolverlo en forma sencilla e incluso resulta a veces demasiado complicada.

Puede utilizarse el Método del Elemento Finito (M.E.F.^{*}) como una opción para conocer el comportamiento mecánico (esfuerzos) y cinemático (deformaciones) de placas de cualquier geometría, cualquier condición de apoyo y formadas por más de un material.

* En lo sucesivo se usaran estas siglas (M.E.F.) para abreviar al Método del Elemento Finito.

El M.E.F. involucra un gran número de operaciones a resolver, esto hace indispensable el uso de una computadora. Lo anterior fue motivo para elaborar el presente trabajo, cuyo título es Sistema para el Análisis de Placas por medio del Método del Elemento Finito.

El objetivo central de esta tesis, es diseñar y desarrollar un sistema que permita hacer un análisis de flexión de placas de diversas geometrías, formadas por más de un material y con diferentes condiciones de apoyo, con una solución propuesta por el M.E.F. El sistema deberá realizar la captura de datos de manera interactiva y sencilla para el usuario del mismo, la graficación de datos para su verificación y la graficación de resultados para su interpretación.

En el primer capítulo se presenta el método analítico (teoría clásica) usado para el análisis de placas. Aquí se plantean las ecuaciones para el cálculo de esfuerzos y deformaciones de placas.

En el segundo capítulo se mencionan las hipótesis básicas del M.E.F., las ecuaciones de Hooke para el análisis de placas y a partir de éstas se llega al proceso de solución del M.E.F. para analizar placas de diversas geometrías, diferentes condiciones de apoyo y que estén formadas por más de un material.

En el tercer capítulo se explican los elementos que intervienen en la elaboración de las gráficas que sirven de apoyo en la verificación de datos y en la interpretación de resultados.

En el cuarto capítulo se describe el Sistema de Análisis de Placas a través del M.E.F., el cual se desarrolló partiendo de los capítulos dos y tres. Por otra parte, se explican sus requerimientos de operación junto con las subrutinas y archivos empleados. Además, se incluye el Manual del Usuario, en donde se detallan los pasos a seguir para utilizar adecuadamente el sistema.

En el quinto y último capítulo, como un ejemplo de aplicación, se hace el análisis de una placa de concreto de geometría regular, mediante su discretización por medio del M.E.F. y la verificación de datos e interpretación de resultados en forma gráfica. También se incluye el análisis de una placa de concreto de geometría irregular, como una generalización del Sistema.

Después del quinto capítulo se encuentran las conclusiones, en donde se muestran los principales resultados del presente trabajo.

Debido a que parte del material utilizado en esta tesis es específico de ingeniería civil, se incluye un glosario con los términos de mayor uso.

Como último punto se encuentran los apéndices, que contienen la siguiente información: desplegados en el monitor de la computadora por el sistema, las hojas de codificación de datos para la captura, figuras, cuadros y los listados de los programas.

CAPITULO I ANALISIS DE PLACAS CON LA TEORIA CLASICA

En ingeniería civil, toda estructura queda definida por su geometría, su material y por sus cargas. Una placa es una estructura, la cual es un cuerpo geométrico cuyo espesor es pequeño comparado con sus otras dos dimensiones. El análisis de una placa se hace con el objeto de conocer su comportamiento mecánico (los esfuerzos) y su comportamiento cinemático (las deformaciones que sufre la placa bajo diferentes condiciones de carga).

En el presente capítulo se hace un análisis de placas mediante la teoría clásica. Esta es una teoría matemática que por medio de ecuaciones diferenciales analiza las deformaciones de una placa dando una solución única (solución cerrada), mientras que el M.E.F. da una solución aproximada.

La teoría clásica sirve únicamente para analizar placas con geometría regular, con una condición de apoyo y que estén elaboradas de un sólo material, por lo que deja fuera de estudio aquellas placas con geometría irregular, diferentes condiciones de apoyo y que estén compuestas por más de un material.

1.1 HIPOTESIS DE LA TEORIA CLASICA.

Para estudiar el comportamiento de las placas se toman en cuenta las siguientes hipótesis:

- 1.- El espesor de la placa debe ser constante y pequeño comparado con sus otras dimensiones.
- 2.- El material de la placa debe ser homogéneo, continuo, isotrópico y obedece a la ley de Hooke (la cual se mencionará más adelante).
- 3.- Las deflexiones de la placa deben ser pequeñas en relación con el espesor, por lo que sólo se considerará a la deflexión de la superficie media de la placa.
- 4.- Cualquier línea perpendicular a la superficie media de la placa antes de la flexión debe permanecer recta y normal a la superficie media después de la flexión.
- 5.- Se despreciarán los esfuerzos normales transversales a la superficie media.
- 6.- Sólo actuarán sobre la placa cargas perpendiculares a su plano.

En las siguientes secciones se construirá un modelo matemático de las leyes que gobiernan el comportamiento de las placas.

1.2 ESFUERZOS Y DEFORMACIONES.

Según la ley de Hooke, los esfuerzos se definen como:

$$\begin{aligned}\sigma_{xx} &= \frac{E}{1-\nu^2} (\epsilon_{xx} + \nu\epsilon_{yy}) \\ \sigma_{yy} &= \frac{E}{1-\nu^2} (\epsilon_{yy} + \nu\epsilon_{xx}) \\ \tau_{xy} &= \frac{E}{1+\nu} \epsilon_{xy}\end{aligned}\tag{1.2.1}$$

donde:

$\sigma_{xx}, \sigma_{yy}, \tau_{xy}$	=	esfuerzos
$\epsilon_{xx}, \epsilon_{yy}, \epsilon_{xy}$	=	deformaciones
E	=	módulo de elasticidad
ν	=	relación de Poisson

y las deformaciones son:

$$\begin{aligned}\epsilon_{xx} &= -z \frac{\delta^2 v}{\delta x^2} \\ \epsilon_{yy} &= -z \frac{\delta^2 v}{\delta y^2} \\ \epsilon_{xy} &= -2z \frac{\delta^2 v}{\delta x \delta y}\end{aligned}\tag{1.2.2}$$

donde:

z = distancia a partir de la superficie media.

$\frac{\delta^2 v}{\delta x^2}$ = curvatura en "x" producida por M_x y M_y .

$\frac{\delta v}{\delta x}$ = desplazamiento angular en "x".

$\frac{\delta^2 v}{\delta y^2}$ = curvatura en "y" producida en M_x y M_y .

$\frac{\delta v}{\delta y}$ = desplazamiento angular en "y".

$\frac{\delta^2 v}{\delta x \delta y}$ = distorsión o cambio de la pendiente en "x" según se avanza en "y".

w = deflexión de la placa en la dirección de la carga en el punto (x,y) .

1.3 RELACIONES MOMENTOS-ESFUERZOS Y
RELACIONES MOMENTOS-DEFORMACIONES.

La relación entre momentos y esfuerzos es:

$$\begin{aligned}
 M_x &= \int_{-h/2}^{h/2} \sigma_{xx} z dz \\
 M_y &= \int_{-h/2}^{h/2} \sigma_{yy} z dz \\
 M_{xy} &= \int_{-h/2}^{h/2} \tau_{xy} z dz
 \end{aligned}
 \tag{1.3.1}$$

o bien:

$$\begin{aligned}
 \sigma_{xx} &= \frac{12 M_x}{h^3} z \\
 \sigma_{yy} &= \frac{12 M_y}{h^3} z \\
 \tau_{xy} &= \frac{12 M_{xy}}{h^3} z
 \end{aligned}
 \tag{1.3.2}$$

Al sustituir las ecuaciones 1.2.1 y 1.2.2 en 1.3.1 se tiene:

$$M_x = -D \left[\frac{\delta^2 w}{\delta x^2} + \nu \frac{\delta^2 w}{\delta y^2} \right]$$

$$M_y = -D \left[\frac{\delta^2 w}{\delta y^2} + \nu \frac{\delta^2 w}{\delta x^2} \right] \quad (1.3.3)$$

$$M_{xy} = -D (1-\nu) \frac{\delta^2 w}{\delta x \delta y}$$

donde :

$$D = \frac{Eh^3}{12(1-\nu^2)}$$

D = rigidez a la flexión de la placa.

E = módulo de elasticidad del material de la placa.

ν = coeficiente de poisson.

h = espesor de la placa.

z = medio espesor de la placa.

M_x = momento flexionante en "x".

M_y = momento flexionante en "y".

M_{xy} = momento torsionante.

1.4 EQUILIBRIO DEL ELEMENTO PLACA.

Al utilizar las ecuaciones de equilibrio en la figura 1.1, se tiene:

$$\Sigma F_z = 0 : \quad \frac{\delta V_x}{\delta x} + \frac{\delta V_y}{\delta y} + q = 0 \quad (1.4.1)$$

$$\Sigma M_x = 0 : \quad \frac{\delta M_x}{\delta x} + \frac{\delta M_{xy}}{\delta y} = V_x \quad (1.4.2)$$

$$\Sigma M_y = 0 : \quad \frac{\delta M_y}{\delta y} + \frac{\delta M_{xy}}{\delta x} = V_y \quad (1.4.3)$$

donde :

q = carga actuante sobre la placa por área unitaria; es una función de "x" y de "y".

1.5 RELACIONES CORTANTES-DEFORMACIONES Y
RELACIONES CORTANTES-ESFUERZOS.

Al sustituir las ecuaciones 1.3.3 en las ecuaciones 1.4.2 y 1.4.3 se obtienen las relaciones entre cortantes y deformaciones:

$$V_x = -D \frac{\delta}{\delta x} \left[\frac{\delta^2 w}{\delta x^2} + \frac{\delta^2 w}{\delta y^2} \right]$$

$$V_y = -D \frac{\delta}{\delta y} \left[\frac{\delta^2 w}{\delta x^2} + \frac{\delta^2 w}{\delta y^2} \right] \quad (1.5.1)$$

Para que se cumpla el equilibrio en la dirección x , sin considerar fuerzas de cuerpo, debe cumplirse que:

$$\frac{\delta \tau_{xx}}{\delta x} + \frac{\delta \tau_{xy}}{\delta y} + \frac{\delta \tau_{xz}}{\delta z} = 0 \quad (1.5.2)$$

donde:

$$\tau_{xz} = \frac{3}{2} \frac{V_x}{h} \left[1 - 4 \frac{z^2}{h^2} \right]$$

$$\tau_{yz} = \frac{3}{2} \frac{V_y}{h} \left[1 - 4 \frac{z^2}{h^2} \right]$$

1.6 ECUACIONES DE EQUILIBRIO Y ECUACIONES DE DEFLEXION DE LA PLACA.

La sustitución de las ecuaciones 1.4.2 en la ecuación 1.4.1 genera la ecuación de equilibrio de la placa:

$$\frac{\delta^2 M_x}{\delta x^2} + 2 \frac{\delta M_{xy}}{\delta x \delta y} + \frac{\delta^2 M_y}{\delta y^2} + q = 0 \quad (1.6.1)$$

Al sustituir las ecuaciones 1.3.3 en la 1.6.1 se encuentra la ecuación diferencial parcial de cuarto orden que define a las deformaciones elásticas de placas isotrópicas con cargas normales a su plano.

$$\frac{\delta^4 w}{\delta x^4} + 2 \frac{\delta^4 w}{\delta x^2 \delta y^2} + \frac{\delta^4 w}{\delta y^4} = \frac{q}{D} \quad (1.6.2)$$

Existen soluciones cerradas para analizar placas, pero sólo para casos específicos, es decir, existe solución para placas simplemente apoyadas (propuesta por Navier), para placas rectangulares (propuesta por Lévy), para placas elípticas y circulares, empotradas y bajo cargas uniformes. Pero para resolver problemas de placas de forma irregular y diferentes condiciones de carga, se propone un método más general y éste es el M.F.F., mismo que se detalla en el siguiente capítulo.

CAPITULO II EL METODO DEL ELEMENTO FINITO

Para estudiar el comportamiento de placas bajo diferentes condiciones de carga, en la etapa de diseño, es necesario idealizarlas mediante modelos matemáticos, como se explicó en el capítulo uno o mediante simulaciones (modelos experimentales), lo cual es muy costoso. Por otro lado, la solución analítica en muchas situaciones es difícil de obtener cuando la placa tiene geometría irregular, ya que matemáticamente es imposible describir su frontera.

Existe una gran variedad de procedimientos, como se mencionó en el capítulo anterior, que resuelven casos de placas con características específicas. En el presente capítulo se presenta el M.E.F., el cual es un método general de solución aproximada, que permite analizar medios continuos, en especial cuando la geometría de éstos es bastante compleja, tienen diferentes condiciones de apoyo y están compuestas por más de un material. De aquí que el M.E.F. resulta ser el idóneo para el análisis de placas en general.

En el presente capítulo se estudian las hipótesis básicas del M.E.F., planteándose las ecuaciones específicas para el análisis de placas y se detalla el procedimiento para analizarlas.

2.1 HIPÓTESIS BÁSICAS.

Las hipótesis del M.E.F. son las siguientes :

- 1.- El medio continuo se divide, mediante líneas, en un número finito de regiones (triángulos, cuadriláteros, tetraedros, hexaedros, etc) , a los cuales llamaremos "elementos finitos".
- 2.- Dichos elementos están interconectados en sus puntos nodales. Los puntos nodales están situados sobre las fronteras de los elementos. Los desplazamientos de los puntos nodales son las incógnitas principales del problema.
- 3.- Se define en forma única, el campo de desplazamientos en cualquier punto del elemento finito, en función de los desplazamientos de los puntos nodales.
- 4.- Conocido el campo de desplazamientos, se definen en forma única, los campos de deformaciones (tensor de deformaciones) y los de esfuerzos (tensor de esfuerzos), en función de los desplazamientos de los puntos nodales.

2.2 ECUACIONES GENERALES DE HOOKE Y CAUCHY.

Las ecuaciones de Hooke y Cauchy, conocidas también como ecuaciones de la ley de Hooke generalizada, o ecuaciones constitutivas del material sólido, elástico, lineal e isótropo, se presentan a continuación en notación tradicional:

$$\sigma_{xx} = \frac{E}{(1+\nu)(1-2\nu)} [(1-\nu) \epsilon_{xx} + \nu (\epsilon_{yy} + \epsilon_{zz})] \quad (2.2.1)$$

$$\sigma_{yy} = \frac{E}{(1+\nu)(1-2\nu)} [(1-\nu) \epsilon_{yy} + \nu (\epsilon_{zz} + \epsilon_{xx})] \quad (2.2.2)$$

$$\sigma_{zz} = \frac{E}{(1+\nu)(1-2\nu)} [(1-\nu) \epsilon_{zz} + \nu (\epsilon_{xx} + \epsilon_{yy})] \quad (2.2.3)$$

$$\sigma_{xy} = \frac{E}{2(1+\nu)} \gamma_{xy} = G\gamma_{xy} \quad (2.2.4)$$

$$\sigma_{yz} = \frac{E}{2(1+\nu)} \gamma_{yz} = G\gamma_{yz} \quad (2.2.5)$$

$$\sigma_{zx} = \frac{E}{2(1+\nu)} \gamma_{zx} = G\gamma_{zx} \quad (2.2.6)$$

$$\sigma_{xx} = \frac{1}{E} [\sigma_{xx} - \nu (\sigma_{yy} + \sigma_{zz})] \quad (2.2.7)$$

$$\sigma_{yy} = \frac{1}{E} [\sigma_{yy} - \nu (\sigma_{zz} + \sigma_{xx})] \quad (2.2.8)$$

$$\sigma_{zz} = \frac{1}{E} [\sigma_{zz} - \nu (\sigma_{xx} + \sigma_{yy})] \quad (2.2.9)$$

$$\gamma_{xy} = \frac{2(1+\nu)}{E} \sigma_{xy} = \frac{1}{G} \sigma_{xy} \quad (2.2.10)$$

$$\gamma_{yz} = \frac{2(1+\nu)}{E} \sigma_{yz} = \frac{1}{G} \sigma_{yz} \quad (2.2.11)$$

$$\gamma_{zx} = \frac{2(1+\nu)}{E} \sigma_{zx} = \frac{1}{G} \sigma_{zx} \quad (2.2.12)$$

y en forma matricial se ordenan así:

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{bmatrix} = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & \nu & 0 & 0 & 0 \\ \nu & 1 & \nu & 0 & 0 & 0 \\ \nu & \nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}(1-\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}(1-\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}(1-\nu) \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix} \quad (2.2.13)$$

y en forma aislada, tenemos dos vectores y una matriz:

$$[\sigma] = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{bmatrix} \quad [\epsilon] = \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix}$$

(2.2.14)

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

donde:

[σ] = vector de esfuerzos.

[ϵ] = vector de deformaciones.

[D] = matriz de coeficientes elásticos.

La notación matricial anterior se reduce a su forma simbólica simple:

$$[\sigma] = [D] [\epsilon] \quad (2.2.15)$$

2.3 ECUACIONES DE HOOKE PARA PLACAS.

Las ecuaciones generales de Hooke y Cauchy pueden simplificarse según el caso que se presente. Cuando se trata de placas, que es lo concerniente a este escrito, se toman las ecuaciones de Hooke y Cauchy con una sola simplificación; ésta es: "Las secciones transversales permanecen planas antes de que se presente la deformación y después de que ésta se produzca"; entonces las ecuaciones, en notación matricial, toman la forma siguiente:

HIPOTESIS $\Rightarrow \epsilon_{zz} = 0 ; \sigma_{zz} = 0$

$$[\sigma] = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{bmatrix} = \begin{bmatrix} [\sigma_b] \\ [\sigma_s] \end{bmatrix} \quad [\epsilon] = \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix} = \begin{bmatrix} [\epsilon_b] \\ [\epsilon_s] \end{bmatrix} \quad (2.3.1)$$

$$[D] = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & \nu & 0 & 0 & 0 \\ \nu & 1 & \nu & 0 & 0 & 0 \\ \nu & \nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}(1-\nu) & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \frac{1}{2}(1-\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}(1-\nu) \end{bmatrix} = \begin{bmatrix} [D_b] \\ [D_s] \end{bmatrix} \quad (2.3.2)$$

donde:

$[D_b]$ = parte a flexión de la matriz $[D]$.

$[D_s]$ = parte a cortante de la matriz $[D]$.

$[\sigma_b]$ = parte a flexión del vector de esfuerzos.

$[\sigma_s]$ = parte a cortante del vector de esfuerzos.

$[\epsilon_b]$ = parte a flexión del vector de deformaciones.

$[\epsilon_s]$ = parte a cortante del vector de deformaciones.

2.4 APROXIMACION DE LA SOLUCION CON EL METODO DEL ELEMENTO FINITO.

El vector de deformaciones para el análisis de placas es:

$$[\epsilon] = \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{bmatrix} \quad (2.4.1)$$

donde:

$$\epsilon_{xx} = \frac{\delta u}{\delta x}$$

$$\epsilon_{yy} = \frac{\delta v}{\delta y}$$

$$\gamma_{xy} = \frac{\delta u}{\delta y} + \frac{\delta v}{\delta x}$$

$$\gamma_{yz} = \frac{\delta v}{\delta z} + \frac{\delta w}{\delta y}$$

$$\gamma_{zx} = \frac{\delta w}{\delta x} + \frac{\delta u}{\delta z}$$

entonces:

$$[\varepsilon] = \begin{bmatrix} \frac{\delta u}{\delta x} \\ \frac{\delta v}{\delta y} \\ \frac{\delta u}{\delta y} + \frac{\delta v}{\delta x} \\ \frac{\delta u}{\delta z} + \frac{\delta w}{\delta y} \\ \frac{\delta w}{\delta x} + \frac{\delta u}{\delta z} \end{bmatrix} = \begin{bmatrix} \frac{\delta}{\delta x} & 0 & 0 \\ 0 & \frac{\delta}{\delta y} & 0 \\ \frac{\delta}{\delta y} & \frac{\delta}{\delta x} & 0 \\ 0 & \frac{\delta}{\delta z} & \frac{\delta}{\delta y} \\ \frac{\delta}{\delta z} & 0 & \frac{\delta}{\delta x} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.4.2)$$

en forma simbólica:

$$[\varepsilon] = [L] [U] \quad (2.4.3)$$

donde:

$$[L] = \begin{bmatrix} \frac{\delta}{\delta x} & 0 & 0 \\ 0 & \frac{\delta}{\delta y} & 0 \\ \frac{\delta}{\delta y} & \frac{\delta}{\delta x} & 0 \\ 0 & \frac{\delta}{\delta z} & \frac{\delta}{\delta y} \\ \frac{\delta}{\delta z} & 0 & \frac{\delta}{\delta x} \end{bmatrix} \quad [U] = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.4.4)$$

(2.4.5)

El método del elemento finito considera la siguiente aproximación:

$$\{ U \} \approx \{ \bar{U} \} = [N] \{ \bar{U} \} \quad (2.4.6)$$

donde:

$\{ U \}$ = vector de desplazamientos.

$\{ \bar{U} \}$ = solución aproximada del vector de desplazamientos

$\{ \bar{U} \}$ = desplazamientos que se obtienen al aplicar el método.

$[N]$ = matriz de funciones de forma.

entonces:

$$\{ \epsilon \} = [L] \{ U \} = [L] [N] \{ \bar{U} \} \quad (2.4.7)$$

y también:

$$\{ \sigma \} = [D] \{ \epsilon \} = [D] [L] [N] \{ \bar{U} \} \quad (2.4.8)$$

se define:

$$[B] = [L] [N] \quad (2.4.9)$$

donde:

$[B]$ = matriz de deformaciones.

al sustituir la ecuación 2.4.9. en 2.4.7. y 2.4.8., se tiene:

$$\{ \epsilon \} = [B] \{ \bar{U} \} \quad (2.4.10)$$

$$\{ \sigma \} = [D] [B] \{ \bar{U} \} \quad (2.4.11)$$

y la parte a cortante de la matriz de deformaciones es de 2 renglones por 24 columnas, los cuales equivalen al cuarto y quinto renglones de la matriz [B]:

$$[B_{\bullet}] = \begin{bmatrix}
 -N_1 & -N_2 & -N_3 & -N_4 & -N_5 & -N_6 & -N_7 & -N_8 & \gg \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gg \\
 \ll 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gg \\
 \ll N_1 & N_2 & N_3 & N_4 & N_5 & N_6 & N_7 & N_8 & \gg \\
 \ll N_{1,y} & N_{2,y} & N_{3,y} & N_{4,y} & N_{5,y} & N_{6,y} & N_{7,y} & N_{8,y} & \gg \\
 \ll N_{1,x} & N_{2,x} & N_{3,x} & N_{4,x} & N_{5,x} & N_{6,x} & N_{7,x} & N_{8,x} & \gg
 \end{bmatrix} \quad (2.5.3)$$

donde:

$$N_1, N_2, \dots, N_8 = \text{Funciones de forma.}$$

en forma simbólica, la división de [B] se presenta así:

$$[B] = \begin{bmatrix} [B_b]_{(2 \times 24)} \\ \hline [B_{\bullet}]_{(2 \times 24)} \end{bmatrix} \quad (2.5.4)$$

La matriz de rigideces de cada elemento se obtiene con base en el principio del trabajo virtual, el cual representa a la ecuación variacional en las ecuaciones de la elasticidad lineal, y se presenta así:

$$[K]^e = \int_A \int_{-z/2}^{+z/2} \left[z^2 [B_b]^T [D_b] [B_b] + [B_s]^T [D_s] [B_s] \right] dz dA \quad (2.5.5)$$

donde:

- $[K]^e$ = matriz de rigideces de cada elemento.
- $[Z]$ = espesor de cada elemento.
- $[B_b]$ = parte a flexión de la matriz de deformaciones.
- $[B_s]$ = parte a cortante de la matriz de deformaciones.
- $[D]$ = matriz de coeficientes elásticos.

El vector de fuerzas de cuerpo es el siguiente:

$$[f_c]^e = \int_{\Omega^e} \rho [N]^T [f] d\Omega \quad (2.5.6)$$

donde:

- $[f_c]^e$ = vector de fuerzas de cuerpo de cada elemento.
- Ω^e = región del elemento
- ρ = peso volumétrico del material.
- $[N]^T$ = vector transpuesto de funciones de forma.
- $[f]$ = vector de fuerzas por unidad de masa

2.6 PROCESO DE SOLUCION POR EL METODO DEL ELEMENTO FINITO.

Para analizar una placa con el M.E.F. deben llevarse a cabo los siguientes pasos:

- 1.- La placa se divide en un número finito de regiones llamadas elementos finitos, los cuales están interconectados por los nodos situados en la frontera de cada elemento. A estos nodos se les llama puntos nodales.
- 2.- Una vez definida la malla se numeran los nodos y los elementos en un sistema de referencia cartesiano global, se obtienen las coordenadas de cada punto nodal con sus restricciones frontera, es decir, el punto nodal puede tener desplazamiento libre o restringido en alguna de sus componentes.
- 3.- Se calcula la matriz de rigideces de cada elemento con la ecuación 2.5.5, con los valores correspondientes según los datos de cada elemento como son el tipo de material y coordenadas.
- 4.- Se ensambla la matriz de rigideces de la estructura en base a la matriz de rigideces de cada elemento.

5.- Se forma el vector de cargas de la estructura con las cargas actuantes en los puntos nodales.

6.- En el caso de considerar fuerzas de cuerpo, éstas se calculan con la ecuación 2.5.6 y se suman en el vector de cargas de la estructura.

7.- Con la matriz de rigideces de la estructura y el vector de cargas de la estructura, ambos conocidos, se resuelve la siguiente ecuación:

$$[K] [U] = [P]$$

en donde [U] es el vector de desplazamientos de la estructura que representa a las incógnitas.

8.- Se obtienen los desplazamientos de cada elemento con base en el vector de desplazamientos de la estructura.

9.- Los esfuerzos de cada elemento se calculan con la ecuación 2.4.11 al utilizar en ella los desplazamientos de cada elemento.

CAPITULO III ELEMENTOS QUE INTERVIENEN EN LA GRAFICACION

La graficación por computadora se usa en un sin número de aplicaciones tales como el diseño auxiliado por computadora de partes mecánicas, circuitos, vehículos, edificios y otros, así como en la simulación del rendimiento de máquinas, en la producción de mapas, en el arte, en la elaboración de caricaturas y en la verificación e interpretación de datos.

Es precisamente esta última aplicación la que se le dará a la graficación por computadora y alrededor de ésta girarán los conceptos que se mencionan en el presente capítulo.

Como se mencionó en el capítulo anterior, el M.E.F. necesita las coordenadas de los nodos que forman la malla y los nodos que forman a cada elemento finito, por lo que el sistema debe permitir verificar la captura de éstos, y una forma rápida es a través de su graficación. Con la gráfica, se tiene una visión total de la forma que toma la malla, para así detectar errores en los datos.

Finalmente se estudia el análisis de regresión utilizado en la graficación de los desplazamientos de ciertos nodos, con el objeto de visualizar gráficamente el desplazamiento de la placa.

3.1 USO DE LAS GRAFICAS.

Las actividades de la ingeniería se pueden organizar en cinco fases : diseño, análisis, dibujo, control de calidad y manufactura/construcción/procesamiento.

En la fase de diseño del proyecto de ingeniería, los ingenieros buscan alternativas que puedan proveer una solución para sus problemas. En general, ellos deben examinar cada alternativa con detalle. Esto puede involucrar dibujos rápidos para mostrar localizaciones relativas de varias partes, como sucede frecuentemente en el ensamblado. Estos datos pueden almacenarse en un archivo en la computadora y hacer los dibujos más rápidamente.

Como en el caso que ocupa el presente trabajo, el Sistema de Análisis de placas a través del M.F.F. genera la gráfica de los nodos, la de los elementos finitos y la de los desplazamientos de los nodos, útiles en la ingeniería civil.

Las dos primeras gráficas se usan con el objeto de verificar que la captura de los datos se haya realizado correctamente antes de iniciar el análisis, ya que si la información lleva errores el sistema arrojará resultados incorrectos. La tercer gráfica se usa para visualizar los desplazamientos de la placa en ciertos puntos nodales.

3.1.1 VERIFICACION DE DATOS.

En esta subsección, se persiguen dos objetivos, el primero es describir cómo se lleva a cabo la verificación de la captura de los nodos de la mallta y el segundo es explicar el procedimiento para hacer la verificación de los nodos que forman los elementos finitos.

Una vez que se capturan las coordenadas de los nodos, se procede a su graficación por medio del Sistema de Análisis de Placas. Si los datos se han capturado correctamente, entonces se desplegará en el monitor una gráfica idéntica a la que se tiene en papel, donde se dibujó la placa. En caso contrario, es posible identificar rápidamente el nodo o nodos que no están en la posición correcta, como lo muestra la figura 5.9. Debido a que el sistema es interactivo, se pueden corregir inmediatamente las coordenadas erróneas y proceder a graficar nuevamente. Dichos pasos se repiten, hasta obtener la mallta correcta de la placa.

Finalmente, en la figura 5.3 se muestra la gráfica para verificar la captura de los nodos de la placa de geometría regular que se analizará en el capítulo cinco.

Para alcanzar el segundo objetivo propuesto, se capturan los nodos que forman a cada elemento finito y se efectúa su graficación auxiliándose del sistema. En la figura 5.4 se puede

observar que los datos fueron capturados correctamente, ya que los elementos finitos están bien graficados. Si existiera error en la captura de los datos, entonces se apreciaría en la gráfica que los nodos no están formando rectángulos perfectos, sino cualquier figura irregular. En cuyo caso nos daríamos cuenta de inmediato que incurrimos en errores en la captura de los nodos, como se aprecia en la figura 5.10.

Debe corregirse y volver a graficar hasta obtener la gráfica correcta de los elementos finitos.

3.1.2 INTERPRETACION DE LOS RESULTADOS.

La gráfica de los resultados nos ayuda a tener una idea del desplazamiento que sufre la placa, después de haberle aplicado determinadas cargas. Un ejemplo de esta gráfica lo proporciona la figura 5.5, en donde se grafican los desplazamientos de los nodos que se encuentran en la parte media de la placa.

3.2 CONCEPTOS GENERALES DE GRAFICACION.

Los sistemas gráficos por computadora se componen de tres elementos, a saber :

1. *Hardware*.- Son los dispositivos físicos (chips, circuitos integrados, impresoras, monitores, etc.).
2. *Software*.- Son los programas de computadora que permiten al sistema ser usado eficientemente en aplicaciones gráficas.
3. *Usuario*.- Este elemento es el que emplea tanto el *hardware* como el *software* para la generación de gráficas.

Los algoritmos básicos para generar gráficas, en la terminología de graficación, reciben el nombre de "Primitivas de Salida", las cuales tienen parámetros que indican sus características. Estos incluyen especificaciones de color, de intensidad, estilos de texto y modelos de llenado de áreas.

La resolución que tiene una pantalla, es el número de puntos (pixels) que pueden graficarse horizontal y verticalmente.

Las instrucciones del lenguaje C, usadas en la graficación de datos en el sistema propuesto, incluyen primitivas de salida e instrucciones más elaboradas, mismas que se mencionarán en la

sección 3.5 donde se incluye la descripción de su funcionamiento y sus parámetros.

A continuación se explica el manejo de memoria que hace la computadora cuando se encuentra en modo texto y modo gráfico.

Las consideraciones de memoria se hacen teniendo en cuenta que se está trabajando con una microcomputadora compatible con IBM y con un adaptador color/gráficos, el cual es un dispositivo que actúa para manipular imágenes gráficas a color en la pantalla. Por tanto lo que a continuación se menciona es válido sólo para máquinas que cumplen estas características.

3.3 GRAFICAS EN MODO TEXTO.

Las imágenes gráficas en la computadora personal se crean con puntos controlados a través de un formato de despliegue conocido como mapeo de memoria. Con el mapeo de la pantalla se puede leer y escribir datos directamente, tan sólo con acceder a la memoria de desplegado que reside en la memoria temporal (buffer) de 16k bytes del adaptador color/gráficos de la microcomputadora.

3.3.1 REQUERIMIENTO DE MEMORIA.

A cada caracter de la pantalla en modo texto, se le asignan dos bytes en un área especial de memoria llamada memoria temporal (buffer) de desplegado. El circuito integrado que contiene esta memoria está físicamente en el adaptador color/gráficos; por lo que el adaptador actúa como una extensión de 16k bytes de la memoria principal del sistema. De esta manera, la posición de un caracter, se define con el formato caracter/atributos de dos bytes.

El byte que contiene el código del caracter tiene 8 bits y puede almacenar hasta 256 valores diferentes, es decir, puede representar 256 caracteres. El código ASCII es el más común para representar caracteres.

Dentro del byte que contiene los atributos, los 4 bits menos significativos (0,1,2 y 3) determinan el color del caracter, por lo que sólo se pueden representar 16 colores, los bits 4, 5 y 6 determinan el color del fondo, que es el área rectangular que rodea al caracter, debido a que son 3 bits se pueden representar sólo 8 colores. El bit 7 determina si el caracter parpadea o no.

Siguiendo esta filosofía se puede cambiar el valor de los dos bytes de cualquier posición de la pantalla.

3.3.2 MAPEO DE MEMORIA.

Debido a la notación y aritmética de la computadora, los datos se numeran comenzando a partir de 0, a esto se le llama el origen cero o base. Para las graficas de color, el almacenamiento comienza en 0HB800. La localidad relativa de esta base es referida como desplazamiento (offset).

Los dos primeros bytes de la memoria temporal de desplegado se localizan en la esquina superior izquierda de la pantalla (la primera posición). Los dos siguientes bytes se encuentran en la siguiente posición, justo a la derecha de los primeros. El proceso continúa hasta el fin del renglón, renglón por renglón hasta el fin de la pantalla.

3.4 GRAFICAS EN MODO GRAFICO.

3.4.1 REQUERIMIENTO DE MEMORIA.

Para indicar si un punto en la pantalla está encendido o apagado basta con un bit para controlar esto. En gráficas de alta resolución, hay $640 \times 200 = 128,000$ puntos, entonces se necesitan 128,000 bits para controlar los puntos, si cada byte tiene 8 bits, entonces se requieren 16,000 bytes por pantalla. Esto es casi la capacidad total de memoria del adaptador color/gráficos el cual tiene $16K = 16,384$ bytes; como sólo quedan sin usar 384 bytes no hay forma de asignarles color a los puntos en gráficas de alta resolución.

En la resolución media, sin embargo, hay sólo la mitad de puntos en la pantalla, por lo que se pueden dedicar 2 bits por punto para indicar el color, con los cuales se pueden especificar 4 valores diferentes, teniendo sólo 4 colores.

3.4.2 MAPEO DE MEMORIA.

En modo gráfico, la memoria temporal de despliegue está organizada en 2 bloques de 8 K bytes cada uno. Las líneas pares se almacenan en los primeros 8 K bytes y las impares en los siguientes. Esto se debe a que el adaptador primero despliega las líneas pares y después las impares.

3.5 INSTRUCCIONES DE GRAFICACION PROPIAS DEL LENGUAJE C.

A continuación se explican las instrucciones del lenguaje C usadas en la elaboración del Sistema para el Análisis de Placas por medio del M.E.F., con el fin de tener una idea más clara de lo que hacen sus rutinas de graficación.

bar(*izquierda, arriba, derecha, abajo*)

Dibuja una barra en dos dimensiones cuyas coordenadas se especifican con sus parámetros y éstos se refieren a puntos (pixels).

closegraph

Restaura la pantalla al modo en el que estaba antes de que se usara *initgraph* y libera la memoria asignada por el sistema de gráficas tal como manejadores gráficos, tipo de letra y memoria interna (buffer).

detectgraph(*manejador, modo*)

Mediante la verificación del *hardware*, determina el adaptador que se tiene en la computadora personal y elige el modo gráfico que provee la resolución más alta para ese adaptador. Dicha instrucción tiene dos parámetros, el primero indica el manejador detectado (CGA, VGA, etc) y el segundo indica el modo gráfico detectado.

getinaxcolor

Indica el número máximo de colores que se pueden tener para el manejador y modo gráfico que se tenga.

getmaxx

Indica el valor máximo sobre las abscisas de la pantalla para el modo gráfico con el que se está trabajando.

getmaxy

Indica el valor máximo sobre las ordenadas de la pantalla para el modo gráfico con el que se está trabajando.

gettext(izquierda, arriba, derecha, abajo, destino)

Almacena el contenido de una parte rectangular de la pantalla en modo texto, en el parámetro *destino*. Todas las coordenadas representan posiciones absolutas de la pantalla y no son relativas a una ventana. Así, la coordenada de la esquina superior izquierda es (1,1). Cada caracter en la pantalla ocupa dos bytes de memoria, el primer byte es el caracter y el segundo es el atributo de video como se explicó en la sección 3.3 de este capítulo. El espacio requerido para almacenar un rectángulo de *w* columnas por *h* renglones se define como:

$$\text{No. de bytes} = h \text{ (renglones)} \times w \text{ (columnas)} \times 2 \text{ (bytes)}$$

graphresult

Indica la clave del error que se cometió en la última operación gráfica, para poderlo identificar.

initgraph(manejador, modo, ruta)

Inicia el modo gráfico, por lo que carga del disco el manejador gráfico y pone al sistema en el modo gráfico indicados por los parámetros.

line(x1, y1, x2, y2)

Despliega una línea definida por las coordenadas (x1, y1) y (x2, y2).

outtext(x, y, texto)

Despliega un texto en la pantalla en la posición dada por (x,y).

putpixel(x, y, color)

Despliega un punto, con un color definido por el tercer parámetro, en la posición x, y.

puttext(izquierda, arriba, derecha, abajo, texto)

Escribe el contenido del parámetro texto en la pantalla, en el rectángulo definido por los cuatro primeros parámetros. Todas las coordenadas son absolutas y se refieren a la pantalla en modo texto y no están relacionadas con una ventana.

setcolor

Esta instrucción cambia el color deseado, indicándolo con los parámetros.

setfillstyle(patrón, color)

Esta instrucción regula, mediante los parámetros, la forma y color de los caracteres usados en modo gráfico.

settextjustify(horizontal, vertical)

El texto que se imprima en la pantalla después de haber invocado esta instrucción, será alineado a la derecha, a la izquierda, arriba, abajo o al centro, según se haya especificado en los parámetros.

setusercharsize(multx, divx, multy, divy)

Controla el tamaño de los caracteres en modo gráfico, mediante sus parámetros. Debe especificarse los factores para modular lo ancho y lo alto. El ancho se controla con *multx* : *divx* y lo alto con *multy* : *divy*.

3.6 ANALISIS DE REGRESION.

Representación del modelo

Suponga que se desea encontrar una función que exprese la relación que existe entre una variable dependiente y una variable independiente. Dada la función tabular definida por el cuadro 3.1, se trata de obtener el valor de los coeficientes de la función :

$$y = f(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_m x^m \quad (3.6.1)$$

cuya gráfica es una curva suave que se acerca a la mayoría de los puntos.

Se llama **residuo** a la diferencia de ordenadas de la curva para $X = X_i$ menos la del punto (X_i, Y_i) . Representado con R_i a este residuo, se tiene

$$R_i = f(X_i) - Y_i \quad (3.6.2)$$

es decir

$$R_i = a_0 + a_1 X_i + a_2 X_i^2 + \dots + a_m X_i^m - Y_i \quad (3.6.3)$$

en donde

$$i = 1, 2, 3, \dots, n$$

Estimación de los parámetros del modelo

El método de los mínimos cuadrados consiste en determinar los valores de los parámetros

$$a_0, a_1, a_2, \dots, a_m$$

de manera que haga mínima la suma de los cuadrados de los residuos. Esta suma vale

$$\sum_{i=1}^n R_i^2 = \sum_{i=1}^n (a_0 + a_1 X_i + a_2 X_i^2 + \dots + a_m X_i^m - Y_i)^2 \quad (3.6.4)$$

Se obtiene el mínimo de ésta, igualando a cero sus primeras derivadas parciales con respecto a todos y cada uno de los parámetros. Derivando con respecto a a_j , donde $j=0,1,2, \dots, m$, se obtiene

$$\begin{aligned} \frac{\delta}{\delta a_j} \sum_{i=1}^n R_i^2 &= \frac{\delta}{\delta a_j} \sum_{i=1}^n (a_0 + a_1 X_i + a_2 X_i^2 + \dots + a_m X_i^m - Y_i)^2 \\ &= \sum_{i=1}^n \frac{\delta}{\delta a_j} (a_0 + a_1 X_i + a_2 X_i^2 + \dots + a_m X_i^m - Y_i)^2 \\ &= \sum_{i=1}^n 2 (a_0 + a_1 X_i + a_2 X_i^2 + \dots + a_m X_i^m - Y_i) X_i^j \end{aligned}$$

Igualando a cero esta derivada, se llega a

$$a_0 \sum_{i=1}^n X_i^j + a_1 \sum_{i=1}^n X_i^{j+1} + a_2 \sum_{i=1}^n X_i^{j+2} + \dots + a_m \sum_{i=1}^n X_i^{j+m} = \sum_{i=1}^n X_i^j Y_i \quad (3.6.5)$$

Finalmente, considerando $j=0,1,2, \dots, m$, se obtiene el siguiente sistema de ecuaciones normales

$$\begin{aligned} n a_0 + a_1 \sum X + a_2 \sum X^2 + \dots + a_m \sum X^m &= \sum Y \\ a_0 \sum X + a_1 \sum X^2 + a_2 \sum X^3 + \dots + a_m \sum X^{m+1} &= \sum XY \\ a_0 \sum X^2 + a_1 \sum X^3 + a_2 \sum X^4 + \dots + a_m \sum X^{m+2} &= \sum X^2 Y \\ \dots & \\ a_0 \sum X^m + a_1 \sum X^{m+1} + a_2 \sum X^{m+2} + \dots + a_m \sum X^{m+m} &= \sum X^m Y \end{aligned}$$

(3.6.6)

en donde, por simplicidad, se han omitido los índices de X y de Y , y los límites de las sumatorias, pero debe entenderse que éstas son sobre todos los valores de X y de Y dados en el cuadro 3.1.

Resolviendo este sistema de ecuaciones se obtiene el valor de los coeficientes

$$a_0, a_1, a_2, \dots, a_m$$

para obtener la ecuación 3.6.1.

CAPITULO IV DESCRIPCION Y REQUERIMIENTOS DEL SISTEMA

La computación, que ha afectado a tan diversos campos del conocimiento humano, también lo ha hecho con la ingeniería civil. Los cambios que ha originado en esta rama, han repercutido tanto en el análisis y diseño de las estructuras como en la enseñanza de la profesión. Más que una herramienta nueva, la computación es un enfoque diferente en métodos y conceptos en ingeniería.

El uso de la computadora se hace indispensable en el diseño estructural y específicamente en el análisis de placas, estudiado en los capítulos uno y dos. Esto se debe a que la metodología de solución requiere del manejo de un gran número de ecuaciones.

Para resolver el problema de análisis de flexión de placas se desarrolló el Sistema de Análisis de Placas por medio del M.E.F. En el presente capítulo se describen sus características, se explican sus requerimientos de operación, el funcionamiento de las subrutinas y la estructura de los archivos empleados. Al final del capítulo se encuentra el Manual del Usuario en donde se detallan los pasos a seguir para usar adecuadamente el sistema.

Este manual se hizo de tal forma que aún las personas que no conozcan de computación, puedan usar adecuadamente el sistema.

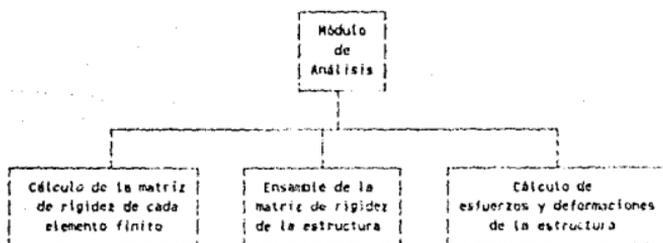
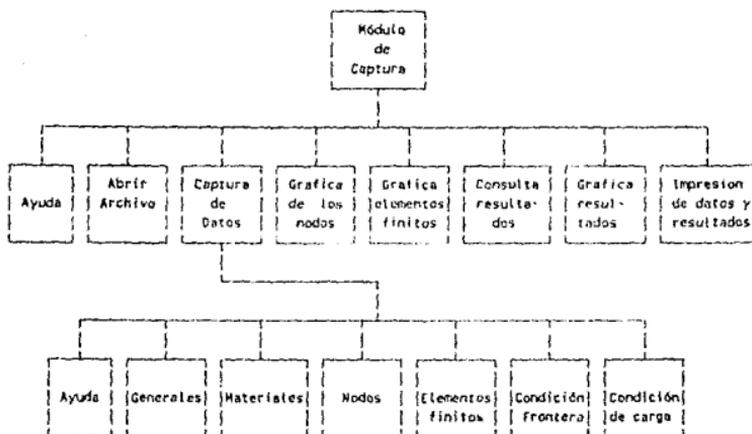
4.1 DESARROLLO DEL SISTEMA.

Un usuario se presentó con la necesidad de hacer un análisis de flexión de placas. Se procedió a investigar y estudiar la teoría de placas y a buscar procedimientos para analizar este tipo de estructuras, con el objeto de entablar una conversación con esta persona para diseñar el sistema de acuerdo a sus necesidades. Los resultados de la investigación se encuentran en el capítulo uno de esta tesis. Con el usuario se logró la comunicación gracias a los conocimientos adquiridos de matemáticas.

Posteriormente se observó que los procedimientos encontrados sólo servían para analizar placas con ciertas características, por lo que fué necesario indagar y proponer uno más general, objeto del presente trabajo. Al consultar con expertos en la materia, éstos recomendaron el uso del M.E.F. para el análisis de placas, así como explicaron el proceso de solución a través del mismo. Los resultados de la consulta fueron expuestos en el capítulo dos.

Al observar que era muy grande el número de operaciones involucradas en la resolución del método, se vió la necesidad de automatizar el procedimiento, por lo que se pensó en el diseño del sistema. Se hizo una primera versión y se presentó al usuario, éste hizo las correcciones necesarias y finalmente se obtuvo una segunda versión del sistema, cuyo esquema general se presenta a continuación.

La estructura general del sistema es como se muestra a continuación :



En el diagrama anterior, se advierte que el sistema está dividido en 2 módulos : el de *recepción de datos* y el de *análisis de placas*.

El primer módulo ejecuta la captura en forma interactiva ofreciendo facilidad y orden en la misma. Permite corregir los datos también interactivamente, graficar los nodos, los elementos finitos y los resultados y la impresión de datos y resultados.

El segundo módulo efectúa el análisis de placas y obtiene los desplazamientos y esfuerzos de la estructura. Los resultados del análisis se graban en un archivo para que se puedan consultar o imprimir, de acuerdo a las necesidades del usuario.

El hecho de que el sistema se haya estructurado en módulos separados, se justifica al considerarlo como una medida de seguridad, pues de tal forma sólo el responsable del sistema puede usar el módulo de análisis de placas quedando el módulo de recepción de datos como la parte del sistema que cubrirá el trabajo rutinario a realizar por capturistas.

Una vez identificados los datos requeridos para el procedimiento, se diseñaron seis pantallas de captura como se muestran en las figuras 4.4 a 4.9, una pantalla de captura para cada tipo de dato (esta parte se explicará más detalladamente en el Manual del usuario que aparece en el capítulo cinco). También

se decidió que la captura sería interactiva, es decir, que el sistema debería permitir agregar y corregir datos en el momento, sin tener que reiniciar el trabajo de captura.

Se decidió incluir un mecanismo en el sistema que permitiera identificar rápidamente errores en la captura, ya que al probar los algoritmos de análisis se encontró que no arrojaban los resultados esperados. Después de una revisión exhaustiva se observó que los datos no habían sido capturados correctamente.

Se adoptó a la graficación como medida para revisar los datos, ya que en forma gráfica se identifican rápidamente los errores en la captura. De esta manera se incluyeron dos opciones dentro del menú principal del sistema, una para graficar las coordenadas de los nodos y otra para graficar los elementos finitos.

Los elementos utilizados en la elaboración de las gráficas se describieron en el capítulo tres.

También se diseñaron hojas de codificación de datos para hacer más eficiente la captura.

Se utilizaron archivos de tipo secuencial debido principalmente, a que representan un ahorro considerable de espacio en disco y requieren de menos memoria para su manejo.

Esto es importante en los programas del elemento finito que ocupan gran cantidad de memoria. La desventaja del acceso secuencial es la dificultad para la localización de datos específicos, sin embargo esto no se requiere en el Sistema, por lo que la organización secuencial resultó ser idónea para este Sistema.

4.2 ORGANIZACION DEL SISTEMA.

El sistema se desarrolló en una microcomputadora con una configuración de 640 kbytes de memoria central, un manejador de disco de 5.25 pulgadas y un disco flexible con capacidad de 360 kbytes. Debido a que los resultados se despliegan en pantalla, no es necesario contar con una impresora, aunque si es posible imprimirlos.

Como se mencionó en la sección anterior el sistema se dividió en dos módulos : recepción de datos y análisis de placas.

El primer módulo se codificó en lenguaje C, por las ventajas que ofrece para el manejo de pantallas y gráficas. Este lenguaje permite dividir el programa en subrutinas completamente independientes y tener un programa principal que las enlace en forma similar a como lo hace FORTRAN. También es factible la compilación, lo cual hace más rápido al programa.

Inicialmente los algoritmos de análisis del segundo módulo se codificaron en lenguaje C, pero se encontró que no se podían declarar arreglos (vectores y matrices) de tamaño variable y tampoco se pueden pasar parámetros matriciales de una subrutina a otra. Considerando la cantidad de memoria que se ocuparía para manejar esto con arreglos de tamaño fijo en lenguaje C, entonces se decidió usar alternativamente el lenguaje FORTRAN que sí permite el paso de vectores y matrices como parámetros de una subrutina a otra sin problemas, así como permite la declaración de arreglos de longitud variable.

La desventaja principal de FORTRAN es la dificultad que presenta para el manejo de pantallas en la recepción y validación de datos y en la corrección de los mismos. Por esta razón el sistema se hizo en dos lenguajes, aprovechando las ventajas de cada uno de ellos.

La comunicación entre los dos lenguajes se hace a través de archivos. El módulo de *recepción de datos*, escrito en lenguaje C, genera la información que es utilizada por el módulo de *análisis de placas*, codificado en FORTRAN.

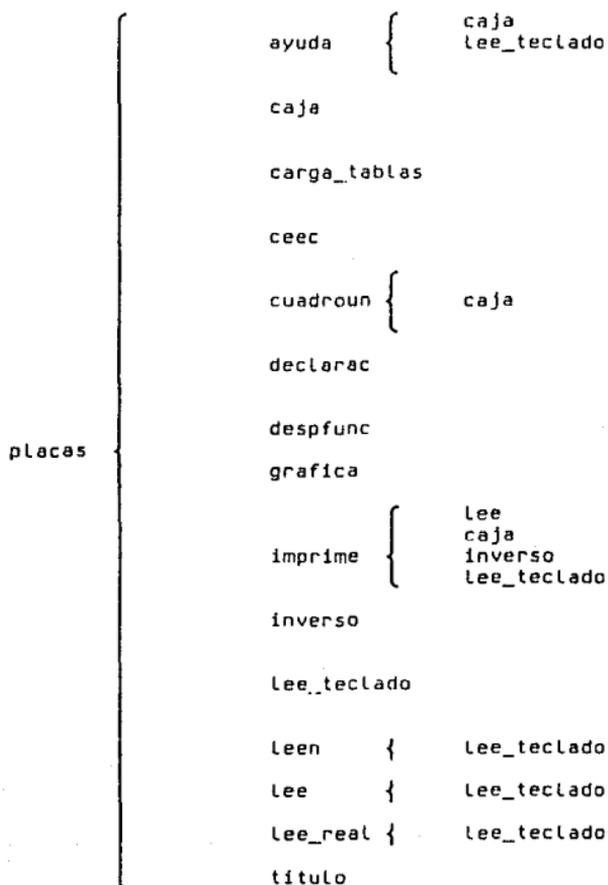
Ambos módulos tienen una estructuración tal que pueden utilizarse como un catálogo de subrutinas y con pequeños cambios sirvan para desarrollar otros sistemas que usen el M.E.F. En las siguientes secciones se citan dichas subrutinas.

4.3 MODULO DE RECEPCION DE DATOS.

Este módulo, está constituido por el programa principal y 14 subrutinas externas a él, las cuales se compilan por separado y posteriormente se encadenan. A continuación se listan las subrutinas, su funcionamiento y los parámetros que requieren :

Nombre de la Subrutina:	Descripción de la Subrutina:
<u>AYUDA</u>	Presenta la pantalla de ayuda, cuyo parámetro es el nombre del archivo de ayuda a consultar.
<u>CAJA</u>	Elabora un cuadro. Lleva como parámetros las coordenadas de las esquinas superior izquierda e inferior derecha y la cadena de caracteres para dibujar el cuadro.
<u>CARGA TABLAS</u>	Inicializa el arreglo con los mensajes de los menús, se le da como parámetro el nombre del arreglo.
<u>CEEC</u>	Convierte números a caracteres y viceversa.
<u>CUADROUN</u>	Dibuja los cuadros donde se despliegan las opciones de los menús.
<u>DECLARAC</u>	Contiene las declaraciones de tipo, estructuras y variables globales del sistema.
<u>DESPFUNC</u>	Despliega las opciones del <i>Menú principal</i> y del menú de <i>Definir datos</i> , el parámetro es 0 ó 1.
<u>GRAFICA</u>	Grafica nodos, elementos finitos y resultados.
<u>IMPRIME</u>	Imprime los datos y resultados.
<u>INVERSO</u>	Rutina que pone en color o tono diferente la opción del menú que se sugiere realizar primero, el parámetro es el número de esta opción.
<u>LEE TECLADO</u>	Lee un caracter e indica el tipo de tecla que se oprimió.
<u>LEEN</u>	Rutina que valida que sólo se lean números enteros.
<u>LEE</u>	Lee una cadena de caracteres
<u>LEE_REAL</u>	Lee números reales. Como parámetro se le da el número de enteros y decimales a leer.
<u>PLACAS</u>	Programa principal del sistema. Contiene las subrutinas de captura, consulta y graficación de datos y resultados.
<u>TIUULO</u>	Despliega la pantalla inicial con el nombre del sistema.

A continuación se presenta la organización esquemática de las subrutinas que conforman al módulo de recepción de datos :



Archivo de datos

El módulo de *recepción de datos* genera un archivo de tipo ASCII de longitud variable, cuyos datos se graban en el siguiente orden :

a) Un registro de datos generales con la siguiente descripción:

CAMPO	DESCRIPCION	LONGITUD	TIPO
NNC	Número de nodos cargados	5	entero
IFC	Fuerzas de cuerpo	1	entero
ICE	Cálculo de esfuerzos	1	entero
NODOS	Cuadrilátero o Lineal	1	entero
ESPESDR	Espesor de los materiales	7	real
GENERAL[1]	Número de puntos frontera	5	entero
GENERAL[2]	No. de condiciones de carga	5	entero
GENERAL[3]	Número de elementos finitos	5	entero
GENERAL[4]	Número de nodos	5	entero
GENERAL[5]	Número de materiales	5	entero
GENERAL[6]	Número de grados de libertad	5	entero

Los siguientes registros aparecen en el archivo tantas veces como se hayan especificado en los datos generales.

b) Datos de los materiales.

CAMPO	DESCRIPCION	LONGITUD	TIPO
MAT_NO	Número de material	5	entero
ELASTI	Elasticidad del material	9	real
POISSO	Coefficiente de poisson	5	real
PESO_V	Peso volumetrico	5	real

c) Coordenadas de los puntos nodales.

CAMPO	DESCRIPCION	LONGITUD	TIPO
CO[2]	Coordenadas X y Y de los nodos	7	real

d) Códigos frontera.

CAMPO	DESCRIPCION	LONGITUD	TIPO
NUDO	Nodo con el código frontera	5	entero
CO.X	Código frontera en el eje X	1	entero
CO.Y	Código frontera en el eje Y	1	entero
CO.Z	Código frontera en el eje Z	1	entero

e) Elementos finitos.

CAMPO	DESCRIPCION	LONGITUD	TIPO
NUM_ELEM	Número de elemento	3	entero
NUM_MAT	Numero de material del elemento	1	entero
NUDO{8}	Conjunto de nodos que forman el elemento	3	entero

f) Condiciones de carga.

CAMPO	DESCRIPCION	LONGITUD	TIPO
NUDO	Número del nodo	5	entero
FX	Condición de carga en "X"	5	real
FY	Condición de carga en "Y"	5	real
FZ	Condición de carga en "Z"	5	real

4.4 MODULO DE ANALISIS DE PLACAS.

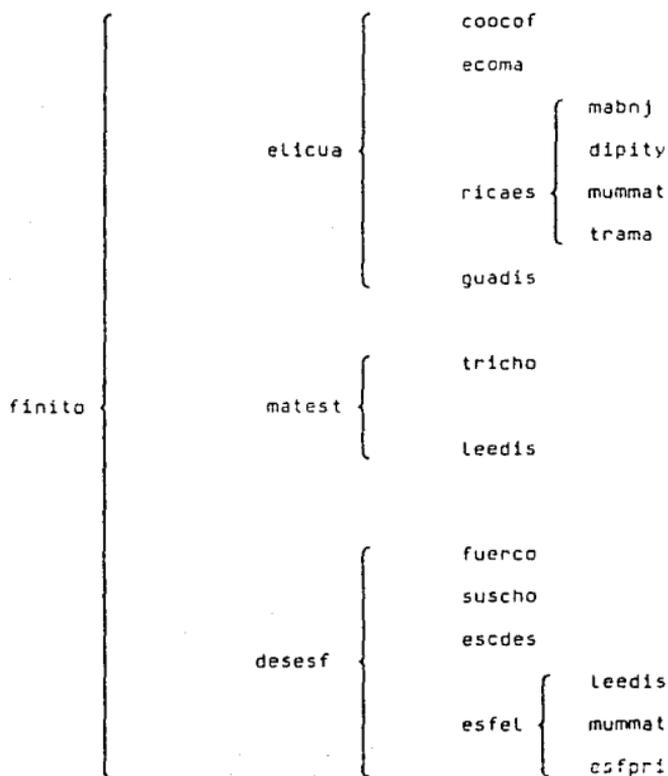
Este módulo calcula los desplazamientos y esfuerzos de la estructura. Está constituido por el programa principal y 17 subrutinas externas a él, las cuales se mencionan a continuación :

Nombre de la Subrutina:	Descripción de la Subrutina:
FINITO	Programa principal.
COOCOF	Se leen las coordenadas y códigos frontera.
DESESF	Se calcula el vector de cargas, desplazamientos y esfuerzos de la estructura.
DIPITY	Calcula las funciones de forma cuadráticas.
ECOMA	Forma la matriz D de coeficientes elásticos para el caso del estado plano de esfuerzos.
ESCDES	Imprime los desplazamientos.
ESFEL	Calcula los esfuerzos.
ESFPRI	Calcula los esfuerzos principales.
FUERCO	Calcula las cargas concentradas.
MABNJ	Calcula la matriz B y el jacobiano para funciones de forma lineales.
MATEST	Forma la matriz de la estructura.
MUMMAT	Realiza la multiplicación de 2 matrices. Los parámetros son las dos matrices a multiplicar, la matriz de resultado y la dimensión de estas.
RICAES	Ensambla la matriz de rigideces de la estructura.
SUSCHO	Realiza la sustitución hacia atrás por el Metodo de Cholesky.
TRAMA	Calcula la transpuesta de una matriz.
TRICHO	Ejecuta la triangulación por el metodo de Cholesky.

Archivo de resultados.

Este archivo contiene los resultados obtenidos después de aplicar el método y es de tipo texto. Sus líneas son de 80 caracteres de longitud y contiene los desplazamientos y esfuerzos de los nodos de la estructura.

A continuación se presenta la organización esquemática de las subrutinas que conforman al módulo de análisis de placas :



4.5 MANUAL DEL USUARIO.

En esta sección se mencionan los datos que se requieren para ejecutar los módulos descritos en las secciones anteriores y posteriormente se indican las instrucciones de cómo se operan cada uno de ellos.

Se cuenta con hojas de codificación especiales para cada tipo de dato, con el objeto de facilitar su captura. Estas se encuentran en el apéndice B en los cuadros 4.1 a 4.6 y se presentan sin datos para tener la posibilidad de fotocopiarlas y utilizarlas cuando se necesiten. La información requerida es :

1.- Datos generales de la placa:

- a) Número de elementos de toda la estructura.
- b) Número total de nodos que tiene la estructura.
- c) El número de nodos que están bajo la acción de una carga.
- d) Número de materiales, es decir, cuántos materiales con propiedades distintas forman parte de la estructura.
- e) Espesor de la placa en metros.
- f) Se considera (1) o no (0) el peso de la estructura.
- g) Número de nodos restringidos, aquí debe indicarse el número de nodos que no estén libres en los ejes cartesianos X, Y o Z o en cualquier combinación de éstos.
- h) Cuadrilátero o Lineal (C/L). EL presente sistema utiliza funciones de forma de la familia serendipity para

aproximar la solución. Para ello considera elementos finitos formados de cuatro nodos llamados cuadriláteros lineales (L) o alternativamente puede usar elementos finitos de 8 nodos, conocidos como cuadriláteros cuadráticos (C). Debe especificar según el caso.

Para la recopilación de estos datos se cuenta con la hoja de codificación que se muestra en el cuadro 4.1.

2.- Datos de los materiales de la placa:

- a) Número de cada material con características distintas al resto de los materiales que se utilizan.
- b) Módulo de elasticidad, en toneladas por metro cuadrado, para cada material.
- c) El coeficiente de poisson, adimensional, para cada material.
- d) Peso volumétrico, en toneladas por metro cúbico, de cada material.

Debe llenarse la hoja de codificación de datos del cuadro 4.2 para facilitar la captura de los datos antes mencionados.

3.- Datos para las coordenadas de los nodos:

- a) Número de nodo
- b) La abscisa X del nodo.
- c) La ordenada Y del nodo.

Para las coordenadas de los nodos se tienen dos opciones:

La primera opción consiste en capturar las coordenadas de los

nodos que ni pueden generar coordenadas de otros ni ellos mismos pueden ser generados por otro nodo.

La segunda es usar la generación automática de coordenadas, para los nodos que sí puedan generar coordenadas de otros.

Para que un nodo pueda generar las coordenadas de otros, deben tener entre sí una diferencia constante en X, en Y y en la numeración de los nodos. El cuadro 4.3 es la hoja de codificación.

4.- Datos de los Elementos Finitos:

- a) Número del elemento finito.
- b) Los ocho puntos nodales que forman al elemento.
- c) Número de material del elemento, este número debe existir en la tabla de los *Datos de los materiales*.

Los datos de los elementos finitos se vacian en la hoja de codificación que muestra el cuadro 4.4. Los nodos se numeran siguiendo el orden que muestra la figura S.2.B.

5.- Datos de los códigos frontera en los nodos:

- a) El número del nodo que está restringido.
- b) Código frontera en X
- c) Código frontera en Y
- d) Código frontera en Z

Para estos tres últimos datos debe teclearse 1 si el nodo está restringido en el eje o se teclea 0 si está libre. Para recopilar estos datos se usa la hoja de codificación que se observa en el cuadro 4.5.

6.- Datos de los nodos cargados:

- a) Número de nodo al que corresponden las cargas.
- b) La carga en toneladas sobre el eje X.
- c) La carga en toneladas sobre el eje Y.
- d) La carga en toneladas sobre el eje Z.

Los datos de los nodos cargados se vacían en la hoja que se muestra en el cuadro 4.6.

Una vez que se tienen definidos todos los datos, se introducen a la computadora mediante el módulo de *recepción de datos*, que permite la captura interactiva de los mismos.

Posteriormente debe ejecutarse el módulo de *análisis de placas* para obtener los desplazamientos y esfuerzos de la estructura. Este se encuentra separado de la captura para controlar el uso del proceso. Es decir, un capturista puede dedicarse a teclear los datos, mientras que el proceso de cálculo, sólo estará accesible para el responsable del sistema.

Al término de la ejecución del módulo de análisis de placas, se pueden consultar los resultados por pantalla, graficarlos o imprimirlos, para lo cual se deberá usar nuevamente el módulo de *recepción de datos*.

En las siguientes dos secciones se describe detalladamente el manejo de los dos módulos.

4.5.1 INSTRUCTIVO DE OPERACION DEL MODULO DE RECEPCION DE DATOS.

Este módulo realiza la captura de datos, la graficación de nodos, elementos finitos y resultados e imprime los resultados.

Nota técnica : Cada vez que se ejecute el sistema, el usuario deberá cercionarse que existan los archivos: PLACAS.EXE, AYUDA1.TXT y AYUDA2.TXT en el subdirectorio o disco flexible donde se va a correr el sistema.

Todas las figuras que se mencionan en esta sección se encuentran en el apéndice A, y se refieren a los desplegados en el monitor de la microcomputadora. Para indicar que debe usarse una tecla, ésta se pondrá dentro de los símbolos ().

Para ejecutar el módulo teclee :

PLACAS (ENTER)

y enseguida aparecerá la pantalla con el nombre de la tesis. Oprima cualquier tecla para continuar. Posteriormente se despliega el menú principal como se muestra en la figura 4.1.

Fr Ayuda.

Al presionar <F1>, se obtiene una breve descripción de las opciones del menú inicial. Como se puede apreciar en la figura 4.3, se tienen 2 opciones : oprimir <PgUp> para pasar a la siguiente página o <F10> para regresar al menu principal.

F2 Abrir archivos.

Al inicio del programa, esta opción siempre se encuentra en color o tono distinto con respecto al resto de las opciones, con el objeto de sugerir que como primer paso se abra el archivo de datos. Esta opción permite definir el nombre y abrir el archivo donde se grabarán los datos capturados. Se oprime <F2> y se digita el nombre del archivo de datos que se desea usar o crear. Si el archivo ya existe se despliega el mensaje "Abriendo archivo", pero si no existe, entonces se crea, y aparece el mensaje "Se crea el archivo. Oprima <ENTER> para continuar". Ahora se tiene listo el archivo de datos para trabajar.

F3 Definir datos.

Esta opción permite capturar los datos que se listaron en la sección 4.4, como se muestra en la figura 4.2. En todas las pantallas de captura que se mencionan en esta figura, se puede avanzar a la siguiente columna si se presiona <ENTER>. Para pasar a los siguientes renglones oprima (), para subir uno o más renglones oprima (), para avanzar una página arriba con <PgUp> y <PgDn> para avanzar una página hacia abajo. Para salir de cualquier pantalla de captura hacia el menú de *Definir datos* oprima <F10>. A continuación se describen las opciones de este submenú :

Ayuda

Se despliega una breve descripción de las opciones del submenú de *Definir datos*, como se puede observar en la Figura 4.10.

Generales

Con esta opción se definen los datos generales de la estructura, los cuales se listaron en la sección anterior. La figura 4.4 muestra la máscara de captura de éstos.

Características de los materiales

En esta opción se capturan los materiales de que está hecha la placa. Para cada material se captura: su número de 5 dígitos, el módulo de elasticidad de 7 enteros y 2 decimales, el coeficiente de poisson de 1 entero y 4 decimales y el peso volumétrico de 1 entero y 4 decimales. Se pueden capturar tantos materiales como se hayan especificado en la pantalla de datos generales. El sistema no permitirá introducir más materiales que los especificados por el usuario. La figura 4.5 muestra el formato de captura.

Coordenadas de nodos

Como se muestra en la figura 4.6, en esta opción se capturan las coordenadas de los nodos (de 5 enteros y 2 decimales) que forman la malla de la placa, el número de nodos ya fue especificado en los datos generales. Al presionar <F1> se permite la generación automática de coordenadas de los nodos, para esto, se debe dar el punto inicial X, el punto inicial Y, el incremento en X y Y, el nodo inicial, el incremento en el nodo y el número de nodos. Para salir de la generación automática se presiona <F10> y enseguida se despliegan las coordenadas generadas.

Elementos

Esta opción permite capturar, para cada elemento finito definido en la pantalla de datos generales, el número del elemento de 5 dígitos, el número de material de 1 dígito y los números de los nodos que forman el elemento, cada número de nodo es de 3 enteros. La mascarilla de captura se encuentra en la figura 4.7.

Condiciones frontera

Se capturan los nodos que tienen condiciones frontera y sus códigos. En la figura 4.8 se observa que es necesario teclear el número del nodo de 5 dígitos y el código frontera de X, Y y Z que sólo puede ser 0 ó 1.

Condiciones de carga

Para cada nodo con condiciones de carga se captura el número del nodo de 5 dígitos, la carga en X, Y y Z, los tres de 2 enteros y 3 decimales. La pantalla de captura se muestra en la fig. 4.9.

F4 Gráfica de los nodos.

Al oprimir <F4> se obtiene la gráfica de la malla. Esta se hace con el objeto de identificar errores en la captura de las coordenadas de los nodos. Para graficar la malla, se debe indicar el nombre del archivo que contiene los datos. Un ejemplo de esta gráfica se tiene en la figura 5.3.

F5 Gráfica de los elementos finitos.

Con esta opción se grafican los elementos finitos para verificar que en la captura, se hayan indicado los nodos correctos para cada elemento. En la figura 5.4 se muestra un ejemplo.

F6 Consulta de resultados.

Se obtiene un desplegado por pantalla de los desplazamientos y esfuerzos de los nodos de la placa, como se muestra en la figura 4.11. Para hacer la consulta, el sistema solicita el nombre del archivo que se generó con el módulo de *análisis de placas*, el cual contiene los datos y resultados.

F7 Gráfica de resultados.

Se grafican los desplazamientos de los nodos. Para elaborar la gráfica debe teclear el nombre del archivo donde se encuentran los resultados y los números de los nodos cuyos desplazamientos se van a graficar. La figura 5.5 muestra una gráfica de este tipo.

F8 Impresión de datos y resultados.

Al elegir esta opción aparece la pantalla que se muestra en la figura 4.12. Después de verificar que la impresora esté lista se debe oprimir (ENTER) para que comience la impresión.

f9 Cerrar archivo.

Cierra el archivo de datos. Si posteriormente se desea hacer cambios a los datos, deberá abrir primero el archivo con *f2*.

Fro Fin.

Esta opción se usa para salir del sistema al DOS (Sistema Operativo en Disco).

4.5.2 INSTRUCTIVO DE OPERACION DEL MODULO DE ANALISIS DE PLACAS.

Con este módulo se obtienen los desplazamientos y esfuerzos de la estructura definida en el archivo de datos, generado por el módulo de *recepción de datos*.

Para iniciar el programa digite lo siguiente:

FINITO (ENTER)

enseguida el módulo solicita el nombre del archivo que contiene los datos. Después pide el nombre del archivo en el cual se grabarán los resultados, finalmente, solicita el nombre del archivo de trabajo. Es importante que primero verifique que los últimos dos archivos no existan, ya que de existir alguno de ellos, se detendrá la ejecución del módulo.

Teclee el nombre de los archivos y su extensión (si tienen) y oprima (ENTER). A continuación aparece el siguiente mensaje :

"En proceso ...".

Debe esperar aproximadamente 15 minutos para que termine el proceso.

Una vez que el programa termina, los resultados se pueden consultar, imprimir y graficar con el módulo de *recepción de datos*.

CAPITULO V EJEMPLOS DE APLICACION

Para ilustrar el procedimiento del M.E.F. en el análisis de placas que se ha presentado en este trabajo, se muestran dos ejemplos de aplicación. El primero es el análisis de una placa de geometría regular y el segundo es el de una placa de geometría irregular.

Las figuras y cuadros que se mencionan en este capítulo se encuentran en los apéndices A y B respectivamente.

5.1 ANALISIS DE UNA PLACA DE CONCRETO DE GEOMETRIA REGULAR, CARACTERISTICAS DEL PROBLEMA.

La placa analizada, que se muestra en la figura 5.1.A, tiene las siguientes dimensiones : 4 metros de ancho por 6 metros de largo y 9 cm de espesor. Se encuentra empotrada en sus cuatro lados y bajo la acción de una carga uniformemente distribuida de 5 ton/m². Es de concreto con un módulo de elasticidad de 2,220,000 ton/m² y coeficiente de poisson de 0.2.

5.2 DISCRETIZACION DE LA PLACA.

Para analizar la placa con el M.E.F. se hizo una malla de 6 elementos finitos y 29 puntos nodales, a lo cual se conoce como discretización. La malla resultante se muestra en la figura 5.2.A.

Para organizar la captura de los datos requeridos por el módulo de *recepción de datos*, se toman de apoyo los cuadros 4.1 a 4.6. A continuación se explica cómo se llenan estas hojas de codificación de datos.

El cuadro 5.1 corresponde a los datos generales de la placa. Estos se describen a continuación: en la figura 5.2.A se tiene que los elementos finitos son 6 y el número de nodos 29. La placa está empotrada en sus cuatro lados y como se observa en la figura 5.1.B tiene 20 nodos restringidos, los cuales se localizan en el contorno de la placa. Como la carga es distribuida, se aplica en todos los nodos excepto en los restringidos, por lo tanto, el número de nodos cargados es de 9. La cantidad de materiales de que está formada la placa es de uno que es el concreto.

Como se mencionó anteriormente, el espesor de la placa es 9 cm; como un requerimiento del sistema, este valor debe darse en metros, así que se indica como 0.09. Para indicar las fuerzas de

cuerpo, se escribe 1 si se consideran y 0 si no se consideran. En este caso sí se consideran las fuerzas de cuerpo.

El cuadro 5.2 muestra las características de los materiales de que está hecha la placa. En el ejemplo que nos ocupa, la placa está constituida por un solo material, que es concreto, con el módulo de elasticidad, el coeficiente de Poisson y el peso volumétrico que se indican en dicho cuadro.

En el cuadro 5.3 se indican las coordenadas X y Y para cada nodo que aparece en la figura 5.2.A. Se considera que el eje "X" es positivo hacia la derecha y el eje "Y" lo es hacia arriba. El origen de las coordenadas se encuentra en la parte inferior izquierda, o sea en el nodo 19, el cual tiene coordenadas $X=0$ y $Y=0$, por lo tanto, las coordenadas del nodo 1 son $X=6$ y $Y=4$, y así sucesivamente.

En el cuadro 5.4 se indica la numeración que le corresponde a los nodos de cada elemento finito, así como el tipo de material de que está hecho. El número de cada elemento finito está en la figura 5.2.A. De esta manera, la numeración de los nodos del elemento finito uno es: 1, 6, 3, 10, 11, 7, 3, 2. Esta numeración está de acuerdo al orden que muestran los nodos en la figura 5.2.B. El número del material es el 1 que corresponde al concreto según el cuadro 5.2.

El cuadro 5.5 contiene los códigos frontera en los nodos de la placa. La placa tiene 20 nodos restringidos con empotramientos y se encuentran localizados en el contorno de la placa. En la figura 5.2.A, se observa que los nodos restringidos son el 1, 2, 3, 4, 5, 6, 8, 9, 13, 14, 16, 17, 21, 22, 24, 25, 26, 27, 28 y 29 y por estar empotrados tienen restricción en X, Y, y Z. Los códigos de frontera se especifican con 1 si están restringidos y con 0 si están libres. Los nodos que no se tomen en cuenta se consideran libres.

Las cargas de la placa se encuentran en el cuadro 5.6, en la cual se observa que en cada uno de los nodos 7, 10, 12, 15, 18, 20 y 23 actúa una carga de 15.484 toneladas en la dirección del desplazamiento de la placa y en los nodos 11 y 19 actúa una carga de 7.742 toneladas en cada uno, en dirección contraria al desplazamiento de la placa. La forma de determinar las cargas que deben aplicarse en cada uno de los nodos y su dirección se explican a continuación.

En la placa, la carga de 5 ton/m^2 uniformemente distribuida, se sustituyó con cargas concentradas actuantes en los nodos, dichas cargas se colocaron en los nodos 7, 10, 11, 12, 15, 18, 19, 20 y 23. La carga de 5 ton/m^2 aplicada en los 24 m^2 de la placa hacen un total de 120 toneladas que se distribuyen en los nodos indicados.

En la práctica, al analizar placas a través del sistema propuesto, se notó que, para dividir la carga uniformemente distribuida en una placa entre el número de nodos de la malla utilizada, se ignoran los nodos que están en las fronteras y sólo se toman los nodos internos de esquina e intermedios. Para esta división se considera que a los nodos de esquina les corresponde el 37.5% de un nodo intermedio. La carga para cada nodo se aplica a los nodos internos intermedios y a los internos de esquina se les aplica la mitad de la correspondiente a los internos intermedios. Esto fue probado empíricamente.

Por lo tanto, la carga de 120 toneladas se divide entre 7.75, donde 7.75 es el número de nodos; esto es, 7 nodos intermedios y 2 nodos vértices. El resultado de la división indica que a cada uno debe aplicársele 15.484 toneladas.

En cada uno de los nodos intermedios actúan 15.484 toneladas de carga en la dirección del desplazamiento y a cada uno de los nodos vértice les corresponden 7.742 toneladas en dirección contraria al desplazamiento. Lo anterior también fue probado en la práctica.

5.3 VERIFICACION DE LOS DATOS A TRAVES DE GRAFICAS.

Intensionalmente se cometieron errores en la captura de las coordenadas de los nodos, usando el módulo de *recepción de datos*, con el fin de mostrar cómo se vería la gráfica con error. En la figura 5.9 se puede observar que los datos se teclearon incorrectamente, ya que la gráfica no es similar a la malla que muestra la figura 5.2.A. Una vez corregidos los datos, se obtiene la gráfica de la figura 5.3, que presenta la malla sin error.

De la misma forma se verifica la captura de los nodos que componen a cada elemento finito. Como se muestra en la figura 5.10 la gráfica no está formando rectángulos perfectos, sino figuras irregulares, lo que nos indica que los datos de los elementos finitos se capturaron con error. Se realizan las correcciones a través del módulo de recepción de datos, obteniéndose la gráfica correcta que muestra la figura 5.4.

5.4 COMPARACION DE RESULTADOS.

Los esfuerzos obtenidos al analizar la placa de geometría regular se encuentran en el cuadro 5.0.

Los desplazamientos se presentan en el cuadro 5.7. A continuación se comparan con el resultado obtenido con la teoría clásica de placas al utilizar la solución propuesta por Westergaard.

El desplazamiento máximo en el centro de la placa obtenido con la teoría clásica de placas es de 2.0289 cm, mientras que el desplazamiento máximo en el nodo 15 obtenido con el M.E.F. al utilizar este sistema es de 2.1047 cm. Observe que la diferencia es de 0.0758 la cual es una aproximación muy buena comparada con el resultado arrojado con la teoría clásica, ya que se tiene un porcentaje de aproximación del 96.6%.

La figura 5.5 contiene la gráfica de los desplazamientos de los nodos que se encuentran en el centro de la malla. Consulte el Manual del Usuario que aparece en el capítulo anterior, para obtener mayor detalle de los pasos a seguir para generar las 3 últimas gráficas.

5.5 PLACA DE CONCRETO, DE GEOMETRIA IRREGULAR.

La geometría de la placa analizada se muestra en la figura 5.6. Esta placa es de concreto, empotrada en un lado y libre en los demás y se encuentra bajo una carga de 0.9 ton/m^2 . La placa tiene un módulo de elasticidad de $2,220,000 \text{ ton/m}^2$ y un coeficiente de poisson de 0.2. La malla de 6 elementos finitos y 31 puntos nodales se presenta en la figura 5.7.

Las cargas que se aplicaron a cada nodo se muestran en la figura 5.8, éstas se definieron con el mismo criterio empleado en el ejemplo de la placa regular.

Los resultados del análisis se encuentran en los cuadros 5.9 (desplazamientos) y 5.10 (esfuerzos).

CONCLUSIONES

Conclusiones particulares :

Después de haber diseñado, utilizado y probado el Sistema de Análisis de Placas por medio del Método del Elemento finito con varios ejemplos se desprenden las siguientes conclusiones en orden jerárquico :

a) La captura interactiva de datos resultó una herramienta importante y amigable en la corrección de los mismos.

b) La graficación de los nodos y de los elementos finitos permiten verificar que los datos se hayan capturado correctamente.

c) La graficación de los resultados da una idea precisa del desplazamiento de los nodos en la parte media de la placa.

d) La generación de nodos de manera automática hace más eficiente la entrada de datos porque no tienen que capturarse todos. Esto evita errores y hace más rápida la captura.

e) El haber diseñado el sistema de manera modular tiene dos ventajas : lo hace más eficiente ya que el catálogo de subrutinas

puede ser usado por otros sistemas y por otro lado, resulta ser una medida de seguridad, cuando hay un usuario que captura y un responsable del sistema, en donde es importante que el primero no tenga acceso al sistema que sólo opera el segundo. Esta organización también favoreció el uso de dos lenguajes, cada uno apropiado para el módulo en que se usó.

f) Es benéfico emplear memoria dinámica y archivos secuenciales, pues disminuye la cantidad de memoria utilizada, por lo que no es indispensable el uso de máquinas con mucha memoria.

g) La ventaja de usar archivos de datos y resultados con nombre variable, es que permiten ejecutar el sistema varias veces, sin que los datos se pierdan en cada ejecución y sin que los resultados se tengan que imprimir inmediatamente o copiar de la pantalla.

h) Se recomienda como proceso general para el desarrollo de sistemas de ingeniería los pasos que se mencionan a continuación :

- Estudiar la teoría de lo que tratará el sistema.
- Entrevistar al usuario para que indique sus requerimientos.
- Presentarle una primera versión del sistema, para que el usuario indique las modificaciones pertinentes.
- finalmente presentar la segunda versión del sistema.

i) Existe un ahorro de tiempo al tener el sistema en versión ejecutable.

En base en el análisis de diferentes tipos de placas se presentan las conclusiones que a continuación se describen, también en orden jerárquico :

a) Si se utilizan elementos finitos de dos metros por dos metros, se obtienen resultados con aproximadamente 3% de error, el cual puede considerarse aceptable.

b) Mientras más pequeño es el elemento finito, se obtiene mayor exactitud en los resultados. Sin embargo el problema se complica porque aumenta el número de datos de entrada (y con ello la posibilidad de error en su captura), incrementa el tiempo en la ejecución del programa de cálculo y aumenta la cantidad de memoria necesaria para su solución, lo que hace necesario el uso de una computadora con mayor capacidad de memoria.

c) Si se utilizan elementos finitos de un metro por un metro, los resultados presentan un error de aproximadamente 1%, pero si la placa no es pequeña se presentan las complicaciones del punto inmediato anterior.

d) Al analizar placas formadas por elementos finitos cuadriláteros lineales y cuadráticos, se evaluaron los resultados de ambos, observándose que para obtener resultados similares con los dos tipos de elementos finitos, la malla de lineales debe estar formada por cuatro veces más elementos que la malla de cuadráticos, POR LO QUE SE CONCLUYE que deben utilizarse los

elementos finitos cuadriláteros cuadráticos, pues son los que dan la mejor aproximación a la solución.

e) Por pequeña que sea la placa, la malla debe estar formada por no menos de cuatro elementos finitos, ya que a partir de ahí se obtiene una buena aproximación.

f) En la práctica, al analizar placas a través del sistema propuesto, se notó que, para dividir la carga uniformemente distribuida en una placa entre el número de nodos de la malla utilizada, se ignoran los nodos que están en las fronteras y sólo se toman los nodos internos de esquina e intermedios. Para esta división se considera que a los nodos de esquina les corresponde el 37.5% de un nodo intermedio. La carga para cada nodo se aplica a los nodos internos intermedios y a los internos de esquina se les aplica la mitad de la correspondiente a los internos intermedios. Esto fue probado empíricamente.

g) Al sustituir la carga uniformemente distribuida por cargas concentradas actuantes en los nodos, las cargas aplicadas en nodos intermedios se colocan en dirección del desplazamiento de la placa y las que estén en nodos de esquina van en dirección contraria al desplazamiento para contribuir con la continuidad de la deformación de la placa y obtener resultados correctos. Los únicos nodos que no se cargan son los restringidos.

Conclusión general :

Al analizar los dos grupos de conclusiones particulares, se desprende en general que el análisis de placas resultó factible a través del sistema propuesto en el presente trabajo, aplicado a placas de diversas geometrías, formadas por más de un material y con diferentes condiciones de apoyo con una solución propuesta por el Método del Elemento Finito.

La formación adquirida a lo largo de la carrera de Matemáticas Aplicadas y Computación permitió analizar un problema, alejado aparentemente del contexto de dicha carrera. Sin embargo, se pudo abordar, sin gran dificultad, con asesoría y a través de las herramientas adquiridas en la misma.

El diseño y desarrollo del sistema propuesto se considera como una ampliación más de la carrera antes mencionada lo cual la justifica, ya que resuelve la problemática descrita en el presente trabajo perteneciente a un área ajena a la formación académica de quien elaboró esta tesis.

GLOSARIO DE TERMINOS

Dado que parte del material usado en este trabajo es específico del campo de conocimiento de la Ingeniería Civil, a continuación se listan en orden alfabético los términos de mayor uso y se da una breve explicación de lo que significan en dicho campo :

Adimensional.- Que no tiene dimensión.

Cargas uniformes.- Cuando las cargas de fuerzas se reparten en toda la losa y la unidad de medida está dada en metros cuadrados.

Coefficiente de Poisson.- Es la razón del valor (con su signo) de la deformación lineal en dirección lateral sobre la deformación lineal en dirección axial.

Compatibilidad de deformaciones.- Una estructura (por ejemplo una placa) se deforma de la misma manera sea cual sea la dirección que se estudie, al aplicar las cargas.

Cortante.- Es una fuerza interna de la placa originada por una carga aplicada en la dirección del eje "x".

Deflexión.- Es la deformación que sufre una placa. La placa se dobla por tres fuerzas : fuerza cortante, fuerza normal y fuerza flexionante.

Discretizar.- Dividir una estructura (por ejemplo una placa) en elementos finitos.

Equilibrio.- Una placa está en equilibrio cuando las fuerzas externas que actúan sobre ella son iguales a las fuerzas internas, de ahí que se tenga que
Fuerzas externas - Fuerzas internas = 0

Esfuerzo.- Una carga repartida en un área determinada.

Estática.- Parte de la mecánica que estudia las fuerzas en equilibrio.

Fuerzas. Son fenómenos de atracción o repulsión entre los cuerpos. En dinámica se le define como masa por aceleración.

Fuerzas de cuerpo.- Es lo que pesa el cuerpo en estudio.

Material elástico.- Se dice que una estructura está hecha de material elástico cuando al aplicarle una carga, éste se deforma, pero si se le quita la carga el material regresa a su estado original.

Material Homogéneo.- Es aquel que tiene el mismo número de moléculas por metro cúbico. Un cuerpo hecho de un sólo material.

Material Isotrópico.- Es un material que tiene el mismo comportamiento si se aplica la carga en diferentes direcciones. Materiales con las mismas propiedades en todas direcciones.

Material Lineal.- Es un material que tiene una deformación proporcional.

Medios continuos.- Son cuerpos que no tienen una forma específica, es decir, tienen forma irregular.

Método del elemento finito.- Método generalizado para problemas continuos de dos y tres dimensiones como placas y cascarones, adecuado para cuerpos con propiedades no homogéneas o con una geometría compleja e irregular. En este método la estructura real continua se sustituye por una estructura idealizada equivalente compuesta de elementos discretos, conocidos como elementos finitos,

que se conectan en cierto número de nodos.

Módulo de elasticidad.- Es un coeficiente que describe la deformación del material, con éste podemos determinar el tipo de material del que se trata.

Peso volumétrico.- Es el peso por metro cúbico del material.

Placa.- Es una estructura que posee tres dimensiones, cuyo espesor es pequeño comparado con sus otras dos dimensiones.

Placa empotrada.- Cuando una placa tiene anclajes en los muros.

Placa simplemente apoyada.- Cuando la placa se recarga en un muro.

Teoría clásica de placas.- Es una teoría matemática que analiza como se deforma una placa, resuelve una ecuación diferencial por un criterio matemático, como si resolviera una ecuación diferencial normal. Esta ecuación diferencial se ha resuelto en forma aproximada por el M.E.F. y no en forma cerrada como lo hace la teoría clásica.

APENDICE A

F I G U R A S

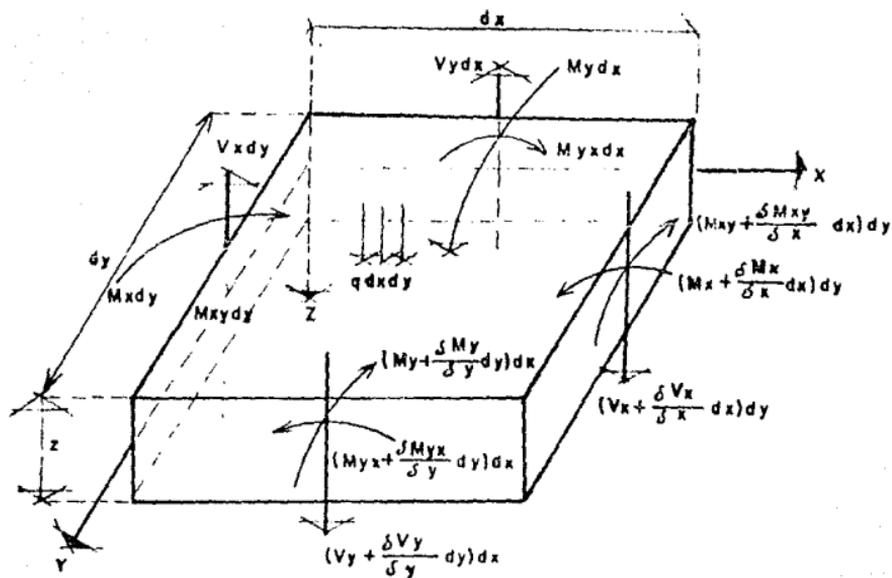


Fig. 1.1 Variaciones de las fuerzas actuantes en una placa.

P L A C A S
Análisis de Placas por el Método del Elemento Finito

Versión 1.0

F1 Ayuda	F2 Abrir Archivo	F3 Definir Datos	F4 Gráfica de los nodos	F5 Gráfica de elementos
F6 Consulta Resultados	F7 Grafica Resultados	F8 Impresión Dat.y Res.	F9 Cerrar Archivo	F10 Fin

Figura 4.1

María Guadalupe Vera Chávez

P L A C A S
Análisis de Placas por el Método del Elemento Finito

Versión 1.0

F1 Ayuda	F2 Generales	F3 Carac. de Materiales	F4 Coordenadas de nodos	F5 Elementos
F6 Condiciones Frontera	F7 Condiciones de Carga	F8	F9	F10 Fin

Figura 4.2

María Guadalupe Vera Chávez

P L A C A S
Análisis de Placas por el Método del Elemento Finito

Versión 1.0

F2 Abrir
Archivo

Permite definir el nombre del archivo del cual se leeran o al cual se grabaran los datos del proceso.

F3 Definir
Datos

Consulta o captura de los datos necesarios para correr el proceso. Estos datos son: generales, de materiales, de coordenadas, de elementos, de condiciones frontera y de carga.

F4 Grafica de
los nodos

Grafica los nudos para verificar la distribución de los mismos dentro de la malla.

PgDn -> Siguiente página

F10 -> Fin

Figura 4.3

María Guadalupe Vera Chávez

P L A C A S D E L G A D A S
Análisis de placas delgadas por el método del elemento finito

DATOS GENERALES

Núm. de puntos	frontera:	20
Núm. de condiciones de carga:		1
Núm. de elementos	finitos:	6
Núm. de	nodos:	29
Núm. de	materiales:	1
Núm. de grados de	libertad:	3
Núm. de nodos	cargados:	9
Indicador de:		
Fuerzas de	cuerpo:	1
Cálculo de	esfuerzos:	1
Espesor de	la placa:	00.09000
Cuadráticos o lineales (C/L):		C

Presione F10 para terminar

Figura 4.4

María Guadalupe Vera Chávez

P L A C A S D E L G A D A S
 Análisis de placas delgadas por el método del elemento finito

CARACTERISTICAS DE LOS MATERIALES

Material No.	Mod. de elasticidad (ton/m ²)	Coef. de Poisson	Peso volumétrico (ton/m ³)
1	2200000.00	0.2000	2.4000

PgDn Pág.sig. PgUp Pág.ant. F10 Fin

Figura 4.5

María Guadalupe Vera Chávez

P L A C A S D E L G A D A S
 Análisis de placas delgadas por el método del elemento finito

DATOS DE COORDENADAS

NODO	X	Y
1	00006.00	00004.00
2	00006.00	00003.00
3	00006.00	00002.00
4	00006.00	00001.00
5	00006.00	00000.00
6	00005.00	00004.00

Generación automática de coordenadas

Punto inicial X: .
 Punto inicial Y: .
 Incremento en X: .
 Incremento en Y: .
 Nudo inicial: .
 Incremento nudo: .
 Núm. de nudos: .

F10 Fin de generación automática de coordenadas

Figura 4.6

María Guadalupe Vera Chávez

P L A C A S D E L G A D A S
Análisis de placas delgadas por el método del elemento finito

DATOS DE LOS ELEMENTOS

elemento No.	Material No.	Numeración de nudos							
		1	2	3	4	5	6	7	8
1	1	1	6	9	10	11	7	3	2
2	1	9	14	17	18	19	15	11	10
3	1	17	22	25	26	27	23	19	10
4	1	3	7	11	12	13	8	5	4
5	1	11	15	19	20	21	16	13	12
6	1	19	23	27	28	29	24	21	20

PgDn Pág.sig. PgUp Pág. ant. F10 Fin

Figura 4.7

María Guadalupe Vera Chávez

P L A C A S D E L G A D A S
Análisis de placas delgadas por el método del elemento finito

CODIGOS DE FRONTERA

Nudo	X	Y	Z
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1
5	1	1	1
6	1	1	1

PgDn Pág.sig. PgUp Pág. ant. F10 Fin

Figura 4.8

María Guadalupe Vera Chávez

P L A C A S
Análisis de Placas por el Método del Elemento Finito

Versión 1.0

DESPLAZAMIENTOS DE LOS NUDOS

NUDO	GIROX (RAD)	GIRO Y (RAD)	DESP. Z (M)
1	.0000	.0000	.0000
2	.0000	.0000	.0000
3	.0000	.0000	.0000
4	.0000	.0000	.0000
5	.0000	.0000	.0000
6	.0000	.0000	.0000
7	-2.4496E-08	1.5565E-02	1.0463E-02
8	.0000	.0000	.0000
9	.0000	.0000	.0000
10	-1.5735E-02	2.5244E-04	1.0552E-02
11	-4.1796E-08	3.9293E-04	2.1038E-02
12	1.5736E-02	2.5245E-04	1.0552E-02
13	.0000	.0000	.0000

PgDn -> Siguiente página

F10 -> Fin

Figura 4.11

María Guadalupe Vera Chávez

P L A C A S
Análisis de Placas por el Método del Elemento Finito

Versión 1.0

F1 Ayuda	F2 Abrir Archivo	F3 Definir Datos	F4 Gráfica de los nodos	F5 Gráfica de elementos
F6 Consulta Resultados	F7 Grafica Resultados	F8 Impresión Dat.y Res.	F9 Cerrar Archivo	F10 Fin

Nombre del archivo de resultados

TESIS.SAL

Verifique que la impresora
esté encendida, el papel
bien colocado y oprima :

<ENTER> para continuar <C> para cancelar la impresión.

Figura 4.12

María Guadalupe Vera Chávez

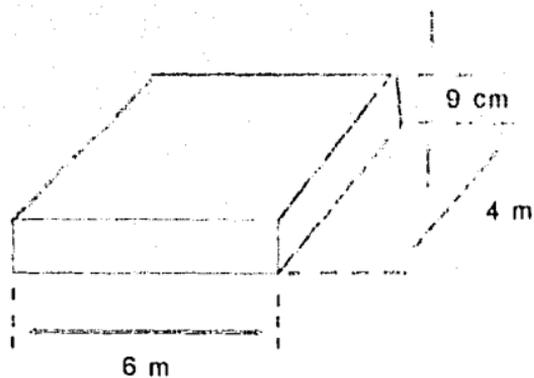


Figura 5.1.A Isométrico de la placa regular analizada

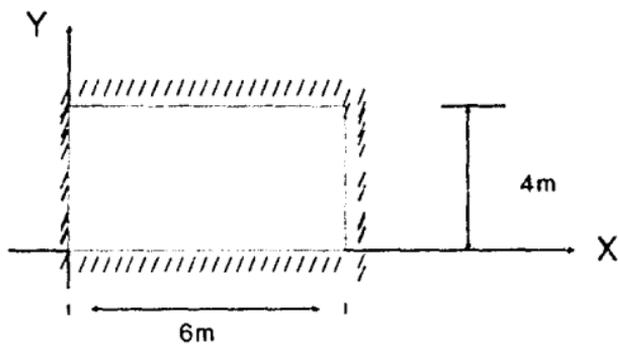


Figura 5.1.B Vista en planta de la placa analizada que muestra el empotramiento en sus 4 lados.

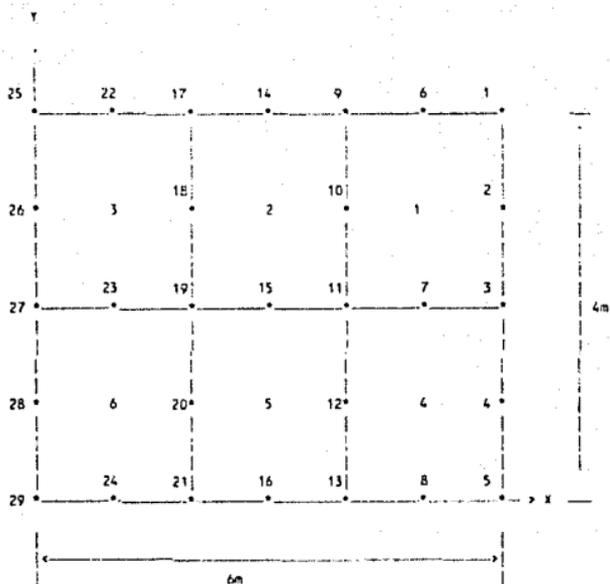


Figura 5.2.A Malla de elementos finitos de la placa regular analizada.

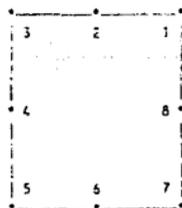


Figura 5.2.B Elemento finito cuadrilátero cuadrático utilizado, perteneciente a la familia Serendipity

GRAFICA DE LOS NODOS



Figura 5.3 Gráfica de los nodos del ejemplo de la placa regular, obtenida con el sistema.

GRAFICA DE ELEMENTOS

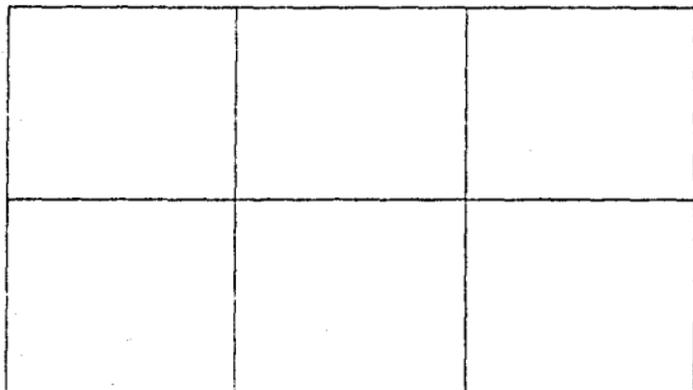


Figura 5.4 Gráfica de los elementos finitos de la placa regular, obtenida con el sistema.

GRAFICA DE DESPLAZAMIENTOS



Figura 3.5 Grafica de los desplazamientos de la placa regular unificada

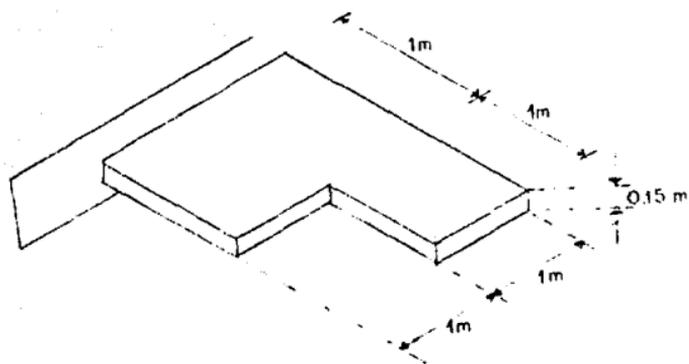


Figura 5.6 Isométrico de la placa irregular analizada.

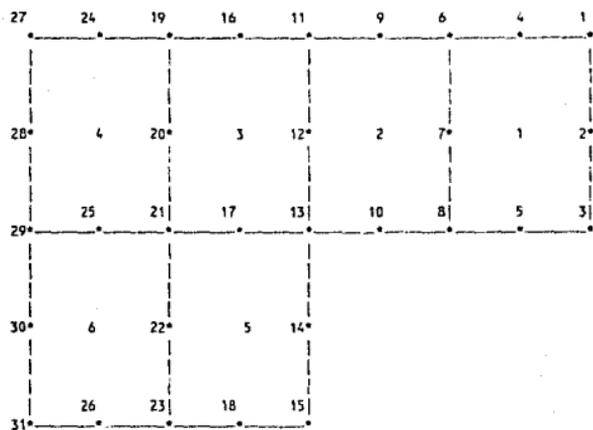
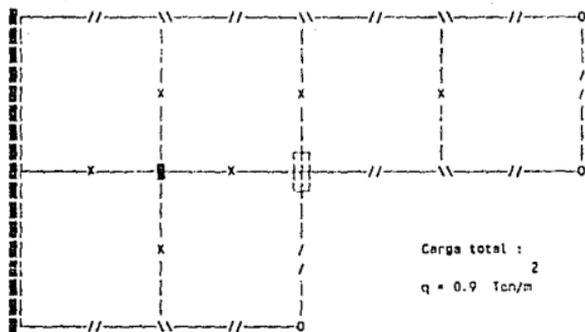


Figura 5.7 Malla de elementos finitos de la placa irregular analizada.



Cargas	Nodos
+0.8471	x
+0.4235	//
-0.2118	\\
-0.1059	o
-0.4235	■
-0.3177	□

Figura 5.8 Cargas aplicadas en cada nodo de la placa irregular analizada.

GRAFICA DE LOS NODOS



Figura 3.7 Gráfica de coordenadas de nodos mal capturadas

GRAFICA DE ELEMENTOS

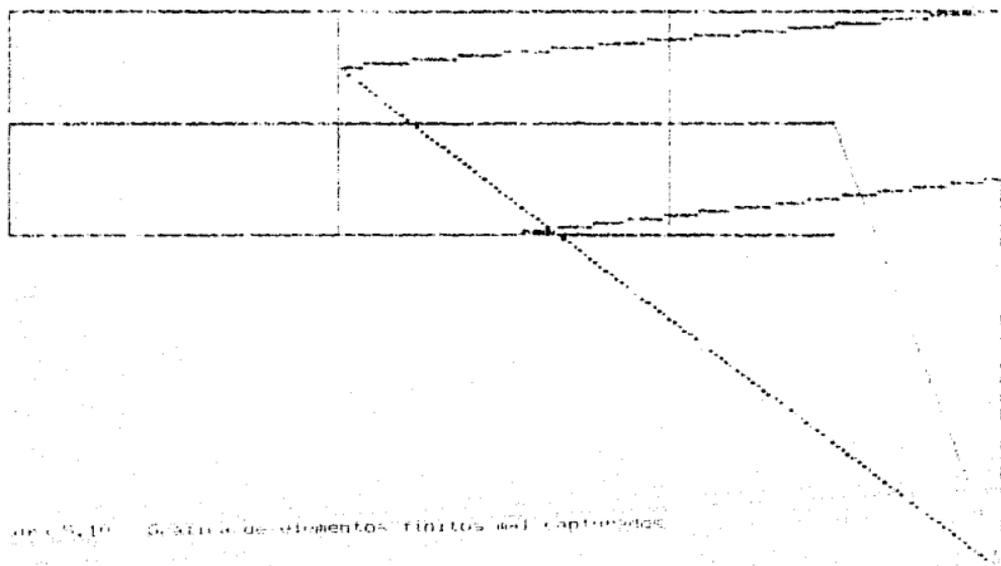


Fig. 5.10. Gráfica de elementos finitos del eje de transmisión.

APENDICE B

CUADROS

X	Y
X 1	Y 1
X 2	Y 2
X 3	Y 3
...	...
...	...
X n	Y n

Cuadro 3.1 Función tabular

CUADRO 4.1 Datos generales

PROYECTO:	
DIRECCION:	
ESTRUCTURA:	
NOMBRE DE ARCHIVO:	
FECHA:	REVISO:
OBSERVACIONES:	
D A T O S G E N E R A L E S	
NUMERO DE ELEMENTOS FINITOS:	
NUMERO DE NUDOS:	
NUMERO DE NUDOS CARGADOS:	
NUMERO DE MATERIALES:	
ESPEJOR DE LA PLACA (metros):	
NUMERO DE PUNTOS FRONTERA:	
NUMERO DE CONDICIONES DE CARGA:	
INDICADOR DE FUERZAS DE CUERPO:	
INDICADOR DE CALCULO DE ESFUERZOS:	
ELEMENTOS CUADRILATEROS = C □ LINEALES = L	
ELABORO:	HOJA _____ DE _____

CUADRO 4.2 Datos de los materiales

PROYECTO:			
DIRECCION:			
ESTRUCTURA:			
NOMBRE DE ARCHIVO:			
FECHA:		REVISO:	
OBSERVACIONES:			
D A T O S D E L O S M A T E R I A L E S			
MATERIAL NO.	MOD. DE ELAST. (T/M ²)	COEF. DE POISSON (ADIM)	PESO VOL. (T/M ³)
ELABORO:		HOJA _____ DE _____	

CUADRO 4 4 Numeración de los elementos

PROYECTO:									
DIRECCION:									
ESTRUCTURA:									
NOMBRE DE ARCHIVO:									
FECHA:					REVISO:				
OBSERVACIONES:									
N U M E R A C I O N D E L O S E L E M E N T O S									
NUMERO DE ELEMENTO	NUMERO DE MATERIAL	N U M E R A C I O N D E N U D O S							
		1	2	3	4	5	6	7	8
ELABORO:					HOJA _____ DE _____				

CUADRO 5.1 Datos generales

PROYECTO: <u>EJEMPLO DE APLICACION</u>	
DIRECCION	
ESTRUCTURA: <u>PLACA REGULAR</u>	
NOMBRE DE ARCHIVO: <u>EJEMPLO</u>	
FECHA: <u>26 DE ENERO DE 1991</u>	REVISO:
OBSERVACIONES	
D A T O S G E N E R A L E S	
NUMERO DE ELEMENTOS FINITOS:	6
NUMERO DE NUDOS	29
NUMERO DE NUDOS CARGADOS:	9
NUMERO DE MATERIALES:	1
ESPESOR DE LA PLACA (metros):	0.09
NUMERO DE PUNTOS FRONTERA:	20
NUMERO DE CONDICIONES DE CARGA:	1
INDICADOR DE FUERZAS DE CUERPO:	1
INDICADOR DE CALCULO DE ESFUERZOS	1
ELEMENTOS CUADRILATEROS = C o LINEALES = L	C
ELABORO:	HOJA <u>1</u> DE <u>8</u>

CUADRO 5.2 Datos de los materiales

PROYECTO: <u>EJEMPLO DE APLICACION</u>			
DIRECCION:			
ESTRUCTURA: <u>PLACA REGULAR</u>			
NOMBRE DE ARCHIVO <u>EJEMPLO</u>			
FECHA: <u>20 DE ENERO DE 1991</u>		REVISOR:	
OBSERVACIONES:			
DATOS DE LOS MATERIALES			
MATERIAL NO.	MOD. DE ELAST. (T/M ²)	COEF. DE POISSON (ADIM)	PESO VOL. (T/M ³)
1	2,200,000	0.2	2.40
ELABORO:		HOJA <u>2</u> DE <u>8</u>	

CUADRO 5.3 Coordenadas de los nodos

PROYECTO: <u>EJEMPLO DE APLICACION</u>						
DIRECCION:						
ESTRUCTURA: <u>PLACA REGULAR</u>						
NOMBRE DE ARCHIVO: <u>EJEMPLO</u>						
FECHA: <u>29 DE ENERO DE 1991</u>				REVISO		
OBSERVACIONES:						
C O O R D E N A D A S D E L O S N U D O S						
N U D O	C O O R D E N A D A S		G E N E R A R	D I F E R E N C I A E N		
	X	Y		K	Y	NUDO
1	6	4	4	0	-1	1
6	5	4	2	0	-2	1
9	4	4	4	0	-1	1
14	3	4	2	0	-2	1
17	2	4	4	0	-1	1
22	1	4	2	0	-2	1
25	0	4	4	0	-1	1
ELABORO:				HOJA <u>3</u> DE <u>8</u>		

CUADRO 5.4 Numeración de los elementos

PROYECTO: <u>EJEMPLO DE APLICACION</u>									
DIRECCION:									
ESTRUCTURA: <u>PLACA REGULAR</u>									
NOMBRE DE ARCHIVO <u>EJEMPLO</u>									
FECHA: <u>28 DE ENERO DE 1991</u>						REVISO:			
OBSERVACIONES:									
N U M E R A C I O N D E L O S E L E M E N T O S									
NUMERO DE ELEMENTO	NUMERO DE MATERIAL	N U M E R A C I O N D E N U D O S							
		1	2	3	4	5	6	7	8
1	1	1	6	9	10	11	7	3	2
2	1	9	14	17	18	19	15	11	10
3	1	17	22	25	26	27	23	19	18
4	1	3	7	11	12	13	8	5	4
5	1	11	15	19	20	21	16	13	12
6	1	19	23	27	28	29	24	21	20
ELABORO:						HOJA <u>4</u> DE <u>8</u>			

CUADRO 5.5 Códigos de frontera en los nudos

PROYECTO: <u>EJEMPLO DE APLICACION</u>			
DIRECCION:			
ESTRUCTURA: <u>PLACA REGULAR</u>			
NOMBRE DE ARCHIVO <u>EJEMPLO</u>			
FECHA: <u>28 DE ENERO DE 1991</u>		REVISO:	
OBSERVACIONES:			
CODIGOS DE FRONTERA EN LOS NUDOS			
N U D O	C F. EN X	C F. EN Y	C F EN Z
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1
5	1	1	1
6	1	1	1
8	1	1	1
16	1	1	1
24	1	1	1
ELABORO:		5 8 HOJA _____ DE _____	

CUADRO 5.5 Códigos de frontera en los nodos
(continuación)

PROYECTO: <u>EJEMPLO DE APLICACION</u>				
DIRECCION				
ESTRUCTURA: <u>PLACA REGULAR</u>				
NOMBRE DE ARCHIVO: <u>EJEMPLO</u>				
FECHA: <u>26 DE ENERO DE 1991</u>			REVISO	
OBSERVACIONES:				
CODIGOS DE FRONTERA EN LOS NUDOS				
N U D O	C. F. EN X	C. F. EN Y	C F. EN Z	
9	1	1	1	
17	1	1	1	
25	1	1	1	
13	1	1	1	
21	1	1	1	
29	1	1	1	
14	1	1	1	
22	1	1	1	
26	1	1	1	
ELABORO:			HOJA <u>6</u> DE <u>8</u>	

CUADRO 5.6 Cargas en los nodos

PROYECTO: <u>EJEMPLO DE APLICACION</u>			
DIRECCION:			
ESTRUCTURA: <u>PLACA REGULAR</u>			
NOMBRE DE ARCHIVO: <u>EJEMPLO</u>			
FECHA: <u>28 DE ENERO DE 1991</u>		REVISO	
OBSERVACIONES:			
C A R G A S E N L O S N U D O S			
N U D O	C A R G A S		
	X	Y	Z
7	0	0	+15.484
10	0	0	+15.484
11	0	0	- 7.742
19	0	0	- 7.742
12	0	0	+15.484
15	0	0	+15.484
20	0	0	+15.484
23	0	0	+15.484
ELABORO:		HOJA <u>8</u> DE <u>8</u>	

ANALISIS DE PLACAS POR EL ELEMENTO FINITO

DESPLAZAMIENTOS DE LA ESTRUCTURA

MUDO	GIRO X (mts)	GIRO Y (mts)	DESP. Z (mts)
1	.0000	.0000	.0000
2	.0000	.0000	.0000
3	.0000	.0000	.0000
4	.0000	.0000	.0000
5	.0000	.0000	.0000
6	.0000	.0000	.0000
7	-2.4496E-08	1.5565E-02	1.0463E-02
8	.0000	.0000	.0000
9	.0000	.0000	.0000
10	-1.5735E-02	2.5244E-04	1.0552E-02
11	-4.1796E-08	3.9293E-04	2.1038E-02
12	1.5736E-02	2.5245E-04	1.0552E-02
13	.0000	.0000	.0000
14	.0000	.0000	.0000
15	-4.9980E-08	3.5186E-07	2.1047E-02
16	.0000	.0000	.0000
17	.0000	.0000	.0000
18	-1.5736E-02	-2.5240E-04	1.0552E-02
19	-3.9445E-08	-3.9287E-04	2.1039E-02
20	1.5736E-02	-2.5241E-04	1.0552E-02
21	.0000	.0000	.0000
22	.0000	.0000	.0000
23	-2.6653E-08	-1.5566E-02	1.0463E-02
24	.0000	.0000	.0000
25	.0000	.0000	.0000
26	.0000	.0000	.0000
27	.0000	.0000	.0000
28	.0000	.0000	.0000
29	.0000	.0000	.0000

CUADRO 5.7 Desplazamientos de la placa regular analizada.

ANÁLISIS DE PLACAS POR EL ELEMENTO FINITO

ELEMENTOS MECÁNICOS (TON - M)

ELEM*	PUNTO*	COORDENADAS *		MOMENTOS *			CORTANTES *	
		X(M)	(M)	MX	MY	MOY	QX	QY
1	1	5.58	3.58	-.64	-.64	-.58	.25	-1.63
1	2	4.42	3.58	.11	-1.89	-.59	-1.87	.69
1	3	4.42	2.42	2.32	2.38	-.58	1.44	-.20
1	4	5.58	2.42	-1.87	.14	-.58	1.14	-5.60
1	5	5.00	3.00	-.02	-.00	-.87	145.00	139.41
2	1	3.58	3.58	-.49	-2.53	-.01	-7.43	1.08
2	2	2.42	3.58	-.49	-2.53	.01	-7.43	-1.08
2	3	2.42	2.42	.55	2.54	.00	-3.23	-.78
2	4	3.58	2.42	.55	2.54	-.00	-3.23	.78
2	5	3.00	3.00	.04	.01	.00	281.28	.00
3	1	1.58	3.58	.11	-1.87	.59	-1.87	-.69
3	2	.42	3.58	-.64	-.64	.58	.25	1.63
3	3	.42	2.42	-1.87	.14	.58	1.14	5.60
3	4	1.58	2.42	2.32	2.38	.58	1.44	-.20
3	5	1.00	3.00	-.02	-.00	.87	145.00	-139.41
4	1	5.58	1.58	-1.87	.14	.58	-1.14	-5.60
4	2	4.42	1.58	2.32	2.38	.58	-1.44	-.20
4	3	4.42	.42	.11	-1.89	.59	1.87	.69
4	4	5.58	.42	-.64	-.64	.58	-.25	-1.63
4	5	5.00	1.00	-.02	-.00	.87	-145.00	139.41
5	1	3.58	1.58	.55	2.54	.00	3.23	.78
5	2	2.42	1.58	.55	2.54	-.00	3.23	-.78
5	3	2.42	.42	-.49	-2.53	-.01	7.43	-1.08
5	4	3.58	.42	-.49	-2.53	.01	7.43	1.08
5	5	3.00	1.00	.04	.01	.00	-281.28	.00
6	1	1.58	1.58	2.32	2.38	-.58	-1.44	.20
6	2	.42	1.58	-1.87	.14	-.58	-1.14	5.60
6	3	.42	.42	-.64	-.64	-.58	-.25	1.63
6	4	1.58	.42	.11	-1.89	-.59	1.87	-.69
6	5	1.00	1.00	-.02	-.00	-.87	-145.00	-139.41

CUADRO 5.8 Elementos mecánicos de la placa regular analizada.

ANALISIS DE PLACAS POR EL ELEMENTO FINITO

DESPLAZAMIENTOS DE LA ESTRUCTURA

NUDO	GIRO X (mts)	GIRO Y (mts)	DESP. Z (mts)
1	.1328E-02	-.1422E-01	.4174E-01
2	.1319E-02	-.1421E-01	.4108E-01
3	.1333E-02	-.1423E-01	.4042E-01
4	.1288E-02	-.1417E-01	.3463E-01
5	.1330E-02	-.1416E-01	.3331E-01
6	.1265E-02	-.1391E-01	.2758E-01
7	.1312E-02	-.1391E-01	.2694E-01
8	.1322E-02	-.1382E-01	.2628E-01
9	.1031E-02	-.1302E-01	.2075E-01
10	.1623E-02	-.1298E-01	.1944E-01
11	.6132E-03	-.1189E-01	.1440E-01
12	.1055E-02	-.1162E-01	.1399E-01
13	.1680E-02	-.1083E-01	.1331E-01
14	.1797E-02	-.9507E-02	.1233E-01
15	.1504E-02	-.8498E-02	.1142E-01
16	.2797E-04	-.1017E-01	.8790E-02
17	.9859E-03	-.9235E-02	.8254E-02
18	.1337E-02	-.8334E-02	.7148E-02
19	-.3271E-03	-.7871E-02	.4243E-02
20	.2082E-03	-.7550E-02	.4266E-02
21	.3651E-03	-.7262E-02	.4099E-02
22	.6158E-03	-.6813E-02	.3833E-02
23	.1213E-02	-.6532E-02	.3341E-02
24	-.3003E-03	-.4458E-02	.1136E-02
25	.1355E-03	-.4409E-02	.1150E-02
26	.6270E-03	-.3545E-02	.8590E-03
27	.0000	.0000	.0000
28	.0000	.0000	.0000
29	.0000	.0000	.0000
30	.0000	.0000	.0000
31	.0000	.0000	.0000

CUADRO 5.9 Desplazamientos de la placa irregular analizada.

ANALISIS DE PLACAS POR EL ELEMENTO FINITO

ELEMENTOS MECANICOS (TON - M)

ELEMENTO NO.	PUNTO NO.	COORDENADAS		ELEMENTOS			CORTANTES	
		X (M)	Y (M)	MX	MY	MYX	QX	QY
1	1	.79	.79	-.02	.00	-.01	-.01	.16
1	2	.21	.79	-.35	-.02	.00	.06	.62
1	3	.21	.21	-.42	-.07	-.03	-.04	.69
1	4	.79	.21	-.03	.02	.00	.03	.18
1	5	.50	.50	-.19	-.01	.00	.35	-3.98
2	1	.79	.79	-.97	-.02	-.01	.11	1.08
2	2	.21	.79	-1.61	.13	-.20	.00	.93
2	3	.21	.21	-2.40	.16	-.31	1.31	2.08
2	4	.79	.21	-.86	.02	.05	-.14	.96
2	5	.50	.50	-1.41	.09	-.08	1.77	-19.39
3	1	.79	.79	-2.15	.17	-.43	-.19	2.22
3	2	.21	.79	-2.93	.11	-.33	-.51	1.68
3	3	.21	.21	-2.75	-.22	-.47	.07	2.49
3	4	.79	.21	-2.04	.29	.70	.09	-2.55
3	5	.50	.50	-2.52	.09	-.47	-1.76	-8.74
4	1	.79	.79	-4.04	-.21	-.15	.50	2.61
4	2	.21	.79	-5.86	-.98	.09	1.52	2.20
4	3	.21	.21	-5.87	-1.10	-.04	-.95	3.01
4	4	.79	.21	-3.68	-.57	-.20	.04	.69
4	5	.50	.50	-4.85	-.70	-.07	-2.49	-14.75
5	1	.79	.79	-1.54	-.10	-.87	2.70	3.94
5	2	.21	.79	-2.50	-.23	-.56	-3.10	.35
5	3	.21	.21	-2.39	.06	-.29	-1.74	1.50
5	4	.79	.21	-.47	-.24	-.54	.66	.47
5	5	.50	.50	-1.71	-.12	-.54	-6.73	-14.62
6	1	.79	.79	-3.52	-.46	-.41	1.54	3.44
6	2	.21	.79	-5.58	-1.03	-.23	-2.04	3.98
6	3	.21	.21	-4.85	-.78	-.36	-3.77	.24
6	4	.79	.21	-3.60	-.08	-.43	1.65	.73
6	5	.50	.50	-4.37	-.57	-.36	-.25	-9.41

CUADRO 5.10 Elementos mecánicos de la placa irregular analizada.

APENDICE C

LISTADOS

DE LOS

PROGRAMAS

```

/* .....
 * PUTINA PRINCIPAL DE CAPTURA
 * .....
 */

```

```

/* Continuarion del programa PLACAS pag. 2/24 */

```

```

#include "declarac.h"

void graba(int *coordenadas x, int *coordenadas y, int *elementos),
    int no_nodos, int no_elementos;
void lee(char cadena[], void *func, int *funcion) { tamaño;
void lee_real(char cadena[], void *func, int *funcion) { tamaño;
void lee_int(char cadena[], int *funcion);
void lee_real_descaral();
void lee_real_descaral(int n);
void lee_real_descaral(char *ch, int *funcion);
void lee_real_descaral(int n);
void lee_real_descaral(char *func, char *ch, int *funcion);
void lee_real_descaral(int n);
void lee_real_descaral(char *func, char *ch, int *funcion);
void lee_real_descaral(int n);

/* Variables usadas por el programa
1 Apunta al renglon en donde se va a escribir
lin_des Indica el elemento que se va a imprimir
*/

void desp_carga(int i, int lin_des)
{
    escribe(p_carga[lin_des].nudo, 5, 2*(i*21));
    escribe_real(p_carga[lin_des].fx, 2, 3, 11, 2*(i*21));
    escribe_real(p_carga[lin_des].fy, 2, 3, 22, 2*(i*21));
    escribe_real(p_carga[lin_des].fz, 2, 3, 33, 2*(i*21));
}

void desp_material(int i, int lin_des)
{
    escribe(p_mat[lin_des].material.num, 5, 3, 2*(i*21));
    escribe_real(p_mat[lin_des].elast.cuidad, 7, 2, 2, 2*(i*21));
    escribe_real(p_mat[lin_des].poisson, 1, 8, 8, 2*(i*21));
    escribe_real(p_mat[lin_des].peso_volumetrico, 1, 4, 6, 2*(i*21));
}

void desp_frontera(int i, int lin_des)
{
    escribe(p_codigos[lin_des].nudo, 5, 2*(i*21));
    gotoxy(11, 2*(i*21)); putch(p_codigos[lin_des].x);
    gotoxy(16, 2*(i*21)); putch(p_codigos[lin_des].y);
    gotoxy(20, 2*(i*21)); putch(p_codigos[lin_des].z);
}

void desp_elementos(int i, int lin_des, int num_nodos)
{
    int k;
    escribe(p_elem[lin_des].num_elemento, 1, 6, 2*(i*21));
    gotoxy(19, 2*(i*21)); putch(p_elem[lin_des].num_material);
    for (k=0; k<num_nodos; k++)
        escribe(p_elem[lin_des].nudo[k], 3, 28+(k*5), 2*(i*21));
}

```

```

void inicializa()
{
    int i, j;
    for (i=0; i<6; i++) strcpy(geom[1][i], "51");
    for (j=0; j<2; j++) for (i=0; i<100; i++) strcpy(coordenas[1][i][j], "21");
    for (i=0; i<5; i++)
    {
        strcpy(p_mat[i].material.num, "51");
        strcpy(p_mat[i].elast.cuidad, "91");
        strcpy(p_mat[i].poisson, "51");
        strcpy(p_mat[i].peso_volumetrico, "51");
    }
    for (i=0; i<20; i++)
    {
        strcpy(p_elem[i].num_elemento, "21");
        p_elem[i].num_material="";
        for (j=0; j<2; j++) strcpy(p_elem[i].nudo[j], "71");
    }
    for (i=0; i<100; i++)
    {
        strcpy(p_codigos[i].nudo, "51");
        strcpy(p_codigos[i].x, "0");
        strcpy(p_codigos[i].y, "0");
        strcpy(p_codigos[i].z, "0");
    }
    for (i=0; i<50; i++)
    {
        strcpy(p_carga[i].nudo, "51");
        strcpy(p_carga[i].fx, "51");
        strcpy(p_carga[i].fy, "51");
        strcpy(p_carga[i].fz, "51");
    }
}

```

Comienza el Programa principal

```
main()
{
  int x,y,funcion,i1,i2,i1k,i2k,lin_des,comienzo,incorrecto,abrir=0;
  int nodo_inicio,i_nodo,incremento,num_nodos,nodos_car,no_nodos;
  int coordenada_x[100],coordenada_y[100],elementos[50][2];
  char pal[600],pant[600];
  float general_x,general_y,k,incremento_x,incremento_y,kx,k,ux;
  linea[general?];

  cargar_tablas(1);
  t[0][0];
  p[0][0][0];
  d[0][0];
  cuadro_ini();
  dete_funciones(1);
  gettext(1,0,0,0,0);
  sel_funcion(1);
  for (i=0;i<3;i++) strcpy(general[i],"");

  while (func[1]!=0)
  {
    if (func[1])
      ayuda[ayuda].xat = "0";

    if (func[2])
    {
      canal_1,10,37,12,"00000000";
      canal_1,12,37,14,"00000000";
      gotoxy(1,12); printf("Borre del archivo de datos ");
      invierte();
      gotoxy(12,12); lee(archivo_datos,&func,&funcion,12);
      recat[recat];
      arch_datos=fopen(archivo_datos,"r");
      k=toxy(2,23);
      if (arch_datos==NULL)
      {
        printf("Se crea el archivo. Presione ENTER para continuar");
        arch_datos=fopen(archivo_datos,"w");
        inicializa();
      }
      else
        printf("Abriendo archivo . Presione ENTER para continuar");
    }
  }
}
```

Lectura de datos generales

```
for (i=0;i<3;i++) gen_general[i]=fgetc(arch_datos);
p_gen_ico=fgetc(arch_datos);
p_gen_ice=fgetc(arch_datos);
p_gen_nodos=fgetc(arch_datos);
for (i=0;i<3;i++) gen_exposor[i]=fgetc(arch_datos);
for (i=0;i<3;i++) gen_general[i]=fgetc(arch_datos);
fgetc(arch_datos);
for (i=0;i<3;i++)
{
  llena_caros[ap_gen_general[i]];general[i]=cuantos_gen_general[i][5];
}
llena_caros[4]=gen_nodos;
nodos_car[car]=gen_nodos[5];
no_nodos=ap_gen_nodos+gen_nodos;

/* Lectura de los materiales */
for (i=0;i<3;i++)
{
  for (j=0;j<5;j++) mate[i].material_nudo[j]=fgetc(arch_datos);
  for (j=0;j<5;j++) mate[i].elasticidad[j]=fgetc(arch_datos);
  for (j=0;j<5;j++) mate[i].poisson[j]=fgetc(arch_datos);
  for (j=0;j<5;j++) mate[i].peso_volumetrico[j]=fgetc(arch_datos);
  fgetc(arch_datos);
}

/* Lectura de las condiciones */
for (i=0;i<3;i++)
{
  for (j=0;j<7;j++) con_ondas[i][0][j]=fgetc(arch_datos);
  for (j=0;j<7;j++) con_coordenadas[i][1][j]=fgetc(arch_datos);
  fgetc(arch_datos);
}

/* Lectura de los codigos de frontera */
for (i=0;i<3;i++)
{
  for (j=0;j<3;j++) codigos[i].nudo[j]=fgetc(arch_datos);
  p_codigos[i].x=fgetc(arch_datos);
  p_codigos[i].y=fgetc(arch_datos);
  p_codigos[i].z=fgetc(arch_datos);
  fgetc(arch_datos);
}

/* Lectura de los elementos */
for (i=0;i<3;i++)
{
  for (j=0;j<3;j++) elem[i].num_elemento[j]=fgetc(arch_datos);
  p_elem[i].num_material=fgetc(arch_datos);
}

for (i=0;i<3;i++) for (j=0;j<3;j++) elem[i].nudo[k[j]]=fgetc(arch_datos);
fgetc(arch_datos);
}
```

Lectura de las condiciones de carga

```

for (i=0; i<nodos_car+1; i++)
{
for (j=0; j<1; j++) p.cargal[i][nodosj]=fgetc(arch_datos);
for (j=0; j<5; j++) p.cargal[i][5+j]=fgetc(arch_datos);
for (j=0; j<5; j++) p.cargal[i][15+j]=fgetc(arch_datos);
for (j=0; j<5; j++) p.cargal[i][25+j]=fgetc(arch_datos);
fgetc(arch_datos);
}
}
getch();
window(2,10,79,21);
clrscr();
window(1,1,80,25);
gotoxy(1,23);
for (i=1; i<79; i++) putchar(' ');
clrscr();
puttext(1,1,80,25,p);
sel_funcion(f10,1);
}

if (func==f3) /* Definir datos */
{
despl_funciones(f11,1,5);
gotoxy(1,25);cputa(" Figura 4.2");
gettext(1,1,80,25,pant);
sel_funcion(f11,1);
selecciona(func,&n,&funcion);
while(func!=10)
{
if (func==f1) ayuda("ayuda.txt " "10");
if (func==f2 || func==f4 || func==f6 || func==f5 || func
func==f7)
despl_f1();
if (func==f2)
{
gotoxy(1,25);cputa(" Figura 4.4");
datos_generales();
for (i=0; i<6; i++) escribe(p.gen.genera1[i],5,53,8+i);
escribe(p.gen.nnc,5,53,16);
gotoxy(53,16);putch(p.gen.ice);
gotoxy(53,17);putch(p.gen.ice);
escribe_real(p.gen.espeso,2,5,53,18);
gotoxy(53,19);putch(p.gen.modos);
i++;
invierte(i);
}
}
}

```

```

while (func!=10 && i<6)
{
gotoxy(53,8+i);
leer(&p.gen.genera1[i],&func,&funcion,5);
if (func==farr)
{
reabreice();
escribe(p.gen.genera1[i],5,53,8+i);
if (i==0) i++; else i--;
invierte(i);
escribe(p.gen.genera1[i],5,53,8+i);
}
if (func==farr || func==frot)
{
reabreice();
escribe(p.gen.genera1[i],5,53,8+i);
i++;
if (i==6)
{
invierte(i);
escribe(p.gen.genera1[i],5,53,8+i);
}
}
if (func!=10)
{
gotoxy(1,16);leer(&i,&n,&func,&funcion,5);
incorrecto=1;
while (incorrecto)
{
gotoxy(53,16);leer(&p.gen.ice,&func,&funcion,1);
if (p.gen.ice=="1" || p.gen.ice=="0")
{
gotoxy(53,16);putch(p.gen.ice);incorrecto=0;
}
}
incorrecto=1;
while (incorrecto)
{
gotoxy(53,17);leer(&i,&n,&func,&funcion,1);
if (p.gen.ice=="1" || p.gen.ice=="0")
{
gotoxy(53,17);putch(p.gen.ice);incorrecto=0;
}
}
}
gotoxy(53,18);leer_real(&p.gen.espeso,&func,&funcion,2,5);
incorrecto=1;
}
}

```



```

if (func==retur) { if (cas==1)
if (func==gdn)
if (lin_des!=general[4]-1)
restablece();
desp.material[1,lin_des];
comienzo=lin_des+1;
comienzo=comienzo+general[4] ; comi=2*% . general[4] ;
for (i=0; i<= % $% ; i=comienzo+general[4]; i++)
desp.material[1,comienzo];
for (i=1; i<=i+1)
{
gotoxy(1,2*(i+1));for (j=1; j<=69; j++) printf(" ");
i=0; i++;lin_des=comienzo+invierte(i);
desp.material[1,lin_des];
}
if (func==gkup)
if (lin_des>0)
{
restablece();
desp.material[1,lin_des];
comienzo=lin_des+1; lin_des=6 - 0;
for (i=0; i<= % $% ; i=comienzo+general[4]; i++)
desp.material[1,comienzo];
i=0; i++;lin_des=comienzo+invierte(i);
desp.material[1,lin_des];
}
restablece();
window(1,1,80,25);
gotoxy(80,5);printf(1,1,80,25,pant);
}
if (func==1) /* Datos de coordenadas */
{
coordenadas();
gotoxy(1,25);printf(" Figura %d",i);
for (i=1; i<= % $% ; i=general[3]; i++)
{
caso(1,2*(i+2),42,2*(i+2),"%d%$s%$");
gotoxy(11,2*(i+1));printf("23",i);
}
window(5,9,42,21);
func=0;
for (i=0; i<= % $% ; i=general[3]; i++)
for (j=0; j<=2; j++) escribe_real(p,coordenadas[i][j],5,2,11*(i+1),2*(j+2));
i=0; i++;lin_des=comienzo+0;
}

```

```

invierte();
escribe_real(p,coordenadas[i][0],5,2,11,2);
escribe_real(p,coordenadas[i][1],5,2,22,2*(i+2));
while (func!=10)
{
gotoxy(11,2*(i+1));
leen_real(p,coordenadas[lin_des][3],4func,4funcion,5,2);
if (func==arr)
if (lin_des!=0)
{
restablece();
escribe_real(p,coordenadas[lin_des][0],5,2,11,2*(i+2));
escribe_real(p,coordenadas[lin_des][1],5,2,22,2*(i+2));
if (i>0)
{
lin_des=i-1;
}
else
{
if (lin_des==0)
{
gotoxy(1,1);in(1,1);in(1,1);lin_des= ;
window(1,1,80,25);
caso(1,9,42,11,2*(i+2),"%d%$s%$");
restablece();
window(5,9,42,21);
}
}
invierte();
gotoxy(4,2*(i+2));printf("24",lin_des+1);
escribe_real(p,coordenadas[lin_des][0],5,2,11,2*(i+2));
escribe_real(p,coordenadas[lin_des][1],5,2,22,2*(i+2));
}
if (func==darr)
if (lin_des!=general[3]-1)
{
restablece();
escribe_real(p,coordenadas[lin_des][0],5,2,11,2*(i+2));
escribe_real(p,coordenadas[lin_des][1],5,2,22,2*(i+2));
if (i<5)
{
lin_des=i+1;
}
else
{
if (lin_des==general[3])
{
gotoxy(1,1);d(1,1);d(1,1);lin_des++;
}
}
}
}

```



```

    if (func=="darr" || func=="retu")
    {
        restablece();
        escribe_real(genera[1],3,2,23,4+i);
        i=i+6; i=i+1; i=0;
        invierte();
        escribe_real(genera[1],3,2,23,4+i);
    }
    restablece();
    clrscr();
    window(1,1,80,25);
    gotoxy(2,23);cputs("F) Generacion automatica   P&Dn P&g.sig.  ");
    cputs("PgUp P&g. ant.   F10 Fin ");
    for (i=0;i<7;i++)
    {
        ceros_llena(genera[i],5);
        llena_ceros(genera[i]);
    }
    inicial_x  *cin_real(genera[0],5)/100;
    inicial_y  *cin_real(genera[1],5)/100;
    A_incremento*cin_real(genera[2],5)/100;
    Y_incremento*cin_real(genera[3],5)/100;
    nodo_inicial  *cin(genera[4],5)/100;
    nodo_incremento*cin(genera[5],5)/100;
    num_nodos     *cin(genera[6],5)/100;
    i=0;
    for (k=nodo_inicial;k<=genera[3] && k<num_nodos;k+=nodo_incremento)
    {
        i++;
        icn(inicial_x*100,5p.coordenadas[k-1][0],7);
        icn(inicial_y*100,5p.coordenadas[k-1][1],7);
        inicial_x+=k_incremento;
        inicial_y+=y_incremento;
    }
    window(9,9,42,21);
    for (i=0;i<6 && i<genera[3];i++)
        for (j=0;j<7;j++)
    {
        gotoxy(4,2+(i*2));printf("%2d",i+1);
        escribe_real(ip.coordenadas[1][j],5,2,11+j*11,2+(i*2));
        i=0;j=0;lin_des=0;comienzo=0;
        invierte();
        escribe_real(ip.coordenadas[0][0],5,2,11,2);
        escribe_real(ip.coordenadas[0][1],5,2,22,2);
        func=norm;
    }
}

```

```

    gotoxy(80,5);puttext(1,1,80,25,pant);
    restablece();
    window(1,1,80,25);
}
if (func=="f5") /* Datos de elementos */
{
    datos_de_elementos();
    gotoxy(1,25);cputs("
    for (i=1;i<=6 Figura 4.7");
    for (i=1;i<=6 && i<genera[3];i++
    caja(5,7+(i*2),76,9+(i*2),"|_|_|_|_|_|");
    gotoxy(2,23);cputs(" P&Dn P&g.sig.   PgUp P&g. ant.   F10 Fin ");
    func=norm;
    window(5,9,76,21);
    for (i=0;i<6 && i<genera[2];i++) desp_elementos(i,i.no_nodos);
    i=0;j=1;k=0;lin_des=0;comienzo=0;
    invierte();
    desp_elementos(i.lin_des,no_nodos);
    while (func!="f10")
    {
        switch (j)
        {
            case 1: k=0;gotoxy(6,2+(i*2));
                    leen(&p.elem[lin_des].num_elemento,&func,&funcion,3);break;
            case 2: gotoxy(19,2+(i*2));
                    leen(&p.elem[lin_des].num_material,&func,&funcion,1);break;
            case 3: gotoxy(28+(k*5),2+(i*2));
                    leen(&p.elem[lin_des].nodo(k),&func,&funcion,3);break;
        }
    }
    if (func=="uarr")
    {
        if (lin_des==0)
        {
            restablece();
            desp_elementos(i.lin_des,no_nodos);
            if (i>0)
            {
                lin_des--;i--;j*=1;
            }
            else
            {
                if (lin_des>0){
                    gotoxy(1,1);insline();insline();lin_des--;
                    window(1,1,80,25);
                    caja(5,9,76,11,"|_|_|_|_|_|");
                    restablece();
                    window(5,9,76,21);
                }
            }
            invierte();
            desp_elementos(i.lin_des,no_nodos);
        }
    }
}

```

```

if (func=="darr")
if (lin_des!=general[2]-1)
{
restablece();
desp_elementos(i,lin_des,no_nodos);
if (i<5 && lin_des<general[2]-1)
|
lin_des++;i++;j=1;
|
else
{
if (lin_des<general[2]-1)
gotoxy(1,1);delline();delline();lin_des++;
window(1,1,80,25);j=1;
case(5,19,76,21,"p-[[#=#]]~");
restablece();
window(5,9,76,21);|
|
invierte();
desp_elementos(i,lin_des,no_nodos);
|
}
}

if (func=="retu")
if (j<3)
j++;
else
if (k<no_nodos-1)
k++;

if (func=="pgdn")
if (lin_des!=general[2]-1)
{
restablece();
desp_elementos(i,lin_des,no_nodos);
comienzo=lin_des/6+2;
comienzo=(comienzo*6<general[2]) ? comienzo*6 : general[2]-1;
for (i=0;i<6 && i-comienzo<general[2];i++)
desp_elementos(i,i-comienzo,no_nodos);
for (ii=i;i<6;ii++)
|
gotoxy(2,2+(ii*2));for (i=1;i<66;i++) putchar(' ');
|
i=0;j=1;lin_des=comienzo;invierte();
desp_elementos(i,lin_des,no_nodos);
|
}
}

```

```

if (func=="pgup")
if (lin_des!=0)
{
restablece();
desp_elementos(i,lin_des,no_nodos);
comienzo=(lin_des<6>0) ? lin_des-6 : 0;
for (i=0;i<6 && i-comienzo<general[2];i++)
desp_elementos(i,i-comienzo,no_nodos);
i=0;j=1;lin_des=comienzo;invierte();
desp_elementos(i,lin_des,no_nodos);
|
}
restablece();
window(1,1,80,25);
gotoxy(80,5);puttext(1,1,80,25,pant);
}

if (func=="t6") /* Condiciones frontera */
{
codi_frontera1;
gotoxy(1,25);cputs(" Figura 4.8");
for (i=0;i<7;i++) for (k=0;k<7;k++) i-comienzo<general[0];i++)
case(29,3+(i*2),50,9+(k*2),"p-[[#=#]]");
func=norm;
window(29,9,50,21);
for (i=0;i<6 && i<general[0];i++) desp_frontera1(i);
i=0;j=1;lin_des=0;comienzo=0;
invierte();
desp_frontera(0,lin_des);
while (func!="F10")
|
switch (j)
{
case 1: gotoxy(1,2+(i*2));
leer(p.codigo[lin_des].nudo.&func.&funcion,5);
break;
case 2: incorrecto=TRUE;
while (incorrecto)
|
gotoxy(1,2+(i*2));
leer(p.codigo[lin_des].x.&func.&funcion,1);
if (p.codigo[lin_des].x=="0" ||
p.codigo[lin_des].x=="1") incorrecto=FALSE;
|
}
break;
}
}

```

```

case 3:
  incorrecto=TRUE;
  while (incorrecto)
  {
    gotoxy(16,2*(i*2));
    leen(&p.codigo$[lin_des].y,&func,&funcion,1);
    if (p.codigo$[lin_des].y=="0" ||
        p.codigo$[lin_des].y=="1") incorrecto=FALSE;
  }
  break;
case 4:
  incorrecto=TRUE;
  while (incorrecto)
  {
    gotoxy(20,2*(i*2));
    leen(&p.codigo$[lin_des].z,&func,&funcion,1);
    if (p.codigo$[lin_des].z=="0" ||
        p.codigo$[lin_des].z=="1") incorrecto=FALSE;
  }
  i=i+1;
  break;
}

if (func=="retu" || (i<4) >>>);

if (func=="arr")
  if (lin_des!=0)
  {
    restablece();
    desp_frontera(1,lin_des);
    if (i>0)
    {
      lin_des--:i--:i+1;
    }
    else
    {
      if (lin_des>0)
      {
        gotoxy(1,1);insline():insline():lin_des--;
        window(1,1,80,25);
        caja(29,9,50,11,"|-----|");
        restablece();
        window(29,9,50,21);
      }
    }
    invierte();
    desp_frontera(1,lin_des);
  }
}

```

```

if (func=="dart")
  if (lin_des!=general[0]-1)
  {
    restablece();
    desp_frontera(i,lin_des);
    if (i<5 && lin_des=general[0]-1)
    {
      lin_des++:i++:i+1;
    }
    else
    {
      if (lin_des<general[0]-1)
        gotoxy(1,1);delline():delline():lin_des++;
      window(1,1,80,25);
      caja(29,9,50,21,"|-----|");
      restablece();
      window(29,9,50,21);
    }
    invierte();
    desp_frontera(1,lin_des);
  }

if (func=="pgdn")
  if (lin_des!=general[0]-1)
  {
    restablece();
    desp_frontera(i,lin_des);
    comienzo=lin_des/6+1;
    comienzo=(comienzo*6+general[0]-1)
    for (i=0:i<6 54 i=comienzo+general[0]:i++)
    desp_frontera(1,i*comienzo);
    for (i=1:i<6:i++)
    {
      gotoxy( 3,2*(i*2));for (i=1:i<20:i++) putch(' ');
    }
    i=0:i=1:lin_des=comienzo:invierte();
    desp_frontera(1,lin_des);
  }

if (func=="pgup")
  if (lin_des!=0)
  {
    restablece();
    desp_frontera(1,lin_des);
    comienzo=(lin_des-6)/6+1;
    for (i=0:i<6 54 i=comienzo+general[0]:i++)
    desp_frontera(1,i*comienzo);
    i=0:i=1:lin_des=comienzo:invierte();
    desp_frontera(1,lin_des);
  }
}

```



```

if (func==pgup)
if (lin_des!=0)
{
restablece();
desp_carga(lin_des);
comienzo=(lin_des-6)*7; lin_des=6+0;
for (i=0; i<=4; i=comienzo+nodos_car;i++) desp_carga(i+comienzo);
i=0; i=1; lin_des=comienzo+invierte(i);
desp_carga(i,lin_des);
}
restablece();
window(1,1,80,25);
gotoxy(80,5);puttext(1,1,80,25,pant);
}

```

```

/* if (func==f10) Fin de Definir Datos */
lee_teclado(&func,&sch,&funcion);
gotoxy(1,1);puttext(1,1,80,25,pa);
}

```

```

if (func==f6) /* Grafica de los nodos */
{
for (ii=0; ii<general[3]; ii++)
{
llena_ceros(&p.coordenadas[ii][0]);
ceros_llena(&p.coordenadas[ii][0],7);
llena_ceros(&p.coordenadas[ii][1]);
ceros_llena(&p.coordenadas[ii][1],7);
coordenadas_x[ii]=cin(p.coordenadas[ii][0],7);
coordenadas_y[ii]=cin(p.coordenadas[ii][1],7);
}
gettext(1,1,80,25,pant);
graphf4(coordenadas_x,coordenadas_y,general[3]);
puttext(1,1,80,25,pant);
}

```

```

if (func==f5) /* Grafica de los elementos */
{
for (ii=0; ii<general[3]; ii++)
{
llena_ceros(&p.coordenadas[ii][0]);
ceros_llena(&p.coordenadas[ii][0],7);
llena_ceros(&p.coordenadas[ii][1]);
ceros_llena(&p.coordenadas[ii][1],7);
coordenadas_x[ii]=cin(p.coordenadas[ii][0],7);
coordenadas_y[ii]=cin(p.coordenadas[ii][1],7);
}
for (ij=0; ij<general[2]; ij++)
for (i)=0; i<=8; i++)
{
llena_ceros(&p.elem[ii].nudo[i][j]);
ceros_llena(&p.elem[ii].nudo[i][j],3);
elementos[i][i]=cin(p.elem[ii].nudo[i][j],3);
}
gettext(1,1,80,25,pant);
graphf5(coordenadas_x,coordenadas_y,elementos,general[3],general[2]);
puttext(1,1,80,25,pant);
}

```

```

if (func==f6) /* Consulta de resultados */
consulta();

if (func==f7) /* Grafica de resultados */
graphf6();

if (func==f8) /* Impresion de datos y resultados */
{
gettext(1,1,80,25,pant);
imprime();
puttext(1,1,80,25,pant);
}

gotoxy(80,5);
lee_teclado(&func,&sch,&funcion);
}

```

```

if (labrio)
{
fclose(arch_datos);
arch_datos=fopen(archivo_datos,"w");

/* Impresion de datos generales */
for(i=0;i<3;i++) {putc(p.gen.gen[i],arch_datos);
putc(p.gen.afc,arch_datos);
putc(p.gen.ace,arch_datos);
putc(p.gen.podos,arch_datos);
llena_ceros(4,p.gen.espeor,1);
ceros_llena(4,p.gen.espeor,7);
for(i=0;i<7;i++) {putc(p.gen.espeor[i],arch_datos);
for (j=0;j<6;j++) {putc(p.gen.general[i][j],arch_datos);
putc('\n',arch_datos);}}

/* Impresion de los materiales */
for (i=0;i<general[4];i++)
{
llena_ceros(5,p.mate[i].poisson);
llena_ceros(4,p.mate[i].peso_volumetrico);
ceros_llena(4,p.mate[i].poisson,5);
ceros_llena(4,p.mate[i].peso_volumetrico,5);
for (j=0;j<5;j++) {putc(p.mate[i].material_num[i],arch_datos);
for (k=0;k<9;k++) {putc(p.mate[i].elasticidad[k],arch_datos);
for (l=0;l<5;l++) {putc(p.mate[i].poisson[l],arch_datos);
for (m=0;m<5;m++) {putc(p.mate[i].peso_volumetrico[m],arch_datos);
putc('\n',arch_datos);}}}}

/* Impresion de las coordenadas */
for (i=0;i<general[3];i++)
{
for(j=0;j<2;j++) {putc(p.coordenadas[i][0][j],arch_datos);
for(k=0;k<2;k++) {putc(p.coordenadas[i][1][k],arch_datos);
putc('\n',arch_datos);}}

/* Impresion de los codigos de frontera */
for (i=0;i<general[6];i++)
{
for (j=0;j<5;j++) {putc(p.codigos[i].nudo[j],arch_datos);
putc(p.codigos[i].x,arch_datos);
putc(p.codigos[i].y,arch_datos);
putc(p.codigos[i].z,arch_datos);
putc('\n',arch_datos);}}
}

```

```

/* Impresion de los elementos */
for (i=0;i<general[2];i++)
{
for(j=0;j<3;j++) {putc(p.elem[i].num_elemento[j],arch_datos);
putc(p.elem[i].num_material,arch_datos);
for(k=0;k<6;k++) {for(l=0;l<3;l++) {putc(p.elem[i].nudo[k][l],arch_datos);
putc('\n',arch_datos);}}}}

/* Impresion de las condiciones de carga */
for (i=0;i<nodos_car[i]**)
{
for (j=0;j<3;j++) {putc(p.carga[i].nudo[j],arch_datos);
for (k=0;k<3;k++) {putc(p.carga[i].fx[k],arch_datos);
for (l=0;l<3;l++) {putc(p.carga[i].fy[l],arch_datos);
for (m=0;m<3;m++) {putc(p.carga[i].tz[m],arch_datos);
putc('\n',arch_datos);}}}}
fclose(arch_datos);
}
restabrece();
clear();
}

```

```

#include "graphics.h"

int driver, mode, err, color, color_max, max_x, max_y,
    x, y, i, j, ii, jj;
float aux_x, aux_y;

void graphf(int *coordenadas_x, int *coordenadas_y, int no_nodos)
{
    detectgraph(&driver, &mode);
    err = graphresult();
    initgraph(&driver, &mode, "");
    err = graphresult();
    x = getmaxx();
    y = getmaxy();
    color_max = getmaxcolor();
    setlinestyle(SOLID_LINE, 0, NORM_WIDTH);
    color = (color_max > 1) ? 1 : 0;
    setfillstyle(SOLID_FILL, color);
    setcolor(color);
    bar(1, 1, x, y);
    setcolor(color_max);

    setusercharsize(5, 2, 2, 1);
    settextstyle(TRIPLEX_FONT, HORIZ_DIR, 0);
    settextjustify(CENTER_TEXT, CENTER_TEXT);

    outtextxy(x / 2, y / 20 * 2, "G R A F I C A   D E   L O S   N O D O S");

    max_x = *coordenadas_x[0];
    max_y = *coordenadas_y[0];
    for (i = 0; i < no_nodos; i++)
    {
        if (max_x < *coordenadas_x[i])
            max_x = *coordenadas_x[i];
        if (max_y < *coordenadas_y[i])
            max_y = *coordenadas_y[i];
    }
    aux_x = 590.0 / max_x;
    aux_y = 140.0 / max_y;
    for (i = 0; i < no_nodos; i++)
        putpixel(40 * coordenadas_x[i] * aux_x + 50 * coordenadas_y[i] * aux_y, 3);
    getch();
    closegraph();
}

```

```

void graphf(int *coordenadas_x, int *coordenadas_y, int *elementos[8],
            int no_nodos, int no_elem)
{
    detectgraph(&driver, &mode);
    err = graphresult();
    initgraph(&driver, &mode, "");
    err = graphresult();
    x = getmaxx();
    y = getmaxy();
    color_max = getmaxcolor();
    setlinestyle(SOLID_LINE, 0, NORM_WIDTH);
    color = (color_max > 1) ? 1 : 0;
    setfillstyle(SOLID_FILL, color);
    setcolor(color);
    bar(1, 1, x, y);
    setcolor(color_max);

    setusercharsize(5, 2, 2, 1);
    settextstyle(TRIPLEX_FONT, HORIZ_DIR, 0);
    settextjustify(CENTER_TEXT, CENTER_TEXT);

    outtextxy(x / 2, y / 20 * 2, "G R A F I C A   D E   E L E M E N T O S");

    max_x = *coordenadas_x[0];
    max_y = *coordenadas_y[0];
    for (i = 0; i < no_nodos; i++)
    {
        if (max_x < *coordenadas_x[i])
            max_x = *coordenadas_x[i];
        if (max_y < *coordenadas_y[i])
            max_y = *coordenadas_y[i];
    }
    aux_x = 590.0 / max_x;
    aux_y = 140.0 / max_y;
    for (i = 0; i < no_nodos; i++)
        putpixel(40 * coordenadas_x[i] * aux_x + 50 * coordenadas_y[i] * aux_y, 3);
    for (i = 0; i < no_elem; i++)
    {
        for (j = i + 1; j < no_elem; j++)
        {
            ii = elementos[j] - 1;
            line(40 * coordenadas_x[i] * aux_x + 50 * coordenadas_y[i] * aux_y,
                40 * coordenadas_x[j] * aux_x + 50 * coordenadas_y[j] * aux_y);
        }
        ii = elementos[i] - 1;
        line(40 * coordenadas_x[ii] * aux_x + 50 * coordenadas_y[ii] * aux_y,
            40 * coordenadas_x[j] * aux_x + 50 * coordenadas_y[j] * aux_y);
    }

    getch();
    closegraph();
}

```

```

void cuadro_ini()
{
  clrscr();
  gotoxy(1,1);
  cputa(" ");
  cputa(" ");
  gotoxy(69,1);
  cputa("Version 1.0");
  cputa(" Analisis de placas delgadas por el metodo del");
  cputa(" elemento finito");
  cputa(" ");
  gotoxy(1,25);
  cputa(" Pantalla 1
  Maria
  Guadalupe ");
  cputa("Vera Chavez");
  caja(1,9,80,22,"");
  caja(1,22,80,24,"");
  caja(1,3,16,6,"");
  caja(1,6,16,9,"");
  caja(16,3,32,6,"");
  caja(16,6,32,9,"");
  caja(32,3,48,6,"");
  caja(32,6,48,9,"");
  caja(64,3,64,6,"");
  caja(64,6,64,9,"");
  caja(64,3,80,6,"");
  caja(64,6,80,9,"");
}

typedef enum {desc,stab,f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,larr,rarr,uarr,
             darr,home,end,pgup,pgdn,lns,del,clar,crar,alar,sear,ecac,
             norm,back,tab,retul} valfun;

void consulta()
{
  char arch_res[13],pant[4000];
  valfun func;
  int funcion;

  gettext(1,1,80,25,pant);
  caja( 3,10,30,12,"");
  caja( 3,12,38,14,"");
  gotoxy(6,11); cputa(" Archivo con los resultados ");
  invierte();
  strcpy(arch_res," ");
  gotoxy(15,13); lre(&arch_res,&func,&funcion,13);
  restablece();
  ayuda(arch_res,"11");
  puttext(1,1,80,25,pant);
}

```

```

void caja(int x1,
          int y1,
          int x2,
          int y2,
          char *patron)
{
  int i;

  gotoxy(x1,y1);
  putch(*patron);
  patron++;
  for (i=(x1+1);i<=(x2-1);i++)
  {
    gotoxy(i,y1);
    putch(*patron);
  }
  patron++;
  gotoxy(x2,y1);
  putch(*patron);
  patron++;
  for (i=(y1+1);i<=(y2-1);i++)
  {
    gotoxy(x2,i);
    putch(*patron);
  }
  patron++;
  gotoxy(x2,y2);
  putch(*patron);
  patron++;
  for (i=(x2-1);i>*(x1+1);i--)
  {
    gotoxy(i,y2);
    putch(*patron);
  }
  patron++;
  gotoxy(x1,y2);
  putch(*patron);
  patron++;
  for (i=(y2-1);i>*(y1+1);i--)
  {
    gotoxy(x1,i);
    putch(*patron);
  }
}

```

```

typedef enum {desc,stab,f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,larr,rarr,uarr,
darr,home,end,pgup,pgdn,ins,del,slar,crar,alar,sear,esca,
norm,back,tab,retu} valfun;

```

```

void lee_teclado(void fun (*func, char *ch, int *funcion))

```

```

{
    *func=desc;
    *funcion=0;
    *ch=getch();
    if (*ch == 0)
    {
        *ch=getch();
        if (*ch == 15) *func=stab;
        if ((*ch == 59) && (*ch != 68)) *funcion=1;
        if (*ch == 19) *func=f1 ;
        if (*ch == 60) *func=f2 ;
        if (*ch == 61) *func=f3 ;
        if (*ch == 62) *func=f4 ;
        if (*ch == 63) *func=f5 ;
        if (*ch == 64) *func=f6 ;
        if (*ch == 65) *func=f7 ;
        if (*ch == 66) *func=f8 ;
        if (*ch == 67) *func=f9 ;
        if (*ch == 68) *func=f10 ;
        if (*ch == 75) *func=larr;
        if (*ch == 77) *func=rarr;
        if (*ch == 72) *func=uarr;
        if (*ch == 80) *func=darr;
        if (*ch == 71) *func=home;
        if (*ch == 79) *func=end ;
        if (*ch == 73) *func=pgup;
        if (*ch == 81) *func=pgdn;
        if (*ch == 82) *func=ins ;
        if (*ch == 83) *func=del ;
        if (*ch == 115) *func=lar;
        if (*ch == 116) *func=trar;
        if (*ch == 6) *func=alar;
        if (*ch == 6) *func=sear;
    }
    else
    {
        if (*ch == 27) *func=esca;
        if (*ch == 8) *func=back;
        if (*ch == 9) *func=tab ;
        if (*ch == 13) *func=retu;
        if (*ch > 31)
        {
            *func=norm;
            if ((*ch == 'a') && (*ch != 'z')) *ch = toupper(*ch);
        }
    }
}

```

```

#include "graphics.h"

```

```

void titulo()

```

```

{
    int i,driver, mode, err, x, y,color,color_max;

```

```

    detectgraph(&driver,&mode);
    err=graphresult();
    if (err < 0) exit(0);
    initgraph(&driver,&mode,"");
    err=graphresult();
    if (err < 0) exit(0);
    x=getmaxx();
    y=getmaxy();
    setlinestyle(SOLID_LINE,0,NORM_WIDTH);
    color_max=getmaxcolor();
    color=(color_max > 1) ? 1 : 0;
    setfillstyle(SOLID_FILL, color);
    setcolor(color);
    bar(1,1,x,y);
    setcolor(getmaxcolor());

    rectangle(0,0,x,y);
    rectangle(2,2,x-2,y-2);
    rectangle(4,4,x-4,y-4);
    setusercharsize(5,2,1);
    setttextstyle(TRIPLEX_FONT,HORIZ_DIR,0);
    setttextstyle(CENTER_TEXT,CENTER_TEXT);

    outtextxy(x / 2,y / 15 * 2,"P L A C A S");
    outtextxy(x / 2,y / 10 * 4,"D E L G A D A S");
    if (color_max > 1)
        setcolor(getmaxcolor()-1);
    outtextxy(x / 2,y / 10 * 6,"Ma. Gpe. Vera Ch vez");
    i=0;
    while (!kbhit())
    {
        setcolor(i);
        rectangle(0,0,x,y);
        rectangle(2,2,x-2,y-2);
        rectangle(4,4,x-4,y-4);
        delay(500);
        i++;
        if (i > getmaxcolor())
            i = 0;
    }
    getch();
    closegraph();
}

```



```

int cin(char cadena[],int n)
{
    int i,base=1,entero=0;
    for(i = n-1;i>=0;i--)
    {
        entero+=(cadena[i]-48)*base;
        base*=10;
    }
    return(entero);
}

```

```

int cin2(char cadena[],int n)
{
    int i;
    long base=1,entero=0;
    for (i=n-1;i>=0;i--)
    {
        entero+=(cadena[i]-48)*base;
        base*=10;
    }
    return(entero);
}

```

```

float cin_real(char cadena[],int n)
{
    int i;
    long base=1;float entero=0;
    for (i=n-1;i>=0;i--)
    {
        if (cadena[i]!='.')
            entero*=1;
        else
            entero+=(cadena[i]-48)*base;
            base*=10;
    }
    return(entero);
}

```

```

void icn(long entero,char cadena[],int tamano)
{
    unsigned long int i,base=10,entero2;
    for (i=1;i<(tamano-1);i++) base*=10;
    for (i=0;i<tamano;i++)
    {
        entero2=entero / base;
        entero%=base;
        cadena[i]='0'+entero2;
        base/=10;
    }
}

```

```

void sel_funcion(char af[20][15], int numero)

```

```

{
    int i;
    char *f;
    f=af[numero*2-2];
    switch (numero)
    {
        case 1:
            gotoxy( 2,4);cputs(af[numero*2-2]);
            gotoxy( 2,5);cputs(af[numero*2-1]);
            break;
        case 2:
            gotoxy(17,4);cputs(af[numero*2-2]);putch(' ');
            gotoxy(17,5);cputs(af[numero*2-1]);putch(' ');
            break;
        case 3:
            gotoxy(33,4);cputs(af[numero*2-2]);putch(' ');
            gotoxy(33,5);cputs(af[numero*2-1]);putch(' ');
            break;
        case 4:
            gotoxy(49,4);cputs(af[numero*2-2]);putch(' ');
            gotoxy(49,5);cputs(af[numero*2-1]);putch(' ');
            break;
        case 5:
            gotoxy(65,4);cputs(af[numero*2-2]);putch(' ');
            gotoxy(65,5);cputs(af[numero*2-1]);putch(' ');
            break;
        case 6:
            gotoxy( 2,7);cputs(af[numero*2-2]);
            gotoxy( 2,8);cputs(af[numero*2-1]);
            break;
        case 7:
            gotoxy(17,7);cputs(af[numero*2-2]);putch(' ');
            gotoxy(17,8);cputs(af[numero*2-1]);putch(' ');
            break;
        case 8:
            gotoxy(33,7);cputs(af[numero*2-2]);putch(' ');
            gotoxy(33,8);cputs(af[numero*2-1]);putch(' ');
            break;
        case 9:
            gotoxy(49,7);cputs(af[numero*2-2]);putch(' ');
            gotoxy(49,8);cputs(af[numero*2-1]);putch(' ');
            break;
        case 10:
            gotoxy(65,7);cputs(af[numero*2-2]);putch(' ');
            gotoxy(65,8);cputs(af[numero*2-1]);putch(' ');
            break;
    }
}

```

```
/* Rutinas para validar la captura */
```

```
typedef enum {dese,stab,f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,larr,rarr,uarr,  
darr,hora,end,pgup,pgdn,ins,del,clar,crar,alar,aear,esca,  
norm,back,tab,retu} valfun;
```

```
void lee_xp(int long1, valfun *func, int *i, int *x)
```

```
{  
  if (*func==larr)  
  {  
    if (*i > 0)  
    {  
      *i=*i-1;  
      *x=*x-1;  
    }  
    gotoxy(*x,wherey());  
  }  
  if (*func==rarr)  
  {  
    if (*i < long1-1)  
    {  
      *i=*i+1;  
      *x=*x+1;  
    }  
    gotoxy(*x,wherey());  
  }  
}
```

```
void lee_tchar cadena2[], valfun *func, int *funcion, int tamano)
```

```
{  
  int i,x,k,guardax,guarday,ban*0,ban2*0,ban3*0;  
  char ch,cadena[5];  
  
  guardax=wherex();  
  guarday=wherey();  
  x=wherex();  
  for (i = 0; i<tamano-1;i++) {putch(cadena2[i]);cadena[i]=' ';}  
  gotoxy(x,wherey());  
  i*0;  
  lee_teclado(func, &ch, funcion);  
  while ((*func==retu && !*funcion && *func!=uarr && *func!=darr  
    && *func!=pgdn && *func!=pgup)  
  {  
    if (*func==norm && ch=='0' && ch=='9')  
    {  
      ban1*1;  
      putch(ch);  
      cadena[i]=ch;  
      if (i==tamano-1) ban2*1;
```

```
/* Continuación de las Rutinas para validar la captura */
```

```
if (i < (tamano-1))  
{  
  i++;  
  x++;  
  ban*1;  
}  
gotoxy(x,wherey());  
lee_xp(tamano,func,&i,&x);  
if (*func==back)  
{  
  if (i > 0)  
  {  
    putch(' ');  
    cadena[i]=' ';  
    i--;  
    x--;  
    ban*1;  
  }  
  gotoxy(x,wherey());  
}  
lee_teclado(func,&ch,funcion);  
if (tamano==1)  
{  
  if (ban1==1) cadena2[0]=cadena[0];  
}  
else  
if (ban==1)  
{  
  if (ban2) i--;  
  for (k*0;k<i;k++) cadena[tamano-1-k]=cadena[i-k];  
  for (k*0;k<tamano-1-i;k++) cadena[k]=' ';  
  for (k*0;k<tamano;k++) cadena2[k]=cadena[k];  
}  
gotoxy(guardax,guarday);for (k*0;k<tamano;k++) putch(cadena2[k]);
```

/* Continuación de las Rutinas para validar la captura */

```
void leerchar(cadena[], valfun *func, int *funcion, int *tamano)
```

```

{
  int i, k;
  char ch;

  x=wherex();
  for (i = 0; i < *tamano; i++) putch(cadena[i]);
  gotoxy(x, wherex());
  i=0;
  lee_teclado(func, &ch, funcion);
  while (!func || !retu || !*funcion)
  {
    if (*func == norm)
    {
      getch(&ch);
      cadena[i] = ch;
      if (i < (*tamano - 1))
      {
        i++;
        x++;
      }
      gotoxy(x, wherex());
    }
    leeexp(tamano, func, &i, &x);
    if (*func == back)
    {
      if (i > 0)
      {
        putch(' ');
        cadena[i] = ' ';
        i--;
        x--;
      }
      gotoxy(x, wherex());
    }
    lee_teclado(func, &ch, funcion);
  }
}

```

/* Continuación de las Rutinas para validar la captura */

```
void leerxp_real(valfun *func,
```

```

  int *i,
  int *x,
  int guardax,
  int tamano_entero,
  int tamano_decimal,
  char ch)

```

```

{
  if (*func == larr)
  {
    if (*i > 0)
    {
      *i = *i - 1;
      *x = *x - 1;
      if (*x == guardax + tamano_entero) ch = '.';
      gotoxy(*x, wherex());
    }
  }
  if (*func == rarr)
  {
    if (*i < tamano_entero + tamano_decimal - 1)
    {
      *i = *i + 1;
      *x = *x + 1;
      if (*x == guardax + tamano_entero) ch = '.';
      gotoxy(*x, wherex());
    }
  }
}

```

```

void leer_real(char cadena2[],
  valfun *func,
  int *funcion,
  int tamano_entero,
  int tamano_decimal)

```

```

{
  int i, j, x, k, guardax, guarday, ban = 0, tamano, menos = 0;
  char ch, cadena[20];

  guardax = wherex();
  guarday = wherex();
  x = wherex();
  tamano = tamano_entero + tamano_decimal;
  for (i = 0; i < tamano; i++) [putch(cadena2[i]); cadena[i] = ' ';
  putch(' ');
  for (i = tamano_entero; i < tamano; i++)
  {
    putch(cadena2[i]);
    cadena[i] = ' ';
  }
}

```

/* Continuacion de las Rutinas para validar la captura */

/* Rutinas de UTILERIA.C */

```

gotoxy(x,wherey());
i=0;
lee_teclado(func,&ch,function);
while ((*func!=retu) && !*funcion && *func!=uarr && *func!=darr
      && *func!=pda && *func!=pgup)
{
  if ((*func==norm && ch=='0' && ch<'9' || !ch=='-' && !menos && i==0))
  {
    getch(ch);
    cadena[i]=ch;
    if (ch=='-') monoar++;
    if (i < (tamano-1))
    {
      i++;
      x++;
      if (x==guardax+tamano_entero) x--;
      ban++;
    }
    gotoxy(x,wherey());
  }
  lee_xp_real(func,&i,&x,guardax,tamano_entero,tamano_decimal,ch);
  if ((*func==back)
    {
      if (i > 0)
      {
        getch(' ');
        cadena[i]=' ';
        i--;
        x--;
        if (x==guardax+tamano_entero) x--;
        ban++;
      }
      gotoxy(x,wherey());
    }
  lee_teclado(func,&ch,function);
}
j=0;while (cadena[j]!=' ' && j<tamano_entero) j++;
if (ban==1)
{
  j--;
  for (k=0;k<j;k++) cadena[tamano_entero-1-k]=cadena[j-k];
  for (k=0;k<tamano_entero-1-j;k++) cadena[k]=' ';
  for (k=0;k<tamano;k++) cadena2[k]=cadena[k];
}
gotoxy(guardax,guarday);
for (i=0;i<tamano_entero;i++) getch(cadena2[i]);getch(' ');
for (i=tamano_entero;i<tamano;i++) getch(cadena2[i]);

```

```

void invierte()
{
  textcolor(15);
  textbackground(1);
}

void restablece()
{
  textcolor(15);
  textbackground(1);
}

void llena_ceros(char cadena[])
{
  int i;
  for (i=0;cadena[i]!='\0';i++) cadena[i]='0';
}

void ceros_llena(char cadena[],int tamano)
{
  int i;
  for (i=tamano-1;i>=0 && cadena[i]!='\0';i--) cadena[i]='0';
}

void escribe(char cadent[i],int tamano,int x,int y)
{
  int i;
  gotoxy(x,y);for (i=0;i<tamano;i++) getch(cadent[i]);
}

void escribe_real(char cadena[],int entero,int real,int ban)
{
  int i;

  gotoxy(x,y);for (i=0;i<entero;i++) getch(cadena[i]);
  getch(' ');
  for (i=entero;i<entero+real;i++) getch(cadena[i]);
}

```

```

void despliega()
{
  clrscr();
  casa(1, 1, 80, 4, "*****");
  casa(1, 4, 80, 22, "*****");
  casa(1, 22, 80, 24, "*****");
  textcolor(11);
  textbackground(1);
  gotoxy(2, 2);
  cputs("
  PLACAS DELGADAS  \n");
  gotoxy(4, 3);
  cputs("Analisis de placas delgadas por el metodo del elemento finito");
  gotoxy(1, 25);
  cputs("
  Maria Guadalupe");
  cputs("
  ");
}

```

```

void datos_generales()
{
  gotoxy(1, 5); cputs("DATOS GENERALES");
  gotoxy(2, 8); cputs("Num. de puntos frontera:");
  gotoxy(2, 9); cputs("Num. de condiciones de carga:");
  gotoxy(2, 10); cputs("Num. de elementos finitos:");
  gotoxy(2, 11); cputs("Num. de nudos:");
  gotoxy(2, 12); cputs("Num. de materiales:");
  gotoxy(2, 13); cputs("Num. de grades de libertad:");
  gotoxy(2, 14); cputs("Num. de nodos cargados:");
  gotoxy(2, 15); cputs("Indicador de:");
  gotoxy(2, 16); cputs("Fuerzas de cuerpo:");
  gotoxy(2, 17); cputs("Calculo de esfuerzos:");
  gotoxy(2, 18); cputs("Espesor del material:");
  gotoxy(2, 19); cputs("Cuadrados o lineales (CL):");
  gotoxy(2, 23); cputs("Presione F10 para terminar ");
  restablece();
}

```

```

void materiales()
{
  gotoxy(23, 5); cputs("CAPACITRIFICACION DE LOS MATERIALES");
  gotoxy(2, 7);
  cputs("Material Mod. de elasticidad Coef. de Poisson ");
  cputs("Peso volumetrico");
  gotoxy(3, 8);
  cputs("No. (ton/m3) ");
  gotoxy(2, 23); cputs(" PgDn Pg.sig. PgUp Pg. ant. F10 Fin ");
  restablece();
  casa(2, 6, 78, 9, "*****");
}

```

```

void coordenadas()
{
  gotoxy(30, 5); cputs("DATOS DE COORDENADAS");
  gotoxy(10, 8); cputs(" NUDO X Y");
  gotoxy(2, 23); cputs(" F10 Generacion automatica PgDn Pg.sig. PgUp Pg. ant. F10 Fin ");
  restablece();
  casa(9, 7, 42, 9, "*****");
}

```

```

void generacion_automatica()
{
  restablece();
  window(1, 1, 80, 25);
  gotoxy(2, 23); cputs(" F10 Fin de generacion automatica de coord ");
  cputs("
  ");
  window(5, 8, 75, 21);
  gotoxy(6, 1); cputs("Generacion automatica");
  gotoxy(6, 2); cputs(" de coordenadas");
  gotoxy(5, 4); cputs("Punto inicial X:");
  gotoxy(5, 5); cputs("Punto inicial Y:");
  gotoxy(5, 6); cputs("Incremento en X:");
  gotoxy(5, 7); cputs("Incremento en Y:");
  gotoxy(5, 8); cputs("Nudo inicial:");
  gotoxy(5, 9); cputs("Incremento nudo:");
  gotoxy(5, 10); cputs("Num. de nudos");
  restablece();
}

```

```

void datos_de_elementos()
{
  gotoxy(29, 5); cputs("DATOS DE LOS ELEMENTOS");
  gotoxy(7, 7);
  cputs("Elemento Material Numeracion de nudo ");
  gotoxy(10, 8);
  cputs("No. 1 2 3 4 5 6 ");
  gotoxy(3, 23); cputs(" PgDn Pg.sig. PgUp Pg. ant. F10 Fin ");
  restablece();
  casa(5, 6, 76, 9, "*****");
}

```

```

void nodos_fronteras()
{
  gotoxy(30, 5); cputs("CONDICIONES DE FRONTERA");
  gotoxy(30, 8); cputs(" Nudo X Y 2");
  gotoxy(2, 23); cputs(" PgDn Pg.sig. PgUp Pg. ant. F10 Fin ");
  restablece();
  casa(29, 7, 50, 9, "*****");
}

```

```

void carga()
{
  gotoxy(30, 5); cputs("CONDICIONES DE CARGA");
  gotoxy(20, 8); cputs(" NUDO FX(FXND) FY(FYND) FZ(FZND) ");
  gotoxy(2, 23); cputs(" PgDn Pg.sig. PgUp Pg. ant. F10 Fin ");
  restablece();
  casa(29, 7, 50, 9, "*****");
}

```


0000000000

```

.....
*   P R O G R A M A   P R I N C I P A L   *
.....

```

Declaracion de Variables Globales

```

COMMON NODOS,NPG,ESP
COMMON/RISEP,P(20,24),PP(24),S(25,24),X(140,3),Y(140,3)
CHARACTER*1 NODOC
DIMENSION A(60000)
CHARACTER*12 DATOS,SALIDA,TRABAJO
WRITE(5,1000)
READ(5,2000) DATOS
WRITE(5,3000)
READ(5,2000) SALIDA
WRITE(5,4000)
READ(5,2000) TRABAJO
WRITE(5,4010)
OPEN(1,FILE=DATOS ,STATUS='OLD',FORM='FORMATTED')
OPEN(2,FILE=SALIDA ,STATUS='NEW',RECL=132)
OPEN(3,FILE=TRABAJO,STATUS='NEW')
WRITE(2,6010)
READ(1,3000)NMC,IFC,ICE,NODOC,ESP,NPF,NCC,NEL,NPE,NMA,NGL
NPG=3
IF(NODOC.EQ.'C')NODOS=8
IF(NODOC.EQ.'L')NODOS=4
IF(NODOS.EQ.8)WRITE(2,6030)
IF(NODOS.EQ.4)WRITE(2,6020)
WRITE(2,6000)NEL,NPE,NMA,NGL,NMC,NPF,ESP
WRITE(4,*)IESP
K1=1
K2=K1+NGL*NPE
K3=K2+NPE
K4=K3+NPE
K5=K4+NMA
K6=K5+NMA
K7=K6+NMA

```

```

          C      A      L      L
ELICUA(NPE,A(K1),A(K2),A(K3),A(K4),A(K5),A(K6),NEL,NMA,NEL,
INEP,NGL,NPF)
K3=K2+NEC+NEC
K4=K3+NEC
CALL MATEST(NEL,A(K2),A(K3),NEC,NGL)
K5=K4+NEC
          C      A      L      L
DESESP(NEC,A(K1),A(K2),A(K3),A(K4),NPE,NEL,NGL,NCC,IFC,ICE,
INMC)

```

CONTINUACION DEL PROGRAMA PRINCIPAL

```

00  FORMAT(////26X,'ANALISIS DE PLACAS DELGADAS'//
133X,'POR MEDIO DEL'//27X,'METODO DEL ELEMENTO FINITO'////)
2: ARCHIVO DE DATOS 1')
00  FORMAT(A12)
00  FORMAT(/// ARCHIVO DE SALIDA 1')
00  FORMAT(/// ARCHIVO DE TRABAJO: ')
10  FORMAT(////10X,'PROCESANDO...')
00  FORMAT(15,211,A1,F7.5,615)
00  FORMAT(20X,'NUM. DE ELEMENTOS FINITOS      NEL='//13//
120X,'NUM. DE NUDOS',16X,'NPE='//13//
220X,'NUM. DE MATERIALES',11X,'NMA='//13//
320X,'NUM. DE GRADOS DE',12X,'NGL='//13//
420X,'LIBERTAD POR NUDO'//
520X,'NUM. DE NUDOS CARGADOS',7X,'NMC='//15//
620X,'NUM. DE PUNTOS FRONTERA',6X,'NPF='//15//
720X,'ESPESOR DEL MATERIAL',7X,'ESP='//F7.5//)
110  FORMAT(21X,'DATOS GENERALES DE LA ESTRUCTURA')
120  FORMAT(//21X,'ELEMENTOS CUADRILATEROS LINEALES',/)
130  FORMAT(//20X,'ELEMENTOS CUADRILATEROS CUADRICOS',,)
      END

```


00000000

.....
 * TRIANGULACION POR EL METODO DE CHOLESKY *


```

SUBROUTINE TRICHO(A,N)
DIMENSION A(N,N)
A(1,1)=SQRT(A(1,1))
A(2,2)=SQRT(A(2,2)-A(1,2)**2)
A(1,2)=A(1,2)/A(1,1)
IF (N.EQ.2) RETURN
DO 800 J=3,N
  A(1,J)=A(1,J)/A(1,1)
  JM1=J-1
  DO 500 I=2,JM1
    IM1=I-1
    XX=0.
    DO 200 K=1,IM1
      XY=XX+A(K,1)*A(K,J)
200 CONTINUE
    A(I,J)=(A(I,J)-XX)/A(1,1)
500 CONTINUE
    XX=0.
    DO 700 K=1,JM1
      XZ=XX+A(K,J)*A(K,J)
700 CONTINUE
    A(J,J)=SQRT(A(J,J)-XZ)
800 CONTINUE
RETURN
END
  
```

.....
 * SUSTITUCION POR EL METODO DE CHOLESKY *


```

SUBROUTINE SUSCH3(A,B,M)
DIMENSION A(N,N),B(N)
B(1)=B(1)/A(1,1)
DO 400 I=2,N
  IM1=I-1
  XX=0.
  DO 200 K=1,IM1
    XX=XX+A(K,1)*B(K)
    B(I)=(B(I)-XX)/A(1,1)
400 CONTINUE
HACIA ATRAS
B(N)=B(N)/A(N,N)
NM1=N-1
DO 600 M=1,NM1
  I=N-M
  JP1=I+1
  XZ=0.
  DO 600 K=IP1,N
    XX=XX+A(I,K)*B(K)
600 CONTINUE
    B(I)=(B(I)-XX)/A(1,1)
800 CONTINUE
RETURN
END
  
```

00000000

COORDENADAS Y CODIGOS DE FRONTERA

```

SUBROUTINE COCOF(NPE, X, Y, ID, NEC, NEP, NGL, NPF)
DIMENSION X(NPE), Y(NPE), ID(NGL, NPE)
DO 400 I=1, NPE
400 READ(1, 5000) X(I), Y(I)
DO 405 I=1, NPF
405 READ(1, 5010) N, (ID(IL, N), IL=1, NGL)
WRITE(2, 6040)
DO 420 N=1, NPE
420 WRITE(2, 6050) N, X(N), Y(N), (ID(IL, N), IL=1, NGL)
NEC=0
NEP=0
NGL=0
DO 800 I=1, NPE
DO 600 K=1, NGL
IF (ID(K, I) = 1) 450, 500, 550
450 CONTINUE
NEC=NEC+1
ID(K, I)=NEC
GO TO 600
500 CONTINUE
NEP=NEP+1
ID(K, I)=0
GO TO 600
550 CONTINUE
NEP=NEP-1
ID(K, I)=-NEP
600 CONTINUE
800 CONTINUE
RETURN
5000 FORMAT(2F7.2)
5010 FORMAT(15, 311)
6040 FORMAT(11, 16X, 'COORDENADAS', 4X, 'CODIGOS DE //
110X, 'NUDDO X(N) Y(N)', 5X, 'FRONTERA')
6050 FORMAT(10X, 15, 2F7.2, 2X, 315)
END

```

00000000

CALCULO DEL VECTCH DE CARGAS
DESPLAZAMIENTOS Y ESFUERZOS DE LAS ESTRUCTURA

```

SUBROUTINE DEBESF(NEC, ID, CK, PE, P, NPE, NEL, NGL, NCC, IFC, ICD, M)
DIMENSION ID(NGL, NPE), EK(NEC, NEC), P(NEC), PE(NEC)
COMMON NUDDOS, NPG, ESP
COMMON/RIGEP/R(24, 24), PP(24), S(25, 24), A(140, 5), Y(140, 5)
IF (NCC.LE.0) NCC=1
DO 600 N=1, NCC
WRITE(2, 6030) N
WRITE(2, 6000) N=NC
DO 200 I=1, NCC
P(I)=0.0
200 CONTINUE
IF (NCC.EQ.0) GO TO 300
CALL FUENCU(NC, ID, P, NPE, NEC, NGL)
300 CONTINUE
IF (IFC.EQ.0) GO TO 500
DO 400 I=1, NEC
P(I)=P(I)+PE(I)
400 CONTINUE
500 CONTINUE
IF (IFC.EQ.0) WRITE(2, 6010)
IF (IFC.NE.0) WRITE(2, 6020)
CALL SUSCHD(EK, P, NEC)
CALL ESCDE(ID, P, NPE, NEC, NGL)
IF (ICE.NE.0) CALL ESFEL(NEL, P, NEC, NGL)
600 CONTINUE
RETURN
5000 FORMAT(315)
6000 FORMAT(11, 10X, 'NUMERO DE NUDDOS CARGADOS', 15//)
6010 FORMAT(11, 10X, 'NO SE CONSIDERAN LAS CARGAS POR //
112X, 'PESO PROPIO DE LA ESTRUCTURA //)
6020 FORMAT(11, 10X, 'SE CONSIDERAN LAS CARGAS POR //
112X, 'PESO PROPIO DE LA ESTRUCTURA //)
6030 FORMAT(11, 10X, 'CONDICION DE CARGA NO.', 15)
RETURN
END

```

CCCCC

```

.....
*
*           SUBROUTINA ELICUA
*
.....

```

```

SUBROUTINE ELICUA(NPE, ID, X, Y, E, XNU, PV, NEL, NMA, NEC, NEP, NGL, NPF)
DIMENSION ID(NGL, NPE), X(NPE), Y(NPE), E(NMA), XNU(NMA), PV(NMA)
DIMENSION NOE(8), XE(8), YE(8), D(3,3), DS(2,2)
DIMENSION IE(24)
INTEGER*4 K, KE, KMA, NOE
REAL*4 E, XNU, PV
COMMON NODOS, NPG, ESP
COMMON/RIGEP/R(24,24), PP(24), S(25,24), XG(40,5), YG(40,5)
WRITE(2,6040)
WRITE(2,6020)
DO 250 J=1, NMA
  READ(1,5000)K, E(K), XNU(K), PV(K)
  WRITE(2,6000)K, E(K), XNU(K), PV(K)
  CALL CODCOF(NPE, X, Y, ID, NEC, NEP, NGL, NPF)
  WRITE(2,6030)(I, I=1, NODOS)
  WRITE(2,6050)
  WRITE(*,*) NODOS
  DO 800 N=1, NEL
    READ(1,5010)K, KMA, (NOE(I), I=1, NODOS)
    CALL ECOMA(E(KMA), XNU(KMA), D, DS, NGL)
    WRITE(2,6010)K, KMA, ESP, (NOE(I), I=1, NODOS)
    DO 300 I=1, NODOS
      X=NOE(I)
      J=2*(I-1)
      XE(I)=X(K)
      YE(I)=Y(K)
      DO 300 IL=1, NGL
        IF(NGL-IL+1)J=(I-1)+(IL-1)*7
        IE(IJ)=ID(IL, X)
      CONTINUE
      CALL RICAES(XE, YE, D, DS, PV(KMA), ESP, NGL, N)
      CALL GUADIS(IE)
    CONTINUE
  RETURN
5000 FORMAT(15,F9.2,2F5.4)
5010 FORMAT(13,11,8I3)
4000 FORMAT(11X,13,B1,F12.2,14,2(11X,F7.4))
6010 FORMAT(2110,F12.2,21,5I5)
6020 FORMAT(///10X,'MATERIAL MOD. DE ELASTICIDAD COEF. DE POISSON',
1' PESO VOLUMETRICCO'/12X,'NO.',10X,'(TON/M2)',30X,'(TON/M3)')
6030 FORMAT(///5X,'ELEMENTO MATERIAL ESPESOR NUMERACION',
1' DE NUDOS'/7X,21('NO.'), '(M)', B15)
6040 FORMAT(//34X,'CATACTERISTICAS DE LOS MATERIALES')
6050 FORMAT(//)
RETURN
END

```

CCCCCCCC

```

.....
*
*           CALCULO DE LOS ESFUERZOS PRINCIPALES
*           THETA ES EL ANGULO, MEDIDO EN EL SENTIDO
*           HORARIO, ENTRE LA DIRECCION X Y LA DIRECCION
*           DEL ESFUERZO PRINCIPAL MAYOR
*
.....

```

```

SUBROUTINE ESFPRI(SIG,SIP,THETA)
DIMENSION SIG(3),SIP(3)
X1=SIG(1)
X2=SIG(2)
X3=SIG(3)
P1=3.1415926535
IF (X1.NE.X2,OR.XX.NE.O.O) GO TO B1
THETA=0.O
SIP(1)=X1
SIP(2)=X1
SIP(3)=O.O
RETURN
B1 IF (X1.NE.X2) GO TO B2
THETA=45.O
IF (XX.LT.O.O) THETA=-THETA
SIP(1)=X1+XX
SIP(2)=X1-XX
SIP(3)=XX
RETURN
B2 C=(SIG(1)+SIG(2))/2.O
R=SQRT((SIG(1)-C)**2+SIG(3)**2)
TBDT=SIG(3)/(SIG(1)-C)
ALFA=ATAN(TBDT)
IF (ALFA) 7,B,9
B IF (X1.GT.X2) GO TO 25
THETA=90.O
GO TO 109
25 THETA=0.O
GO TO 109
7 IF (XX.LT.O.O) GO TO 26
THETA=(180.O+ALFA+P1)/2.O/P1
GO TO 109
26 THETA=180.O+ALFA/2.O/P1
GO TO 109
9 IF (XX.GT.O.O) GO TO 26
THETA=(180.O+(ALFA-P1))/2.O/P1
109 SIP(1)=C-R
SIP(2)=C+R
SIP(3)=R
RETURN
END

```

FORMACION DE LA MATRIZ D DE COEFICIENTES ELASTICOS PARA
EL CASO DEL ESTADO PLANO DE ESFUERZOS

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

donde:

E = e = Elasticidad
V = xnu = Coeficiente de poisson

```

SUBROUTINE ECOMA(E,XNU,D,DS,NGL)
DIMENSION D(3,3),DS(2,2)
DO 20 I=1,3
DO 20 J=1,3
D(I,J)=0.0
20 D(1,1)=E/(1.0-XNU**2)
D(1,2)=D(1,1)*XNU
D(2,1)=D(1,2)
D(3,3)=1.0-XNU)*D(1,1)/2.0
D(2,2)=D(1,1)
IF (NGL.EQ.2) RETURN
DS(1,1)=D(3,3)
DS(2,2)=D(3,3)
DS(1,2)=0.0
DS(2,1)=0.0
RETURN
END

```

MATRIZ DE LA ESTRUCTURA

```

SUBROUTINE MATEST(NEL,EK,PE,NEC,NGL)
DIMENSION EK(NEC,NEC),PE(NEC),IE(24)
COMMON NODDS,NPS,EGP
COMMON/RIGEP/R(24,24),PP(24),S(25,24),IG(40,3),YG(40,5)
N2=NGL*NODDS
REWIND 3
DO 40 M=1,NEC
PE(M)=0.
DO 40 N=1,NEC
40 EK(M,N)=0.
DO 800 K=1,NEL
CALL LEEDIS(IE)
DO 600 I=1,N2
IK=IE(I)
IF(IE(L.E.0) GO TO 600
DO 400 J=1,N2
JK=IE(J)
IF(JK.L.E.0) GO TO 400
EK(IY,JK)=EK(IK,JK)+R(I,J)
400 CONTINUE
PE(IK)=PE(IK)+PP(I)
600 CONTINUE
800 CONTINUE
CALL TRICHO(EK,NEC)
RETURN
END

```


C
C
C
C
C
C
C

```

.....
*
*
*   CALCULO DE ESFUERZOS
*
*
.....

```

```

SUBROUTINE ESFEL(NCL,P,NEC,NGL)
DIMENSION UE(24),SE(5,24),SIG(3),SIP(1),P(NEC),IE(24)
COMMON NODDS,NPG,ESP
COMMON/RIGEP/R(24,24),PR(24),S(25,24),XG(40,5),YG(40,5)
N2=NGL*NODDS
N3=2*NGL-1
REWIND 3
IF (NGL.EQ.2) WRITE(2,6010)
IF (NGL.EQ.3) WRITE(2,6020)
DO 600 I=1,NEL
CALL LEED15(IE)
DO 300 J=1,N2
P=IE(I)
IF (N.LC.0) GO TO 200
UC(I)=P(X)
GO TO 300
200 CONTINUE
UE(I)=0.0
300 CONTINUE
DO 400 NN=1,NPG
MM=N3+(NN-1)
DO 350 I=1,N3
P=MM+I
DO 350 J=1,N2
SE(I,J)=S(P,J)
350 CONTINUE
CALL MUMAT(SE,UE,SIG,N3,N2,1)
IF (NGL.GT.2) GO TO 360
CALL ESFPR1(SIG,SIP,THETA)
WRITE(2,6000)N,NN,XG(N,NN),YG(N,NN),SIG,SIP,THETA
GO TO 400
360 WRITE(2,6000)N,NN,XG(N,NN),YG(N,NN),SIG
400 CONTINUE
600 CONTINUE
RETURN
6000 FORMAT(1X,2I4,2X,2F7.2,1X,3F12.2,2X,4F12.2)
6010 FORMAT(//////,4X,'E SFUERZOS CALCULADOS EN',
1' (CONV2) '//10X,'ELEMENTOS COORDENADAS',10X,' REFERENCIA',
2' GLOBAL',10X,' REFERENCI PRINCIPAL',7X,' DIRECCION',10X,
3'NO. PUNTO X(M) Y(M) NORMAL SXX NORMAL SYY CORTANTE SXY',
4' MAXIMO SI MINIMO S2 CORT. MAX. T (GRADOS)')
6020 FORMAT(//////,11X,'ELEMENTOS MECANICOS (UNIDADES
1TON, M) '//10X,'ELEM=PUNTO',4X,'COORDENADAS * M O M E N
2 T D S * CORTANTES *//10X,'NO.',10X,'X(M)',3X,'Y
3(M)',6X,'MX',10X,'MY',10X,'MXY',10X,'GX',10X,'DY')
END

```

C
C
C
C
C
C
C

```

.....
*
*   CALCULO DE LAS FUNCIONES DE FORMA, B Y JACOBIANO
*
*
.....

```

C
C
C
C
C

```

SUBROUTINE MABNJ(XS,ETA,XN,B,XJ,X,Y)
DIMENSION XN(4),B(3,8),X(4),Y(4),XN(4),XNE(4),XJ(4),YJ(4)
NN=4
CALCULO DE LAS FUNCIONES DE FORMA

```

C
C
C
C
C
C
C

```

XIP=1.0+XS1
XIM=1.0-XS1
ETP=1.0-ETA
ETM=1.0+ETA
XN(1)=XIP*ETP/4.0
XN(2)=-XIM*ETP/4.0
XN(3)=XIP*ETM/4.0
XN(4)=-XIM*ETM/4.0

```

C
C
C
C

```

DERIVADAS DE FUNCIONES DE FORMA LOCAL

```

C
C
C
C
C
C
C

```

XNE(1)=ETP/4.0
XNY(2)=-XNX(1)
XNX(3)=-ETM/4.0
XNX(4)=-XNX(3)
XNE(1)=XIP/4.0
XNE(2)=-XIM/4.0
XNE(3)=-XNE(2)
XNE(4)=-XNE(3)

```

C
C
C

```

DERIVADAS DE COORDENADAS GLOBALES RESPECTO A LAS LOCALES

```

C
C
C
C
C
C
C

```

XXI=0.0
XET=0.0
YXI=0.0
YET=0.0
DO 10 I=1,NN
XJ=IXN(I)+XJ(1)+XXI
XET=XET+XNE(I)*X(I)
YJ=YXI+XNX(I)*Y(I)
YET=YET+XNE(I)*Y(I)
J=XJ+YET-XET+YXI
IF (XJ.EQ.0) XJ=1.
XJ=XJ/XJ
XET=XET/XJ
YJ=YJ/XJ
YET=YET/XJ

```

10
C
C
C
C
C
C
C

```

DO 500 I=1,NN
XN(I)=YET*XN(I)-YXI*XNE(I)
YJN(I)=-XET*XN(I)+XXI*XNE(I)
DO 600 K=1,NN
J=2*K
I=J-1
B(I,I)=XN(I)
B(I,J)=YJN(I)
B(J,I)=YXI
B(J,J)=XN(I)

```

500
C
C
C
C
C
C
C

```

RETURN
END

```

IMPRESION DE LOS DESPLAZAMIENTOS

```

SUBROUTINE ESCDES(ID,P,NPE,NEC,NGL)
DIMENSION ID(NGL,NPE),P(NEC)
IF (NGL.EQ.2) WRITE(2,6000)
IF (NGL.EQ.3) WRITE(2,6020)
DO 300 NUDD=1,NPE
  DX=0.
  DY=0.
  DZ=0.
  I=ID(1,NUDD)
  IF (I.LE.0) GO TO 100
  DX=P(I)
100 CONTINUE
  J=ID(2,NUDD)
  IF (J.LE.0) GO TO 200
  DY=P(J)
200 CONTINUE
  K=ID(3,NUDD)
  DZ=P(K)
250 CONTINUE
  WRITE(2,6010) NUDD, DX, DY, DZ
300 CONTINUE
  RETURN
6000 FORMAT(///10X,'DESPLAZAMIENTOS DE LOS NUDOS'//
110X,'NUDD  DESP. X (M)  DESP. Y (M)')
6010 FORMAT(10X,13,6(4X,1PG12.4))
6020 FORMAT(///10X,'DESPLAZAMIENTOS DE LOS NUDOS'//10X,'NUDD
GIRO X (M)  GIRO Y (M)  DESP. Z (M)')
END

```

CARGAS CONCENTRADAS

```

SUBROUTINE FUERCO (NMC, ID, P, NPE, NEC, NGL)
DIMENSION ID(NGL,NPE),P(NEC)
WRITE(2,6010)
DO 600 I=1,NMC
  READ(1,5000)IN,FX,FY,FZ
200 CONTINUE
  WRITE(2,6000)IN,FX,FY,FZ
  K=ID(1,IN)
  IF (K.LE.0) GO TO 300
  P(K)=P(K)+FX
300 CONTINUE
  K=ID(2,IN)
  IF (NGL.EQ.2) GO TO 400
  P(K)=P(K)+FY
400 CONTINUE
  K=ID(3,IN)
  IF (K.LE.0) GO TO 500
  P(K)=P(K)+FZ
500 CONTINUE
  450 IN=IN+1
  600 CONTINUE
  RETURN
5000 FORMAT(15,3F5.3)
6000 FORMAT(10X,13,6F10.3)
6010 FORMAT(10X,'NUDD',7X,'FX (TON)',8X,'FY (TON)',8X,'FZ (TON)'
END

```

C
C
C
C
C
C
C

```
.....  
*  
*           MULTIPLICACION DE MATRICES           *  
*  
.....
```

30

```
SUBROUTINE MMAT(A,B,C,NA,MA,MB)  
DIMENSION A(NA,MA),B(MA,MB),C(NA,MB)  
DO 30 J=1,MB  
DO 30 I=1,NA  
C(I,J)=0.0  
DO 30 IN=1,MA  
C(I,J)=A(I,IN)*B(IN,J)+C(I,J)  
RETURN  
END
```

C
C
C
C
C
C
C

```
.....  
*  
*           TRANSPUESTA DE UNA MATRIZ           *  
*  
.....
```

20

```
SUBROUTINE TMMAT(A,NA,MA)  
DIMENSION A(NA,MA),AT(MA,NA)  
DO 20 J=1,MA  
DO 20 I=1,NA  
AT(J,I)=A(I,J)  
RETURN  
END
```

C
C
C
C
C
C
C

```
.....  
*  
*           GUARDAR EN DISCO EL VECTOR IE       *  
*  
.....
```

5000

```
SUBROUTINE GUADIC(IE)  
DIMENSION IE(24)  
WRITE(3,5000) IE(MM),MM=1,24  
FORMAT(24:5)  
RETURN  
END
```

C
C
C
C
C
C
C

```
.....  
*  
*           LEER DEL DISCO EL VECTOR IE        *  
*  
.....
```

5000

```
SUBROUTINE LEEDIS(IE)  
DIMENSION IE(24)  
READ(3,5000) IE(MM),MM=1,24  
FORMAT(24:5)  
RETURN  
END
```

BIBLIOGRAFIA

- [1] Borland International
"Turbo C, Reference Guide"
Version 2.0
Ed. Borland International, 1988.
- [2] Cervantes, R., Porrás, V.
"Elemento Finito"
U.P.A.E.P., 1980
- [3] Chan S. Park
"Interactive microcomputer graphics"
Ed. Addison-Wesley, 1985
Publishing Company
- [4] Donald Hearn
"Gráficas por computadora"
Ed. Prentice Hall, 1987
- [5] Luthé, Rodolfo
"Análisis estructural"
Ed. Representaciones y servicios de ingeniería, 1971
- [6] Popov, P. Egor
"Introducción a la mecánica de sólidos"
Ed. Limusa, 1982
- [7] Timoshenko S. y Woinowsky-Krieger S.
"Theory of plates and shells"
Segunda edición
Ed. McGraw-Hill, 1959
- [8] Zienkiewicz, O.
"El método de los elementos finitos"
Ed. Reverte, S. A., 1980