

0 1173 5
2g



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

DIVISION DE ESTUDIOS DE POSGRADO

**MODELACION NUMERICA DE LA EVOLUCION
DINAMICA DE ESFUERZOS EN SOLIDOS.**

T E S I S

QUE PARA OBTENER EL GRADO DE

MAESTRO EN INGENIERIA

(M E C A N I C A)

P R E S E N T A :

LUCIANO VELA MARTINEZ

ASESOR: DR. JORGE CARRERA BOLAÑOS



MEXICO, D. F.

FEBRERO 1999

**TESIS CON
FALLA DE ORIGEN**

275740



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

PAGINACION

DISCONTINUA.

Reflexiones sobre el ser idealista.

"Cuando pones la proa visionaria hacia una estrella y tiendes el ala hacia tal excelsitud inasible, afanoso de perfección y rebelde a la mediocridad, llevas en ti el resorte misterioso de un ideal."

"El concepto de lo mejor es un resultado natural de la evolución misma. La vida tiende naturalmente a perfeccionarse."

"El libre albedrío es un error útil para la gestación de los ideales."

"Todo ideal... puede contener una parte de error, o serlo totalmente... lo único malo es carecer de ideales y esclavizarse a las contingencias de la vida práctica inmediata, renunciando a la posibilidad de la perfección moral."

"Todo idealista es un hombre cualitativo: posee un sentido de las diferencias que le permite distinguir entre lo malo que observa, y lo mejor que imagina."

"El ideal es un sueño susceptible de convertirse en realidad, es algo que todavía no es pero que puede llegar a ser, [es un gesto de espíritu hacia alguna perfección... no es una verdad pero es una creencia]. El idealista es quien sueña y ambiciona; el hombre de éxito es el que convierte sus sueños y ambiciones en realidad. Sólo un idealista puede ser un hombre de éxito."

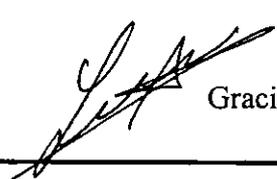
"Ser, hacer y tener son las estrellas brillantes de la grandeza, pero la más brillante es ser. El mejor ideal es buscar un alma justa, una razón recta y un corazón limpio."

"Sé tú mismo, desarrolla tu carácter... [y tu individualidad. Conquista tu espíritu.] Eres único e irrepetible... No hay otro ser igual en ningún otro lugar en este planeta... Este carácter singular de tu individualidad es la base para calcular el valor de tu persona, y también es la base para comprender el carácter particular de tus problemas. Nadie afronta los problemas que la vida ofrece en la forma en que tú lo haces, porque nadie es igual a ti."

Con estos fragmentos de José Ingenieros y Félix Cortés, escritores de la juventud, te invito a reflexionar y luchar por el máximo ideal: fuimos creados a imagen y semejanza de Dios, símbolo de justicia, amor y perfección; somos esa realidad perfectible que se acerca a su límite tanto como nosotros queramos pero sin alcanzarlo. Así pues, establezcámonos un ideal elevado, concentremos todas nuestras fuerzas y luchemos no importa cuánto tiempo para alcanzarlo. Sea cual fuere el resultado, al final habremos enriquecido nuestra vida, sólo así descubriremos de qué somos capaces, y lo más importante: no habremos vivido en vano.

Siempre valdrá la pena, aunque sea por un único instante, ver el brillo de sus ojos, oler su perfume, sentir su calor, tocar su piel...

A través de estas líneas quiero reconocer y agradecer a quienes con su apoyo, me han impulsado para seguir siempre hacia adelante en mi formación como hombre: a nuestro Padre Eterno, a mi familia, a mis maestros, a mis amigos, y a mis compañeros. A todos ustedes


Gracias.

Índice.

Índice	pág.
Introducción	i
Conceptos básicos	iv
	vii

Capítulo 1: Fundamentos físico - matemáticos

A. ELASTICIDAD

1.1 Cinemática	1
<i>Cuerpos y deformaciones. Deformaciones pequeñas. Movimientos.</i>	
1.2 Masa y momento	4
<i>Conservación de masa. Momentum lineal y angular.</i>	
1.3 Fuerza	5
<i>Sistemas de fuerzas. Teorema de existencia del campo tensorial de esfuerzo de Cauchy.</i>	
1.4 Elasticidad finita	7
<i>Cuerpos elásticos, isotrópicos y homogéneos. El campo tensorial de esfuerzo de Piola-Kirchhoff. El campo tensorial de elasticidad.</i>	
1.5 Elasticidad lineal	10
<i>Derivación de la teoría lineal. Elastoestática lineal. El problema asociado de elastoestática. Elastodinámica lineal. El problema asociado de elastodinámica.</i>	

B. MATEMÁTICAS

1.6 Teoría básica de distribuciones	14
1.7 Ecuaciones diferenciales lineales	15
<i>Definición. Condiciones de frontera. Soluciones fuertes y débiles de ecuaciones diferenciales lineales.</i>	
1.8 Teoría de convergencia en espacios con métrica	17
<i>Espacios completos. Espacios de Banach. Contracciones.</i>	
1.9 Aproximaciones e interpolación	20

Capítulo 2: Definición del método de elemento finito

2.1 Mallas de elemento finito	22
<i>Discretizaciones. Elemento finito de referencia.</i>	
2.2 Grados de libertad y funciones de interpolación	24
2.3 El problema asociado del método de elemento finito	25
2.4 Planteamientos teóricos generales para la aplicación del método de elemento finito	25

	<i>Método de Ritz. Método de Galerkin.</i>	
2.5	Definición formal del método de elemento finito	27
Capítulo 3: Planteamiento del problema de elasticidad lineal		
3.1	Problema de elastoestática lineal bidimensional	28
	<i>Definición. Formulación fuerte. Formulación variacional. Formulación de elemento finito. Modelo numérico equivalente.</i>	
3.2	Problema de elastodinámica lineal bidimensional	34
	<i>Aproximaciones en el tiempo. Definición. Formulación fuerte. Formulación variacional. Formulación de elemento finito. Modelo numérico equivalente. Esquemas de aproximación para la derivada temporal.</i>	
Capítulo 4: Caso de estudio: vibración de una viga		
4.1	Definición del problema	41
	<i>Planteamiento. Consideraciones. Discretización del dominio.</i>	
4.2	Análisis de elemento finito	43
	<i>Funciones locales de interpolación. Funciones de deformación. Funciones globales de interpolación.</i>	
4.3	Modelos numéricos de elemento finito	48
	<i>Modelo de la matriz K. Modelo de la matriz M. Propiedades de las matrices de elemento finito. Modelo del vector $f(t)$. Integración numérica.</i>	
4.4	Discretización en el tiempo	52
	<i>Frecuencia natural de vibración. Estabilidad del esquema de aproximación.</i>	
4.5	Solución del modelo numérico equivalente	54
	<i>Imposición de restricciones en los desplazamientos nodales. Recursión. Algoritmo de solución.</i>	
4.6	Manejo de los desplazamientos	57
	<i>Interpolación de los desplazamientos. Deformaciones, esfuerzos y esfuerzos principales. Criterios de falla.</i>	
Capítulo 5: Algoritmos de programación		
5.1	Introducción	61
5.2	Pre-procesador	61
5.3	Procesador	62
5.4	Post-procesador	63
5.5	Diagrama de bloques	63
5.6	Algoritmos de programación	69
	<i>Pre-procesador, procesador, post-procesador.</i>	
5.7	Ejemplo de aplicación: deformación de una viga.	72
	<i>Solución numérica, método de elemento finito. Comparación de soluciones y verificación de resultados.</i>	

Conclusiones	81
Bibliografía	83
Apéndices	
A.1 Deformaciones en coordenadas cartesianas	A-1
A.2 Esquemas de aproximación para la derivada temporal	A-6
A.3 Integración numérica sobre dominios triangulares	A-7
A.4 Solución de ecuaciones lineales simultáneas	A-10
A.5 Soluciones analíticas	A-13
A.6 Métodos aproximados de solución	A-16
A.7 Ejemplos de discretización	A-17
A.8 Listado de programas	A-21

Introducción.

En este trabajo se define el método de elemento finito como una técnica de interpolación para aproximar numéricamente a la solución de un problema en ecuaciones diferenciales parciales de segundo orden. Con esta definición, la aplicación del método de elemento finito se vuelve un proceso sistematizado que permite construir una serie de modelos equivalentes al problema original, hasta concluir con un modelo numérico representado por un sistema de ecuaciones algebraicas lineales. La solución del modelo numérico es, de alguna forma, una representación de la solución al problema en ecuaciones diferenciales parciales original.

Los objetivos generales de este trabajo son: i) presentar una deducción de las ecuaciones básicas que rigen la teoría linealizada de la elasticidad dinámica, ii) establecer los fundamentos matemáticos necesarios para presentar al método de elemento finito como la técnica de solución numérica de ecuaciones diferenciales más adecuada a la formulación física anterior, y como objetivo concreto, iii) generar un modelo numérico de la teoría linealizada de la elasticidad dinámica que pueda ser resuelto por el método de elemento finito. Los marcos teóricos empleados en este trabajo son la mecánica del medio continuo, álgebra lineal, análisis funcional y análisis tensorial.

Con respecto al primer objetivo, es de mucha importancia señalar que con las presentaciones axiomáticas de la mecánica del medio continuo, aún en presentaciones tan buenas como las de Gurtin [9], los modelos concretos para casos específicos deben deducirse por el lector, es decir, en general no se dan esos modelos concretos sino que hay que elaborarlos. También cabe señalar que en libros más prácticos sí se ofrecen los modelos concretos, como es el caso de [6], [16], [21] y [22], entre otros, pero no se da su deducción del marco teórico más general. Por lo tanto, la vinculación de la presentación axiomática de la mecánica del medio continuo con el cumplimiento de este primer objetivo es considerado una contribución importante de esta tesis.

El modelo numérico generado en el tercer objetivo representa la contribución más interesante de este trabajo, pues no sólo es un modelo perfectamente basado en la teoría general sino que no es idéntico a otros modelos que se encuentran en la bibliografía. Como ejercicio interesante de aplicación, este modelo es desarrollado en esta tesis casi en su totalidad por el autor, desde la formulación numérica, la discretización y la implementación computacional, siguiendo el esquema teórico de modelación matemática presentado en [3] y representa, por lo tanto, un esfuerzo en la dirección de los conceptos más actuales de la modelación matemática de fenómenos físicos.

En el capítulo 1 se presentan los fundamentos físicos y matemáticos necesarios para el desarrollo de este trabajo. En una primera parte, bajo la hipótesis y el marco teórico de los medios continuos

se establece parte de la cinemática con la definición de cuerpos, deformaciones y movimientos, la cual será útil para establecer los aspectos dinámicos más adelante. Luego, con base en axiomas básicos de la física como son conservación de la masa y las leyes de balance de momento, los conceptos de fuerza y esfuerzo permiten establecer uno de los teoremas centrales de este trabajo: la existencia del campo tensorial de esfuerzo de Cauchy, y con éste la ecuación de movimiento. Las consideraciones constitutivas, que definen las propiedades físicas de los cuerpos, y que en este caso están representadas por los módulos de Lamé, acotan el campo de trabajo a una clase específica de cuerpos: los cuerpos elásticos. Con tales consideraciones y un cambio de coordenadas conveniente de la ecuación de movimiento, por medio del campo tensorial de esfuerzo de Piola-Kirchhoff, se llegan a establecer finalmente las ecuaciones de campo de la teoría de la elasticidad dinámica, así como los criterios para su linealización. Los conceptos manejados aquí, con los cuales se cumple el primer objetivo, son tomados básicamente de [9] y comparados para su verificación con los desarrollos clásicos [6,], [16], [21] y [22], basados en la mecánica newtoniana.

En la segunda parte del capítulo 1 se establecen los fundamentos matemáticos del método de elemento finito. La inclusión de las distribuciones en este trabajo se debe a que en general, la respuesta a un sistema de fuerzas sobre un cuerpo cualquiera, más que ser una función es una distribución. Luego, puesto que el propósito de la matemática numérica es construir mediante funciones conocidas, la mejor aproximación a la distribución solución de una ecuación diferencial, es importante el manejo de otros conceptos tales como convergencia, sucesiones, contracciones, interpolación, soluciones aproximadas, espacios densos, entre otros. Como ejemplo, el concepto de espacio denso tiene una importancia fundamental en el sentido de la aproximación a la solución: aún cuando se esté trabajando en un espacio donde no se encuentra la solución, si este espacio es denso dentro del espacio donde se encuentra la solución, es posible acercarse a la solución mediante una sucesión de elementos del otro espacio. También es importante destacar el resultado del concepto de interpolación: el conjunto de funciones locales de interpolación de una subregión forman una base mínima dentro de su espacio vectorial, por lo que una combinación lineal adecuada de éstas generará cualquier otra función, incluso la solución o una aproximación a ésta. Y precisamente el método de elemento finito es un generador de combinaciones lineales y subespacios que en el mejor de los casos, tienden a la solución. Estos conceptos son tomados en su mayoría de [4] y reforzados algunos de ellos de [1], [10], [11] y [25].

En el capítulo 2 se da una definición formal del método de elemento finito. Aquí es donde se deja ver el procedimiento sistemático que se menciona al inicio, el cual lleva a una abstracción matemática y permite identificar el espacio en el cual se está trabajando y en el cual se espera encontrar una solución. Esto a su vez, permite conocer y entender los mecanismos que hacen del método de elemento finito una herramienta eficaz para encontrar aproximaciones a la solución de un problema en ecuaciones diferenciales. Los conceptos de este capítulo que concluyen en la definición del método de elemento finito son tomados de [4]. Con la segunda parte del capítulo anterior y este capítulo queda satisfecho el segundo objetivo planteado.

En el capítulo 3 se muestra al método de elemento finito tal como se define en el capítulo 2, como una técnica para encontrar aproximaciones a la solución de problemas en ecuaciones diferenciales; aquí se aplica el método de elemento finito para construir los modelos numéricos del problema de elasticidad dinámica lineal bidimensional. La solución es un campo vectorial en dos dimensiones cuyo dominio es el movimiento del espacio de definición del problema, es decir, hay una variable adicional: el tiempo. Para mayor claridad se separa el caso estático del caso dinámico. En este planteamiento se emplean dos técnicas de aproximación: una espacial dada por el método de elemento finito y una temporal aplicando un esquema recursivo por el método de diferencias finitas para ecuaciones diferenciales con segunda derivada en el tiempo. El resultado final es el modelo numérico equivalente del problema dinámico. La construcción de formulaciones equivalentes es un proceso implícito en la aplicación del método de elemento finito, sin embargo, su clasificación es importante para conocer en qué parte del proceso se encuentra el análisis. Esta clasificación se establece en [4], [13] y más claramente en [18]. Con este capítulo queda cumplido el tercer objetivo.

Cabe mencionar que el problema elastodinámico fue resuelto a partir de las ecuaciones de campo definidas en el capítulo 1 como problemas asociados de la elastoestática y elastodinámica, donde la ecuación de movimiento está caracterizada por el operador divergencia sobre el campo tensorial de esfuerzo; aparentemente se tiene más de un campo incógnita, sin embargo, con el uso de las ecuaciones constitutivas, la incógnita es sólo el campo de desplazamientos. Pero existen expresiones equivalentes de la ecuación de movimiento cuya única incógnita, de inicio, es la función solución. Estas ecuaciones están caracterizadas por el operador laplaciano y el operador gradiente, a veces llamados términos difusivo y convectivo, respectivamente. En este caso la formulación variacional de estas ecuaciones es difícil de manejar debido al operador gradiente, por lo que resultó más conveniente tomar la ecuación de movimiento en su forma ya citada.

Como una aplicación de los modelos numéricos equivalentes obtenidos en el capítulo anterior, en el capítulo 4 se aplica el planteamiento general de la elastodinámica lineal bidimensional a un caso particular, una viga en cantiléver sujeta a una carga dinámica en su extremo libre. Una vez definido el problema, su dominio es discretizado en subregiones de geometría conocida y se definen sus funciones locales y globales de interpolación. Se agrega que como ejercicio matemático se construyeron las últimas con las funciones de deformación de cada subregión así como con su definición formal. Retomando los modelos numéricos de las matrices de elemento finito del capítulo 3 y discretizando su construcción es posible captar, y con ello validar parcialmente el trabajo, sus propiedades de simetría. Las propiedades de bandedo y dominancia diagonal se observan visualmente en los resultados de la corrida de los programas elaborados para la solución de este problema, que más adelante se describen. Se toma el modelo numérico equivalente del problema dinámico y se define la recursión y el algoritmo para encontrar la solución en cada paso del tiempo, incluyendo la solución estática. La solución del problema representa los desplazamientos de puntos específicos del dominio; para determinar otro tipo de variables, se incluye una interpretación de los conceptos de deformación y esfuerzo para obtener una mayor información acerca del fenómeno.

En el capítulo 5 se muestran los algoritmos de programación diseñados para la implementación computacional del método de elemento finito por medio de un sistema de tres módulos codificados en lenguaje C, el flujo de información de un módulo a otro es por medio de archivos de lectura y escritura. El primer módulo, conocido generalmente como pre-procesador, está destinado a la adquisición de datos. En el procesador se llevan a cabo los cálculos necesarios para la construcción y solución del modelo numérico equivalente, a partir de los datos obtenidos en el módulo anterior. Hasta aquí la ecuación diferencial ha sido resuelta, falta interpretar adecuadamente los resultados de acuerdo con el fenómeno físico, tarea que debe realizar el post-procesador, su función es representar y manejar los resultados obtenidos en el procesador y los datos del pre-procesador, ya sea en forma numérica o gráfica. El sistema se representa por medio de bloques, los cuales son detallados en cada uno de sus módulos por una mezcla de lenguaje natural y lenguaje C. Con la ayuda de los programas descritos se resuelve numéricamente el caso estático, aunque se dan las bases para el tratamiento numérico del caso dinámico. Como parte final de este capítulo, se analizan los resultados numéricos obtenidos, como una verificación del análisis de elemento finito, con respecto a una solución conocida del problema. Para ello se emplea una solución analítica de la mecánica de materiales conocida como ecuación diferencial de la curva elástica [2].

Al final de este documento hay una serie de apéndices en los que se ha documentado parte del estudio necesario en el desarrollo de este trabajo, que al considerarse de dominio público no puede incluirse dentro del cuerpo central de este documento pero sí debe tenerse en consideración. Se pueden tomar como una fuente de información o referencia a otras fuentes.

Finalmente deseo agradecer a todas aquellas personas e instituciones que me ayudaron a dar un paso más en mi formación académica. Sin pretender descalificar la participación de muchas más personas, quiero expresar un reconocimiento especial al Dr. Jorge Carrera por su valiosa enseñanza y dirección, y al Dr. Juan Carlos Jáuregui por su invaluable apoyo y confianza. Un reconocimiento más para el Dr. Luis Ferrer por sus comentarios, que junto con los de los primeros, durante el período de revisión y corrección, ayudaron a identificar y realzar la esencia y el valor de este trabajo. A las organizaciones CIATEQ, A. C., CONACYT y Fundación Telmex por su apoyo, interés y preocupación en la formación de recursos humanos de calidad que contribuyan al desarrollo de México.

*Luciano Vela Martínez.
Aguascalientes, Ags.; febrero de 1999.*

Conceptos básicos.

Los conceptos de espacio euclidiano, tensor y campo, utilizados con bastante frecuencia en este trabajo deben ser entendidos de acuerdo con las siguientes definiciones, tomadas de [9, pp. 1-9].

Espacio euclidiano.

El espacio euclidiano \mathcal{E} es el espacio donde se lleva a cabo la física, sus elementos son llamados **puntos** y no es posible hacer operaciones con ellos.

Tensor.

Un **tensor** \mathbf{T} es una transformación lineal de un espacio vectorial en sí mismo, $\mathbf{T} : V \rightarrow V$.

Campo.

Un **campo** es una función cuyo dominio es el espacio euclidiano \mathcal{E} .

El tensor **identidad** \mathbf{I} está definido por $\mathbf{I}v = v$ para cada $v \in V$.

Un tensor \mathbf{S} es **simétrico** si $\mathbf{S} = \mathbf{S}^T$ y **antisimétrico** si $\mathbf{S} = -\mathbf{S}^T$.

Dada una base mínima en V y un origen en \mathcal{E} , la **matriz asociada** a un tensor \mathbf{S} es la representación en coordenadas de \mathbf{S} , y se representará por $[\mathbf{S}]$. El **determinante** de un tensor \mathbf{S} está dado por el determinante de la matriz asociada al tensor \mathbf{S} , $\det \mathbf{S} = \det [\mathbf{S}]$.

Un tensor \mathbf{Q} es **ortogonal** si preserva el producto interno: $(\mathbf{Q}u, \mathbf{Q}v) = (u, v)$.

Una **rotación** \mathbf{R} es un tensor ortogonal con determinante positivo.

Un tensor \mathbf{S} es **definido positivo** si $(v, \mathbf{S}v) > 0$ para cada vector $v \neq 0$.

La nomenclatura usada para identificar el dominio y rango de las funciones manejadas en este documento es la siguiente:

$L(V, V)$	=	conjunto de transformaciones lineales de V en V (tensores).
Sim	=	conjunto de tensores simétricos.
Asim	=	conjunto de tensores antisimétricos.
Ort	=	conjunto de tensores ortogonales.
Ort ⁺	=	conjunto de tensores ortogonales y de determinante positivo (rotaciones).
\mathfrak{R}	=	conjunto de escalares, generalmente el conjunto de los números reales.
\mathfrak{R}^+	=	conjunto de escalares positivos.

El conjunto $L(V, V)$ tiene estructura de espacio vectorial con las operaciones de suma y multiplicación de tensores, y los subconjuntos Sim, Asim, Ort y Ort⁺ son subespacios de $L(V, V)$.

Espacio de trabajo.

Un **espacio afín** \mathcal{E} está formado por el espacio euclidiano \mathcal{E} cuyos elementos llamaremos **puntos**, un espacio vectorial asociado V de dimensión n , cuyos elementos llamaremos **vectores**, y una función de referencia $P : \mathcal{E} \times \mathcal{E} \rightarrow V$ tal que:

$$P(x, y) = y - x = v.$$

El **espacio de trabajo** de la elasticidad bidimensional estará dado en el espacio afín \mathcal{E} , constituido por el conjunto $(\mathcal{E}, \mathfrak{R}^2, P)$.

Un **marco de coordenadas cartesianas** en \mathcal{E} consiste de una base ortonormal $\{e_1, e_2\} \subset \mathfrak{R}^2$, un punto $\mathbf{o} \in \mathcal{E}$ llamado **origen** y un conjunto de **rectas** definidas por:

$$l_1 = \{x : P(\mathbf{o}, x) = x - \mathbf{o}\} \quad y \quad l_2 = \{y : P(\mathbf{o}, y) = y - \mathbf{o}\}.$$

Las **coordenadas** de un punto $p \in \mathcal{E}$ con respecto a un marco de coordenadas cartesianas en \mathcal{E} están dadas por:

$$x = (x - \mathbf{o}) e_1 \quad y \quad y = (y - \mathbf{o}) e_2.$$

Capítulo 1:

Fundamentos físico - matemáticos.

A. ELASTICIDAD.

1.1 CINEMÁTICA.

Cuerpos y deformaciones.

Definición 1.1. Un **cuerpo** \mathbf{B} es una región regular, posiblemente acotada, del espacio euclidiano \mathcal{E} . Generalmente se referirá a \mathbf{B} como la **configuración de referencia**, mientras que los puntos $\mathbf{p} \in \mathbf{B}$, se llamarán **puntos materiales**. Una **parte** P de \mathbf{B} es una subregión regular acotada de \mathbf{B} . Esto es,

$$\mathbf{B} \subset \mathcal{E} \quad \text{y} \quad P \subset \mathbf{B}.$$

Definición 1.2. Una **deformación** del cuerpo \mathbf{B} es una función $f : \mathbf{B} \rightarrow \mathcal{R}$, diferenciable y uno-a-uno tal que:

$$\det \nabla f > 0.$$

donde $\mathcal{R} \subseteq \mathcal{E}$ es una región cerrada.

Definición 1.3. Dada una deformación f , el **campo de desplazamientos** de f es el campo vectorial $u : \mathbf{B} \rightarrow V$ definido por:

$$u(\mathbf{p}) = f(\mathbf{p}) - \mathbf{p} \quad (1)$$

para todo $\mathbf{p} \in \mathbf{B}$, donde V es el espacio vectorial asociado a \mathcal{E} .

Definición 1.4. Si el campo de desplazamientos u es constante, se dice que la deformación f es una **traslación**.

Definición 1.5. El **gradiente de deformación** es el campo tensorial $F : \mathbb{B} \rightarrow L(V, V)$ definido por:

$$F(p) = \nabla f(p).$$

Definición 1.6. Si F es constante, se dice que f es una **deformación homogénea** y admite la siguiente representación:

$$f(p) = f(q) + F(p - q) \quad (2)$$

para todo $p, q \in \mathbb{B}$.

Definición 1.7. Una **rotación** alrededor de q es una deformación homogénea con punto fijo q tal que:

$$f(p) = q + R(p - q) \quad (3)$$

donde R es una rotación.

Definición 1.8. Una **elongación** desde q es una deformación homogénea con punto fijo q tal que:

$$f(p) = q + U(p - q) \quad (4)$$

donde U es un campo tensorial simétrico y definido positivo.

Definición 1.9. Sea f una elongación desde q tal que:

$$U = I + (\lambda - 1) e \otimes e$$

con $\lambda > 0$ y $\|e\| = 1$, se dice que f es una **extensión** de cantidad λ en dirección e .

Lema 1.1. Deformaciones homogéneas¹.

Cada deformación homogénea puede ser descompuesta en una secuencia traslación-rotación-elongación, la cual puede ser realizada en cualquier orden.

Deformaciones pequeñas.

Definición 1.10. Dada una deformación f y su correspondiente campo de desplazamientos u , el **campo tensorial de deformación infinitesimal** $E : \mathbf{B} \rightarrow L(V, V)$ está definido para todo $p \in \mathbf{B}$ por:

$$E(p) = \frac{1}{2} (\nabla u(p) + \nabla^T u(p)) \quad (5)$$

donde ∇u es pequeño.

Movimientos.

Definición 1.11. Sea \mathbf{B} un cuerpo definido sobre el espacio euclidiano \mathcal{E} . Un **movimiento** de \mathbf{B} es una función al menos tres veces continuamente diferenciable

$$\mathbf{x} : \mathbf{B} \times \mathcal{R} \rightarrow \mathcal{E}$$

con $\mathbf{x}(\bullet, t)$ para cada t fijo, una deformación de \mathbf{B} .

Definición 1.12. Sea $p \in \mathbf{B}$, el **lugar** ocupado por el punto material p al tiempo t está dado por:

$$x = \mathbf{x}(p, t);$$

entonces $\mathbf{B}_t = \mathbf{x}(\mathbf{B}, t)$ denota la región del espacio euclidiano \mathcal{E} ocupado por el cuerpo \mathbf{B} al tiempo t , que en lo sucesivo se denominará **configuración espacial**.

Definición 1.13. La **trayectoria** de un movimiento se define por el conjunto

$$T = \{(x, t) \mid x \in \mathbf{B}_t, t \in \mathcal{R}\}.$$

¹ Este lema es una condensación de los teoremas de caracterización de deformaciones homogéneas y deformaciones rígidas, dados en [9, pp. 44, 49, 50]. Una aplicación de este lema se ilustra en el apéndice A-1: *Deformaciones en coordenadas cartesianas*, p. A-1.

Definición 1.14. La **velocidad** y **aceleración** de un movimiento \mathbf{x} son los campos vectoriales definidos por

$$\dot{\mathbf{x}}(\mathbf{p}, t) = \frac{\partial}{\partial t} \mathbf{x}(\mathbf{p}, t) \quad \text{y} \quad \ddot{\mathbf{x}}(\mathbf{p}, t) = \frac{\partial^2}{\partial t^2} \mathbf{x}(\mathbf{p}, t),$$

respectivamente.

Definición 1.15. Un **campo material** es una función con dominio $\mathbf{B} \times \mathcal{R}$, mientras que un **campo espacial** es una función con dominio \mathcal{T} .

1.2 MASA Y MOMENTO.

Conservación de masa.

Definición 1.16. Dada una deformación f de un cuerpo \mathbf{B} , existe un **campo de densidad** $\rho_f : f(\mathbf{B}) \rightarrow \mathcal{R}^+$, tal que la **masa** de alguna parte $P \subseteq \mathbf{B}$ está dada por:

$$m(P) = \int_{f(P)} \rho_f(\mathbf{x}) dV. \quad (6)$$

Axioma 1.2. Ley de conservación de la masa².

Definición 1.17. Una **distribución de masa** para un cuerpo \mathbf{B} es una familia de campos de densidad diferenciables

$$\rho_f : f(\mathbf{B}) \rightarrow \mathcal{R}^+,$$

una para cada deformación f , tales que:

$$\int_{f(P)} \rho_f(\mathbf{x}) dV = \int_{g(P)} \rho_g(\mathbf{x}) dV \equiv m(P) \quad (7)$$

para alguna parte $P \subseteq \mathbf{B}$ y todas las deformaciones f y g . El número $\rho_f(\mathbf{x})$ representa la **densidad** en el lugar $\mathbf{x} \in f(\mathbf{B})$ en la deformación f .

² Este axioma indica que la masa es una propiedad de los cuerpos independiente de las deformaciones que éstos sufren.

Definición 1.18. Si el campo $\rho_0 : \mathbf{B} \rightarrow \mathfrak{R}^+$ es una función constante para todo $p \in \mathbf{B}$, se conoce como **densidad del cuerpo**.

Momentum lineal y angular.

Definición 1.19. Sea \mathfrak{X} un movimiento de \mathbf{B} . Dada una parte $P \subseteq \mathbf{B}$ y un origen $o \in \mathcal{E}$, el **momentum lineal** $l(P, t)$ y el **momentum angular** $a(P, t)$ (alrededor de o) de P al tiempo t están definidos por:

$$\begin{aligned} l(P, t) &= \int_{P_t} v(x, t) \rho(x, t) dV, \\ a(P, t) &= \int_{P_t} r(x) \times v(x, t) \rho(x, t) dV, \end{aligned} \tag{8}$$

donde

$r : \mathcal{E} \rightarrow V$ es el campo vectorial de posición definido por $r(x) = x - o$.

1.3 FUERZA.

Sistemas de fuerzas.

Definición 1.20. Dado un cuerpo \mathbf{B} , sea N el conjunto de todos los vectores unitarios normales a $\partial\mathbf{B}$. Un **sistema de fuerzas** para el cuerpo \mathbf{B} durante un movimiento \mathfrak{X} con trayectoria \mathcal{T} , es el par de funciones $(s(n, x, t), b(x, t))$

$$s : N \times \mathcal{T} \rightarrow V, \quad b : \mathcal{T} \rightarrow V,$$

donde:

$s(n, x, t)$ es una función diferenciable de $x \in \mathbf{B}$, para cada $n \in N$ y $t \in \mathfrak{R}$.

$b(x, t)$ es una función continua de $x \in \mathbf{B}$, para cada $t \in \mathfrak{R}$.

Definición 1.21. Dado un origen $o \in \mathcal{E}$, la **fuerza** $f(P, t)$ y el **momento** $m(P, t)$ alrededor de o , sobre una parte P al tiempo t se definen por:

$$f(P, t) = \int_{\partial P_t} s(n, x, t) dA + \int_{P_t} b(x, t) dV \tag{9}$$

$$m(P, t) = \int_{\partial P_t} \mathbf{r}(x) \times \mathbf{s}(n, x, t) dA + \int_{P_t} \mathbf{r}(x) \times \mathbf{b}(x, t) dV.$$

donde n es el campo de vectores externos unitarios y normales a ∂P_t y r es el campo vectorial de posición.

Axioma 1.3. Leyes de balance de momento.

Dado un sistema de fuerzas para \mathcal{B} durante un movimiento, para cada parte P y tiempo t :

$$\dot{f}(P, t) = \dot{l}(P, t), \tag{10}$$

$$m(P, t) = \dot{a}(P, t).$$

donde $l(P, t)$ es el momentum lineal y $a(P, t)$ es el momentum angular (alrededor de o) de P al tiempo t (definición 1.19).

Teorema de existencia del campo tensorial de esfuerzo de Cauchy.

Definición 1.22. Sea $(\mathbf{s}(n, x, t), \mathbf{b}(x, t))$ un sistema de fuerzas para \mathcal{B} durante un movimiento. Una condición necesaria y suficiente para que las leyes de balance de momento sean satisfechas, es la existencia de un campo tensorial espacial simétrico T , llamado **campo tensorial de esfuerzo de Cauchy**, tal que para cada vector unitario n :

$$\mathbf{s}(n, x, t) = T(x, t) n; \tag{11}$$

además,

$$\operatorname{div} T(x, t) + \mathbf{b}(x, t) = \rho(x, t) \mathbf{v}(x, t), \tag{12}$$

conocida como **ecuación de movimiento**.

Definición 1.23. Dado un movimiento \mathcal{X} de \mathcal{B} , un **proceso dinámico** es el par $(\mathcal{X}, T(x, t))$ donde T es un campo tensorial espacial simétrico y $T(x, t)$ es una función diferenciable de $x \in \mathcal{B}_t$.

Definición 1.24. Un **cuerpo material** es un cuerpo \mathcal{B} junto con una distribución de masa y una familia \mathcal{C} de procesos dinámicos.

Definición 1.25. Las **consideraciones constitutivas** son hipótesis adicionales que establecen relaciones específicas entre el movimiento y la forma del campo tensorial de esfuerzo de Cauchy.

Definición 1.26. La **clase constitutiva** de un cuerpo \mathbf{B} es una familia \mathcal{C} de procesos dinámicos consistente con las consideraciones constitutivas del cuerpo.

1.4 ELASTICIDAD FINITA.

Cuerpos elásticos, isotrópicos y homogéneos.

Definición 1.27. Un **cuerpo elástico** es un cuerpo material cuya clase constitutiva está definida por una función de respuesta diferenciable

$$\hat{T} : L(V, V)^+ \times \mathbf{B} \rightarrow \text{Sim}$$

como el conjunto \mathcal{C} de todos los procesos dinámicos $(\mathbf{x}, T(x, t))$ tales que

$$T(x, t) = \hat{T}(F(p, t), p),$$

donde $x = \mathbf{x}(p, t)$.

Definición 1.28. Dado un cuerpo elástico \mathbf{B} y un punto $p \in \mathbf{B}$. Una **transformación simétrica** en p es un campo tensorial $Q \in \text{Ort}^+$ tal que:

$$\hat{T}(F(p, t), p) = \hat{T}(F(\bullet, t) Q(p, t), p)$$

para cada $F \in L(V, V)^+$.

Definición 1.29. Dado un cuerpo elástico \mathbf{B} y un punto $p \in \mathbf{B}$. El **grupo simétrico** en p es el conjunto de todas las transformaciones simétricas en p y se denota por G_p .

$$G_p = \{ Q \in \text{Ort}^+ \mid \hat{T}(F(p, t), p) = \hat{T}(F(\bullet, t) Q(p, t), p) \}.$$

Definición 1.30. Dado un cuerpo elástico \mathbf{B} y un punto $p \in \mathbf{B}$. Se dice que el material en p es **isotrópico** si cada rotación es una transformación simétrica en p , esto es:

$$G_p = \text{Ort}^+;$$

y se dice que el material en p es **anisotrópico** si

$$G_p \neq \text{Ort}^+.$$

Definición 1.31. Dado un cuerpo elástico \mathbf{B} y un punto $p \in \mathbf{B}$, el campo tensorial de esfuerzo en p cuando el cuerpo no está deformado:

$$T_R(x, t) = \hat{T}(I(p, t), p) = \hat{T}(I, p),$$

donde $x = \mathbf{x}(p, t)$, se conoce como el **esfuerzo residual** en p .

Definición 1.32. Dado un cuerpo elástico \mathbf{B} y un punto $p \in \mathbf{B}$. Se dice que el cuerpo es **homogéneo** si $\rho_0(p)$ y $\hat{T}(F(p, t), p)$ son independientes del punto material p .

El campo tensorial de esfuerzo de Piola-Kirchhoff.

El análisis de cuerpos elásticos utilizando el campo tensorial de esfuerzo de Cauchy, requiere del conocimiento de la configuración espacial para cada instante del movimiento, lo que en la mayoría de las aplicaciones reales no es posible. De ahí que se tenga la necesidad de un segundo campo tensorial que represente al primero desde la configuración material, satisfaciendo las leyes de balance de momento.

Definición 1.33. Sea $(\mathbf{x}, T(x, t))$ un proceso dinámico. Dada una parte P de un cuerpo \mathbf{B} , el campo tensorial material $S : \mathbf{B} \times \mathcal{R} \rightarrow L(V, V)$ definido por

$$S(p, t) = T_m(p, t) G(p, t) = (\det F(p, t)) T_m(\bullet, t) F^{-T}(p, t), \quad (13)$$

tal que

$$\int_{\partial P_t} T(x, t) m(x) dA = \int_{\partial P} S(p, t) n(p) dA.$$

se conoce como **campo tensorial de esfuerzo de Piola-Kirchhoff**.

Definición 1.34. El campo material vectorial $b_0 : \mathbf{B} \times \mathcal{R} \rightarrow V$ definido por

$$b_0(p, t) = \det F(p, t) b_m(p, t)$$

se conoce como la **densidad de fuerza de cuerpo de referencia**, y mide la fuerza de cuerpo por unidad de volumen en la configuración de referencia.

Teorema 1.3. Propiedades del campo tensorial de Piola-Kirchhoff.

El campo tensorial de esfuerzo de Piola-Kirchhoff satisface las leyes de balance:

$$\int_{\partial P} S(p, t) \cdot n(p) \, dA + \int_P b_o(p, t) \, dV = \int_P \rho_o(p) \dot{\mathbf{X}}(p, t) \, dV, \tag{14}$$

$$\int_{\partial P} r_m(p) \times S(p, t) \cdot n(p) \, dA + \int_P r_m(p) \times b_o(p, t) \, dV = \int_P r_m(p) \times \rho_o(p) \dot{\mathbf{X}}(p, t) \, dV$$

para cada parte P.

El campo tensorial de elasticidad.

Definición 1.35. Dado un cuerpo elástico \mathbf{B} y un punto $p \in \mathbf{B}$, la transformación lineal $C : L(V, V) \rightarrow L(V, V)$ definida por:

$$C = D\hat{S}(I(p, t), p) = D\hat{S}(I, p),$$

es un campo tensorial simétrico conocido como **campo tensorial de elasticidad** para el punto material p.

El campo tensorial C puede verse como la derivada del campo tensorial de esfuerzo de Piola-Kirchhoff con respecto a $F(p, t)$ en $F(p, t) = I$; esto significa que determina el comportamiento de la ecuación constitutiva

$$S(p, t) = \hat{S}(F(p, t), p) \tag{15}$$

cuando $F(p, t)$ se aproxima a I.

Teorema 1.4. Propiedades del campo tensorial de elasticidad.

Dado un cuerpo elástico \mathbf{B} con esfuerzo residual cero, para cada $H \in L(V, V)$:

$$C[H] = C[E + W] = C[E].$$

donde

$$E = \frac{1}{2}(H + H^T) \quad y \quad W = \frac{1}{2}(H - H^T)$$

son las partes simétrica y antisimétrica de $H \in L(V, V)$, respectivamente.

Teorema 1.5. Módulos de Lamé.

Sea un cuerpo elástico \mathcal{B} con esfuerzo residual cero y un punto material $p \in \mathcal{B}$ isotrópico. Entonces existen campos escalares $\mu : \mathcal{B} \rightarrow \mathfrak{R}$ y $\lambda : \mathcal{B} \rightarrow \mathfrak{R}$ tales que:

$$C[E(p, t)] = 2\mu(p)E(p, t) + \lambda(p)(\text{tr}E(p, t)) I$$

para cada campo tensorial $E \in \text{Sim}$, conocidos como **módulos de Lamé** en p .

1.5 ELASTICIDAD LINEAL.

Derivación de la teoría lineal.

La teoría lineal de la elasticidad, la cual es apropiada para deformaciones pequeñas de los cuerpos, se deduce a partir de la linealización de la ecuación constitutiva general (15).

Teorema 1.6. Forma asintótica de la ecuación constitutiva.

Dado un cuerpo elástico \mathcal{B} y un punto $p \in \mathcal{B}$, sean C y E los campos tensoriales de elasticidad y de deformación infinitesimal, respectivamente. Supóngase que el esfuerzo residual en la configuración de referencia es cero. Entonces la ecuación constitutiva (15) adquiere la forma:

$$S(p, t) = C[E(p, t)] + \theta[\nabla u(p, t)] \tag{16}$$

Esto es, con un error de $\theta[\nabla u]$ cuando $\theta[\nabla u]$ tiende a cero, el campo tensorial de esfuerzo S es una función lineal del campo tensorial de deformación infinitesimal E . También, puesto que C es simétrico, con el mismo error, S es un campo tensorial simétrico.

La teoría lineal de elasticidad está basada en la ley de esfuerzo-deformación (16) con los términos de orden $\theta[\nabla u(p, t)]$ despreciables, en la relación desplazamiento-deformación (5) y en la ecuación de movimiento en su configuración material, esto es:

$$\begin{aligned} S(p, t) &= C[E(p, t)], \\ E(p, t) &= \frac{1}{2} [\nabla u(p, t) + \nabla u^T(p, t)], \\ \text{Div } S(p, t) + b_0(p, t) &= \rho_0(p) \ddot{u}(p, t). \end{aligned} \tag{17}$$

Las ecuaciones (17) forman un sistema de ecuaciones diferenciales parciales lineales para los campos u , E y S , dados C , ρ_0 y b_0 . Por el teorema de la sección anterior sobre la forma del campo tensorial de elasticidad, cuando el cuerpo es isotrópico, la ecuación (17)₁ puede tener la forma:

$$S(p, t) = 2\mu E(p, t) + \lambda(\text{tr}E(p, t)) I. \quad (18)$$

Además, cuando el cuerpo es homogéneo, las funciones ρ_0 , μ y λ son constantes.

Definición 1.36. Supóngase ahora que el cuerpo \mathbf{B} es homogéneo e isotrópico. Entonces la teoría lineal de elasticidad está modelada por la ecuación:

$$\mu \Delta u(p, t) + (\mu + \lambda) \nabla(\text{Div } u(p, t)) + b_0(p, t) = \rho_0(p) \ddot{u}(p, t), \quad (19)$$

que es la **ecuación de movimiento de desplazamientos**. En la teoría estática, cuando $\ddot{u}(p, t) = 0$, se tiene la **ecuación de equilibrio de desplazamientos**:

$$\mu \Delta u(p, t) + (\mu + \lambda) \nabla(\text{Div } u(p, t)) + b_0(p, t) = 0. \quad (20)$$

Elastoestática lineal.

El sistema de ecuaciones de campo que rige el comportamiento estático de un cuerpo elástico \mathbf{B} , dentro del marco teórico de la teoría lineal, está formado por la relación deformación-desplazamiento, la relación esfuerzo-deformación y la ecuación de equilibrio:

$$\begin{aligned} E(p) &= \frac{1}{2} (\nabla u(p) + \nabla^T u(p)), \\ S(p) &= C[E(p)], \\ \text{Div } S(p) + b_0(p) &= 0, \end{aligned} \quad (21)$$

para cada $p \in \mathbf{B}$. El campo tensorial de elasticidad C generalmente dependerá de la posición p en \mathbf{B} y se asumirá que C es una función diferenciable en cada $p \in \mathbf{B}$.

Definición 1.37. Una lista $[u, E, S]$ de campos diferenciables sobre \mathbf{B} que satisfacen las ecuaciones de campo (21) para una densidad de fuerza de cuerpo b_0 dada, se conoce como **estado elástico** correspondiente a b_0 .

El problema asociado de elastoestática.

Definición 1.38. Dado un cuerpo elástico \mathbf{B} . Sean S_1 y S_2 subconjuntos regulares complementarios de la frontera de \mathbf{B} , esto es:

$$\partial\mathbf{B} = S_1 \cup S_2 \quad \text{y} \quad S_1^\circ \cap S_2^\circ = \emptyset,$$

donde S_α° es el interior respecto de S_α . El **problema asociado** de elastoestática se establece como:

- dados un cuerpo \mathbf{B} con fronteras S_1 y S_2 , un campo tensorial de elasticidad C sobre \mathbf{B} , un campo vectorial de densidad de fuerza de cuerpo b_0 sobre \mathbf{B} , un campo vectorial de desplazamientos superficiales \hat{u} sobre S_1 y tracciones de superficie \hat{s} sobre S_2 ,
- encontrar un estado elástico $[u, E, S]$ que corresponda a b_0 y satisfaga las condiciones de frontera

$$u(p) = \hat{u}(p) \quad \text{sobre } S_1,$$

$$S(p)n = \hat{s}(p) \quad \text{sobre } S_2,$$

que será llamado una **solución**.

Elastodinámica lineal.

El comportamiento dinámico de un cuerpo elástico \mathbf{B} , dentro del marco teórico de la teoría lineal, está regido por las ecuaciones:

$$\begin{aligned} E(p, t) &= \frac{1}{2} (\nabla u(p, t) + \nabla^T u(p, t)), \\ S(p, t) &= C[E(p, t)], \end{aligned} \tag{22}$$

$$\text{Div } S(p, t) + b_0(p, t) = \rho_0(p) \ddot{u}(p, t),$$

para cada $p \in \mathbf{B}$ y tiempo t , donde \mathbf{B} es acotado y ρ_0 es una función continua.

Definición 1.39. Sea $[u, E, S]$ una lista de campos sobre $\mathbf{B} \times [0, \infty)$ donde u es al menos dos veces continuamente diferenciable, y E y S son diferenciables; supóngase que el sistema de

ecuaciones (22) se mantiene para alguna una densidad de fuerza de cuerpo b_0 sobre $\mathbf{B} \times [0, \infty)$. Entonces $[u, E, S]$ se conoce como **proceso elástico** correspondiente a b_0 .

El problema asociado de elastodinámica.

Definición 1.40. Dado un cuerpo elástico \mathbf{B} . Sean S_1 y S_2 subconjuntos regulares complementarios de la frontera de \mathbf{B} , esto es:

$$\partial\mathbf{B} = S_1 \cup S_2 \quad \text{y} \quad S_1^\circ \cap S_2^\circ = \emptyset,$$

donde S_α° es el interior respecto de S_α . El **problema asociado** de elastodinámica se establece como:

- dados un cuerpo \mathbf{B} con fronteras S_1 y S_2 , un campo tensorial de elasticidad C sobre \mathbf{B} , un campo de densidad de referencia ρ_0 sobre \mathbf{B} , un campo vectorial de densidad de fuerza de cuerpo b_0 sobre $\mathbf{B} \times [0, \infty)$, un campo vectorial de desplazamientos superficiales \hat{u} sobre $S_1 \times [0, \infty)$, un campo vectorial de tracciones superficiales \hat{s} sobre $S_2 \times [0, \infty)$, un campo vectorial de desplazamiento inicial u_0 sobre \mathbf{B} y un campo vectorial de velocidad inicial v_0 sobre \mathbf{B} ,
- encontrar un proceso elástico $[u, E, S]$ que corresponda a b_0 y satisfaga las condiciones iniciales

$$u(p, 0) = u_0(p),$$

$$\dot{u}(p, 0) = v_0(p),$$

para cada $p \in \mathbf{B}$,

y las condiciones de frontera:

$$u(p, t) = \hat{u}(p, t) \quad \text{sobre } S_1 \times [0, \infty),$$

$$S(p, t)n = \hat{s}(p, t) \quad \text{sobre } S_2 \times [0, \infty),$$

que será llamado una **solución**.

B. MATEMÁTICAS.

1.6 TEORÍA BÁSICA DE DISTRIBUCIONES.

Definición 1.41. El **soporte** de una función $f : \mathfrak{R} \rightarrow V$ es el conjunto $\{x : f(x) \neq 0\}$, denotado por $\text{sop}(f)$. Una función tiene **soporte acotado** si existen escalares $a, b \in \mathfrak{R}$ tales que $\text{sop}(f) \subset [a, b]$.

Definición 1.42. Una función $f : \mathfrak{R} \rightarrow V$ es **n veces continuamente diferenciable** si sus primeras n derivadas existen y son continuas. Se dice que f es **suave** o infinitamente diferenciable si sus derivadas de todos los órdenes existen y son continuas.

Definición 1.43. Una **función de prueba** es una función $f : \mathfrak{R} \rightarrow V$ suave con soporte acotado. Sea \mathcal{D} el conjunto de todas las funciones de prueba.

Definición 1.44. Un **funcional lineal** sobre \mathcal{D} es una función $f : \mathcal{D} \rightarrow V$ tal que:

$$f(a\varphi_1 + b\varphi_2) = a f(\varphi_1) + b f(\varphi_2) \quad (23)$$

para todo escalar $a, b \in \mathfrak{R}$ y toda función de prueba $\varphi_1, \varphi_2 \in \mathcal{D}$.

Definición 1.45. Sea $\{\varphi_n\}$ una sucesión de funciones de prueba y Φ otra función de prueba. Se dice que φ_n **converge a Φ en \mathcal{D}** si

- existe un intervalo $[a, b]$ que contiene a $\text{sop}(\varphi_n)$ y $\text{sop}(\Phi)$ para todo n ,
- para todo k , $\varphi_n^{(k)}(x)$ converge a $\Phi^{(k)}(x)$ cuando $n \rightarrow \infty$ uniformemente para $x \in [a, b]$, donde $\varphi_n^{(k)}$ denota la k -ésima derivada de φ_n .

Definición 1.46. Un funcional f sobre \mathcal{D} es **continuo** si relaciona cada sucesión convergente en \mathcal{D} con una sucesión convergente en V , esto es, si $f(\varphi_n)$ converge a $f(\Phi)$ en V siempre que φ_n converge a Φ en \mathcal{D} . Esto es, existen escalares $\varepsilon, \delta \in \mathfrak{R}$ tales que:

$$\|f(\varphi_n) - f(\Phi)\| < \varepsilon \quad \text{siempre que} \quad \|\varphi_n - \Phi\| < \delta. \quad (24)$$

Definición 1.47. Un funcional lineal continuo sobre \mathcal{D} es una **distribución** o **función generalizada**.

Definición 1.48. Una función $f: \mathfrak{R} \rightarrow V$ es **localmente integrable** si $\int_a^b \|f(x)\| dx$ existe para todo escalar $a, b \in \mathfrak{R}$.

Definición 1.49. Sea φ una función de prueba. Se dice que una distribución F es **generada** por una función f localmente integrable si se define por:

$$F(\varphi) = \int_{-\infty}^{\infty} f(x)\varphi(x) dx . \quad (25)$$

Definición 1.50. Una distribución generada por una función localmente integrable se llama **regular**. El resto de las distribuciones se llaman **singulares**.

1.7 ECUACIONES DIFERENCIALES LINEALES.

Definición.

Definición 1.51. Sean $L: V \rightarrow W$ un operador diferencial entre espacios vectoriales funcionales, D el operador derivada y $\{a_i\}$, $i = 0, 1, 2, \dots, n$, un conjunto de funciones suaves. El problema de encontrar u tal que:

$$L u = f \quad (26)$$

donde $L = a_0(x)D^n + a_1(x)D^{n-1} + a_2(x)D^{n-2} + \dots + a_n(x)$ y las funciones $u \in U \subseteq V$ y $f \in W$ son distribuciones, se conoce como **ecuación diferencial lineal de orden n** . El conjunto U se conoce como **conjunto de funciones admisibles** de la ecuación diferencial, y la distribución u se llama **solución** de la ecuación diferencial.

Definición 1.52. Si en una ecuación diferencial la derivada o derivadas de los elementos de U son con relación a un solo parámetro, el problema (26) se conoce como **ecuación diferencial ordinaria**; cuando son con relación a más de un parámetro el problema (26) se conoce como **ecuación diferencial parcial** (en lo sucesivo EDP).

Condiciones de frontera.

Definición 1.53. Sea u una función definida en una región regular $R \subseteq \mathcal{E}$. Se dice que la función u cumple **condiciones de frontera de orden k** si existen k funciones continuas g_i , $i = 0, 1, 2, \dots, k-1$, con la posible excepción de un número finito de conjuntos de medida cero, tal que:

$$u(x) = g_0(x)$$

$$\begin{aligned} Du(x) &= g_1(x) \\ \vdots & \quad \quad \quad \vdots \\ D^{k-1}u(x) &= g_{k-1}(x) \end{aligned}$$

en la frontera ∂R .

Debido a su naturaleza, las condiciones de frontera se han clasificado de la siguiente manera:

- a) Condiciones Dirichlet: Son establecidas sobre los valores de la función en la frontera, son de orden cero.
- b) Condiciones Neumann: Son establecidas sobre los valores de las primeras $n-1$ derivadas de la función en la frontera si el operador es de orden n , son de orden $n-1$.
- c) Condiciones mixtas: Sobre una misma frontera y una misma condición se incluyen valores de la función y sus derivadas.
- d) Condiciones completas: Son establecidas sobre el valor de la función y de sus primeras $n-1$ derivadas en la frontera, si el operador es de orden n .
- e) Condiciones mezcladas: Son una combinación de condiciones sobre diferentes partes de la frontera.

Las condiciones de frontera nos permiten, desde un punto de vista matemático, escoger de entre la multiplicidad de posibles soluciones una sola. Las condiciones de frontera son restricciones impuestas a la función solución y sus derivadas, sobre su valor en la frontera de la región de definición.

Soluciones fuertes y débiles de ecuaciones diferenciales lineales.

Definición 1.54 (Clasificación de soluciones a ecuaciones diferenciales).

- a) Una **solución generalizada** es una distribución que satisface la ecuación (26).
- b) Una **solución clásica** es una función ordinaria que es n veces diferenciable y satisface la ecuación (26).
- c) Una **solución débil** es una función ordinaria que no es n veces diferenciable y por lo tanto no es una solución clásica, pero satisface la ecuación (26).
- d) Una **solución distribucional** es una distribución singular que satisface la ecuación (26).

Definición 1.55. Un planteamiento de un problema en EDP de orden n , donde a la función solución se le exige ser n veces continuamente diferenciable se llama **formulación fuerte** del problema, y la función solución se llama **solución fuerte**. Por otro lado, un planteamiento equivalente a una formulación fuerte donde se ha conseguido que a la función solución se le exija

menor grado de diferenciación se llama **formulación débil** del problema, y la función solución se llama **solución débil**.

Definición 1.56. Una formulación fuerte del problema está definida punto a punto sobre el dominio por medio de una EDP y se conoce como **planteamiento local**. Una formulación débil, en cambio, está definida por una ecuación integro - diferencial sobre todo el dominio, por lo que se le conoce como **planteamiento global**.

1.8 TEORÍA DE CONVERGENCIA EN ESPACIOS CON MÉTRICA.

Espacios completos.

El problema de encontrar la solución de una ecuación diferencial por aproximaciones, dentro de un espacio vectorial con métrica es que, generalmente se tiene el procedimiento para generar una sucesión de aproximaciones de las que se espera se acerquen poco a poco a la solución exacta, pero se debe justificar tal procedimiento demostrando que tal sucesión converge a la solución [10, p. 95].

Definición 1.57. Sea A un conjunto cualesquiera no vacío y N el conjunto de los números naturales. Se dice que A tiene **cardinalidad** $n \in N$ si es posible definir una función biunívoca entre los primeros n números naturales y los elementos de A . También se dice que A es un conjunto finito de n elementos.

Definición 1.58. Si un conjunto tiene un subconjunto propio con su misma cardinalidad, se dice que tiene un número infinito de elementos y se conoce como **conjunto infinito**.

Definición 1.59. Sea $\{x_n\}$ una sucesión con $x_1, x_2, \dots, x_n \in V$ un espacio vectorial con métrica. Se dice que x_n **converge a** $x \in V$ si dado un escalar $\epsilon > 0$, existe $N \in N$ tal que

$$\|x_n - x\| < \epsilon \quad (27)$$

para todo $n > N$.

Definición 1.60 (Convergencia de una serie). Sea $\{x_n\}$ una sucesión con $x_1, x_2, \dots, x_n \in V$ un espacio vectorial con métrica. Sea $\{s_n\}$ la sucesión de sumas parciales donde

$$s_n = x_1 + x_2 + \dots + x_n \quad (28)$$

que llamaremos serie. Se dice que la serie s_n **converge a** $x \in V$ si la sucesión $\{s_n\}$ converge a $x \in V$, y se escribe

$$\sum_{n=1}^{\infty} x_n = x_1 + x_2 + \dots + x_n = x. \quad (29)$$

Definición 1.61. Sea $\{x_n\}$ una sucesión con $x_1, x_2, \dots, x_n \in V$ un espacio vectorial con métrica. Se dice que $\{x_n\}$ es una **sucesión de Cauchy** si para algún escalar $\varepsilon > 0$ existe un $N \in \mathbb{N}$ tal que $\|x_m - x_n\| < \varepsilon$ para todo $m, n > N$.

Definición 1.62. Si una sucesión convergente es una sucesión de Cauchy, se dice que la sucesión cumple con el **criterio de Cauchy**.

Definición 1.63. Un espacio vectorial con métrica en donde todas las sucesiones de Cauchy convergen se llama **espacio completo**, e **incompleto** en el otro caso.

Espacios de Banach.

Definición 1.64. Sean V un espacio vectorial con métrica y $U \subseteq V$. Un elemento $v \in V$, que puede o no ser elemento de U , se llama **punto de acumulación** o **punto frontera** de U si existe una sucesión de elementos de U que converja a v .

Definición 1.65. Sean V un espacio vectorial con métrica y $U \subseteq V$, el conjunto de todos los puntos de acumulación de U , denotado por ∂U , se llama **adherencia** o **frontera** de U .

Definición 1.66. Sean V un espacio vectorial con métrica y $U \subseteq V$. Se dice que U es un **conjunto abierto** si no contiene su frontera, por el contrario, se dice que U es un **conjunto cerrado** si contiene su frontera.

Definición 1.67. Sean U, V subconjuntos de un espacio normado con $U \subseteq V$. Se dice que U es **denso** en V si para cada $v \in V$ y cada escalar $\varepsilon > 0$ existe un $u \in U$ con

$$\|u - v\| < \varepsilon. \quad (30)$$

Definición 1.68. También se dice que U es **denso** en V si y sólo si para cada $v \in V$ existe una sucesión $\{u_n\}$ de elementos de U , tal que u_n converge a v . Esto es, los elementos de U están densamente dispersos entre los elementos de V .

Definición 1.69. Si un conjunto V tiene un subconjunto denso cuya cardinalidad sea la misma que la de los números naturales, se dice que V es **separable**.

Definición 1.70. Un espacio vectorial H de dimensión infinita con producto interno y separable, se llama **espacio de Hilbert**.

Definición 1.71. Un espacio completo es un **espacio de Banach**.

Contracciones.

El enlace entre la teoría de los espacios vectoriales y la solución de ecuaciones diferenciales e integrales de las matemáticas aplicadas que involucran elementos de tales espacios, es la definición de una transformación, función u operador sobre éstos [10, p. 116].

Definición 1.72. Sea $T : V \rightarrow V$ una transformación lineal de un espacio vectorial normado en sí mismo. Las soluciones de las ecuaciones de matemáticas aplicadas expresadas como

$$Tx = x \tag{31}$$

donde x es un elemento desconocido de V , se conocen como **puntos fijos** de T y son elementos del espacio vectorial que no son alterados por la acción de T .

Definición 1.73. Sea $T : V \rightarrow V$ una transformación lineal de un espacio vectorial normado en sí mismo. Se dice que T es una **contracción** si existe un escalar $a \in \mathfrak{R}$, $0 < a < 1$, tal que:

$$\|Tx - Ty\| \leq a\|x - y\| \tag{32}$$

para todo $x, y \in V$.

Una contracción es tal que aplicada a dos elementos del espacio los transforma a nuevos elementos que están más cercanos entre sí que el par original. Si T es una contracción sobre un subconjunto X de un espacio normado, la sucesión de aproximaciones $x_{n+1} = Tx_n$ es una sucesión de Cauchy. Si el espacio es completo la sucesión de Cauchy debe converger aunque su límite no necesariamente se encuentra en X , por lo que éste puede no ser un candidato para una solución de la ecuación $Tx = x$. Sin embargo, si X es un conjunto cerrado el límite de la sucesión debe estar en X y puede esperarse que satisfaga la ecuación $Tx = x$.

Teorema 1.7. Contracciones.

Si $T : X \rightarrow X$ es una contracción de un subconjunto cerrado X de un espacio de Banach, entonces existe exactamente un punto fijo $x \in X$ tal que $Tx = x$. Además, para cualquier $x_0 \in X$, la sucesión $\{x_n\}$ definida por $x_{n+1} = Tx_n$ converge a x .

1.9 APROXIMACIONES E INTERPOLACIÓN.

Las geometrías complejas de la región en que se encuentra definida la función solución de una ecuación diferencial, como las encontradas en problemas reales, hacen imposible la obtención de soluciones exactas, por lo que es necesario representar las soluciones por medio de soluciones aproximadas utilizando funciones ya conocidas, entre ellas: polinomios, funciones algebraicas, funciones trascendentes, series de potencias, etc. Un concepto que nos define el sentido de las soluciones aproximadas es el de métrica [4, pp. 99-106].

Definición 1.74. Sea V un espacio vectorial con métrica. Dado $v \in V$, un escalar $\epsilon > 0$ y una vecindad Ω de v , se dice que el elemento $v_i \in \Omega$ es una **aproximación** de v si $\|v - v_i\| \leq \epsilon$.

Definición 1.75. Sea V un espacio vectorial con métrica. Dado $v \in V$, el **error** de una aproximación v_i a v está dado por

$$\Delta\xi = \|v - v_i\|. \tag{33}$$

Definición 1.76. Sean dos conjuntos M y N (no necesariamente diferentes), un conjunto finito S de pares ordenados (m, n) tales que $m \in M$ y $n \in N$, y un conjunto de funciones $F : M \rightarrow N$. El problema de encontrar una función $f \in F$ tal que para cada elemento de S se tenga $(m, n) = (m, f(m))$ se llama **problema de interpolación**. Los elementos $m \in S$ se llaman **puntos de interpolación** y los elementos $n \in S$ se llaman **valores de interpolación**, mientras que F se conoce como **conjunto de funciones de interpolación**.

Lema 1.8. Polinomio de interpolación en espacios vectoriales.

Dados n pares ordenados (x_i, y_i) , $i = 1, 2, \dots, n$ donde cada $x_i, y_i \in V$ un espacio vectorial, existe uno y sólo un polinomio $p : V \rightarrow V$ de grado $n-1$ tal que $(x_i, y_i) = (x_i, p(x_i))$.

Lema 1.9. Polinomio de interpolación en espacios reales.

Sea el conjunto $F = \{p : \mathfrak{R} \rightarrow \mathfrak{R} \mid p(x) = Ax + B, A, B \in \mathfrak{R}, x \in [a, b]\}$, una base mínima de F es el subconjunto $B = \{\varphi_1(x) = \alpha_1x + \beta_1, \varphi_2(x) = \alpha_2x + \beta_2\}$ tal que

a) $\varphi_1(a) = 0$ y $\varphi_1(b) = 1,$

b) $\varphi_2(a) = 1$ y $\varphi_2(b) = 0$.

Entonces:

$$\varphi_1(x) = \frac{1}{b-a}(x-a) \quad \text{y} \quad \varphi_2(x) = -\frac{1}{b-a}(x-b).$$

La importancia del lema anterior es que el conjunto de polinomios de la forma $p(x) = Ax + B$ puede ser generado a partir de los dos polinomios de interpolación $\varphi_1(x)$ y $\varphi_2(x)$ así definidos.

Capítulo 2:

Definición del método de elemento finito.

2.1 MALLAS DE ELEMENTO FINITO.

Discretizaciones.

Definición 2.1. Dado un problema en EDP, el **dominio** del problema es una región regular $R \subset \mathcal{E}$ en la cual está definida su solución.

Definición 2.2. Sea R el dominio de un problema en EDP y sea $k \in N$ la dimensión del espacio euclidiano \mathcal{E} . La **discretización** o malla de la región R es una región regular $R' \subset \mathcal{E}$ constituida por un conjunto finito de n subregiones regulares, llamadas **elementos finitos**, cuyas fronteras tienen una representación analítica a trozos. Cada trozo es un subconjunto suave llamado **lado** del elemento finito y los puntos donde los trozos se encuentran se llaman **nodos**. Las características generales de una discretización son las siguientes:

$$a) R' = \sum_{j=1}^n E_j \quad . \quad (1)$$

b) Existe un escalar $\varepsilon > 0$ “suficientemente pequeño” tal que:

$$\max \|(R \setminus R') \cup (R' \setminus R)\| < \varepsilon. \quad (2)$$

c) Cada elemento finito es una subregión regular de medida no cero y convexa.

d) Dos elementos finitos no se interceptan, excepto posiblemente en su frontera:

$$E_i \cap E_j = \emptyset, \quad i \neq j. \quad (3)$$

El conjunto de intersección es de dimensión $k-1$, y se conoce como **intersección de primer orden**.

e) Sean E_i y E_j elementos finitos con intersección de primer orden. Siendo ambos subregiones regulares, su frontera está compuesta por lados:

$$\partial E_i = \bigcup_{l=1}^p \partial_l E_i, \quad \partial E_j = \bigcup_{h=1}^q \partial_h E_j. \quad (4)$$

Entonces existe uno y sólo un índice r y uno y sólo un índice s tales que los lados $\partial_r E_i$ y $\partial_s E_j$ son el mismo subconjunto:

$$\partial_r E_i = \partial_s E_j. \quad (5)$$

La definición anterior exige que los elementos finitos cubran toda la región R' sin traslaparse. En caso de intersección, ésta se da en la frontera y en elementos adyacentes debe coincidir un lado completo de la frontera de un elemento finito con uno y sólo un lado completo de la frontera de su adyacente.

Definición 2.3. Sean R' y R'' discretizaciones de una misma región R con el mismo tipo de elementos finitos. Si todo elemento de R'' es subconjunto de uno y sólo un elemento finito de R' con la posible excepción de puntos en la frontera de los elementos finitos, y existe al menos un elemento finito de R'' que es un subconjunto propio de un elemento de R' , entonces se dice que R'' es una **discretización más fina** que R' , de la región R .

Definición 2.4. La construcción de una malla más fina con base en una discretización dada, se llama **refinamiento** de malla. La alteración de un subconjunto propio de la malla original se llama **refinamiento local**.

Elemento finito de referencia.

Definición 2.5. El elemento finito de referencia (EFR) es una técnica de generación de mallas de elementos finitos. Consiste en elegir una figura geométrica que se usará en la discretización, ésta se ubica en relación a un sistema de coordenadas de manera que la representación analítica de sus frontera sea lo más sencilla posible. Mediante funciones de deformación convenientes de esta figura geométrica, llamada **elemento finito de referencia**, se cubre la región a discretizar. Dependiendo del tipo de función de deformación usada (traslaciones, rotaciones, elongaciones, etc.), así como de las diferentes geometrías de referencia empleadas, será la naturaleza de la malla.

2.2 GRADOS DE LIBERTAD Y FUNCIONES DE INTERPOLACIÓN.

Definición 2.6. Dado un problema en EDP y un conjunto $\{\phi_i\}$ de funciones de interpolación de la función solución $u : R \subseteq \mathcal{E} \rightarrow \mathfrak{R}$, los **interpolantes** $u_k : R \rightarrow \mathfrak{R}$ de la función solución u de la forma:

$$u_k(x) = \sum_{i=1}^n a_i \phi_i(x) \quad (6)$$

tales que

$$u_k(x_i) = a_i, \quad (7)$$

definen un conjunto $G = \{a_i\}$ de coeficientes, llamado **grados de libertad de primer orden** (son los valores de la función u en puntos específicos de su dominio), que se vuelven las incógnitas del problema.

Definición 2.7. Sea D el operador derivada y sea m la dimensión del espacio euclidiano \mathcal{E} . Dado un origen $\mathbf{o} \in \mathcal{E}$ y un sistema de coordenadas en el espacio afín $(P, \mathfrak{R}^m, \mathcal{E})$, la n -ésima derivada de una función $u : R \subseteq \mathcal{E} \rightarrow \mathfrak{R}$ es una función lineal $D^{n-1}A : \mathfrak{R}^m \rightarrow \mathfrak{R}$ cuya representación, por el teorema de Riesz [9, pp. 2, 29], está dada por un vector $\nabla^n u$, llamado **n -ésimo gradiente de u** . Las coordenadas del n -ésimo vector gradiente de u definen un conjunto $\{(\nabla^n u)_i\}$ de coeficientes llamado **grados de libertad de orden n** .

Definición 2.8. Dada una discretización R' de una región regular R y $E_k \subset R'$ un elemento finito, una **función local de interpolación** Φ_{ki} es una función que vale uno en el i -ésimo nodo de E_k y cero en el resto.

Definición 2.9. Dada una discretización R' de una región regular R , una **función global de interpolación** Φ_i es una función que vale uno en el i -ésimo nodo de R' y cero en el resto. Sobre cada uno de los elementos finitos E_k que comparten el i -ésimo nodo, es idéntica a la función local de interpolación ϕ_{ki} , que vale uno en ese nodo y cero en el resto.

Lema 2.1. Funciones globales de interpolación.

Una función global de interpolación tiene como soporte los elementos finitos que comparten el nodo en el cual Φ_i vale uno.

2.3 EL PROBLEMA ASOCIADO DEL MÉTODO DE ELEMENTO FINITO.

Definición 2.10. Dado un problema en EDP sobre una región R , el método de elemento finito consiste en definir una discretización R' sobre R , especificar un conjunto de grados de libertad G y definir un conjunto de funciones globales de interpolación Φ (las funciones locales están implícitamente definidas) sobre R' en función de G . Al conjunto $\{R', G, \Phi\}$ lo denominaremos **problema asociado de elemento finito** al problema en EDP sobre R .

Lema 2.2 de problemas asociados de elemento finito.

Dado un problema en EDP, existe una infinidad de problemas asociados del método de elemento finito.

Lema 2.3 de problemas asociados de elemento finito.

Dada una región regular R , cada conjunto $\{R'_i, G_i, \Phi_i\}$, $i = 1, 2, \dots$ es una base de un espacio vectorial V_i de problemas asociados del método de elemento finito.

Definición 2.11. En cada problema asociado del método de elemento finito existe una función $u_i \in V_i$ tal que, si es posible definir un criterio adecuado de convergencia y la función u_i converge a la función solución u del problema en EDP, entonces se dice que cada u_i es una **aproximación a la solución u** .

Definición 2.12. El **método de elemento finito** es una metodología para construir espacios vectoriales que contienen aproximaciones a la solución de un problema en EDP o uno de sus modelos matemáticos.

Definición 2.13. Sea V el espacio vectorial donde se encuentra la solución u a un problema en EDP. La sucesión $\{V_i\}$ de subespacios de V de dimensión cada vez mayor, construidos mediante el método de elemento finito, tal que $u_i \in V_i$ converge a u , se llama **aproximación interna**. La sucesión $\{V_i\}$ de espacios vectoriales que, sin ser parte de V lo van aproximando cada vez mejor, dado un criterio adecuado de convergencia, se llama **aproximación externa**.

2.4 PLANTEAMIENTOS TEÓRICOS GENERALES PARA LA APLICACIÓN DEL MÉTODO DE ELEMENTO FINITO.

Teorema 2.4. Unicidad de la solución.

Sea T un operador simétrico y definido positivo y sea $U \subseteq H$ un subespacio cerrado. El problema en EDP

$$Tu = f \tag{8}$$

tiene solución única $u_0 \in U$ si y sólo si

$$(Tu_0, v) = (v, f)$$

para todo $v \in U$.

Método de Ritz.

Sean H un espacio de Hilbert, $\{\alpha_1, \alpha_2, \dots, \alpha_n, \dots\}$ una base de H y los subespacios $H_i \subset H$ definidos por las sucesiones $\{\alpha_1\}$, $\{\alpha_1, \alpha_2\}$, \dots , $\{\alpha_1, \alpha_2, \dots, \alpha_i\}$:

$$H_i = \{ v \mid v = a_1\alpha_1 + a_2\alpha_2 + \dots + a_i\alpha_i \}, \quad i = 1, 2, \dots, n, \dots$$

Entonces, en cada subespacio H_i la solución tiene la forma

$$u_i = \sum_{k=1}^i a_k \alpha_k, \quad (9)$$

que en nuestro problema en EDP (8) viene a ser:

$$(Tu_i, v) = (v, f) = \left(T \sum_{k=1}^i a_k \alpha_k, v \right), \quad (10)$$

expresión que debe ser válida para todo $v \in H_i$, especialmente para $v = \alpha_k$:

$$\left(\sum_{k=1}^i a_k T\alpha_k, \alpha_j \right) = (\alpha_j, f), \quad j = 1, 2, 3, \dots, i. \quad (11)$$

Esta técnica se conoce como **método de Ritz** y consiste en construir un conjunto de subespacios de dimensión finita $H_i \subset H$ tales que:

$$H_1 \subset H_2 \subset H_3 \subset \dots \subset H_n \subset \dots \subset H,$$

y entonces resolver el problema en cada subespacio. Esto es, dar las condiciones que permitan asegurar que el conjunto $\{u_i\} \in H_i$ es una sucesión que converge a u_0 .

Método de Galerkin.

Sean H un espacio de Hilbert y $\{\alpha_1, \alpha_2, \dots, \alpha_n, \dots\}$ una base ortonormal de H . Entonces, la proyección de u en cada subespacio $H_i \subset H$ es una proyección ortogonal.

La utilización de una base ortonormal de H es lo que se conoce como **método de Galerkin**. Con esta característica es posible representar f en función de la base de H :

$$f = \sum_{k=1}^i f_k \alpha_k . \quad (12)$$

Entonces, el problema en EDP (8) resulta:

$$(Tu_i, v) = (v, f) , \quad (13)$$

$$\left(T \sum_{k=1}^i a_k \alpha_k, \alpha_j \right) = \left(\alpha_j, \sum_{k=1}^i f_k \alpha_k \right) , \quad (14)$$

$$\left(\sum_{k=1}^i a_k T \alpha_k, \alpha_j \right) = f_j . \quad (15)$$

Definición 2.14. Sea H un espacio de Hilbert con una base $\{\alpha_1, \alpha_2, \dots, \alpha_n, \dots\}$. Un **sistema fundamental** de H es una base que genera un subespacio denso en H .

2.5 DEFINICIÓN FORMAL DE MÉTODO DE ELEMENTO FINITO.

Definición 2.15. Dado un problema en EDP definido sobre una región $R \subset \mathcal{E}$ del espacio euclidiano y un espacio de Hilbert H , el **método de elemento finito** es una técnica de interpolación a la distribución que es solución del problema, mediante la construcción de un sistema fundamental polinomial de H a partir de un conjunto $\{R', G, \Phi\}$ sobre la región R .

Con un sistema fundamental polinomial y ortonormal de H se define una sucesión de subespacios polinomiales de dimensión finita, los cuales son densos en los espacios de distribuciones en donde se encuentra la solución del sistema original. Esto significa que, aún cuando la solución del problema en EDP sea una distribución, podremos aproximarnos a ésta tanto como nosotros queramos mediante la construcción de subespacios polinomiales de dimensión cada vez mayor.

Capítulo 3:

Planteamiento del problema de elasticidad lineal.

3.1 PROBLEMA DE ELASTOESTÁTICA LINEAL BIDIMENSIONAL.

Definición.

El problema de encontrar el **estado elástico** $[u, E, S]$ correspondiente a un cuerpo elástico que ocupa una región Ω del espacio, sujeto a un sistema de fuerzas estacionarias (\hat{s}, \mathbf{b}_0) , esto es, independientes del tiempo, está definido por el problema asociado de elastoestática por medio del modelo físico en ecuaciones diferenciales parciales (1), conocido como formulación fuerte del problema.

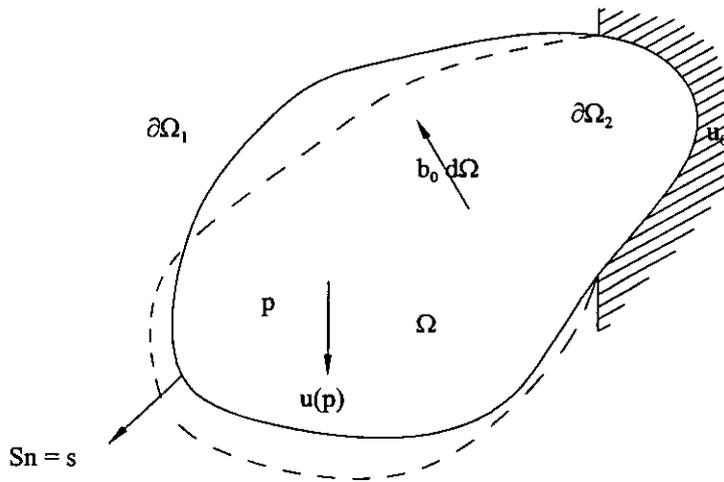


fig. 3-1: El problema de elasticidad estática.

Formulación fuerte.

Sea $\text{Div} : L(V, V) \rightarrow V$ el operador divergencia entre campos materiales:

$$\text{Div } S + \mathbf{b}_0 = \mathbf{0} \text{ en } \Omega,$$

$$S\mathbf{n} = \hat{\mathbf{s}} \text{ en } \partial\Omega_1, \quad (1)$$

$$\mathbf{u} = \mathbf{u}_0 \text{ en } \partial\Omega_2,$$

donde $S : \Omega \times \mathfrak{R} \rightarrow L(V, V)$ es el campo tensorial de esfuerzo de Piola-Kirchhoff, $\hat{\mathbf{s}} : \partial\Omega_1 \rightarrow V$ es el campo vectorial de fuerza superficial, $\mathbf{b}_0 : \Omega \rightarrow V$ es el campo vectorial de fuerzas de cuerpo y $\mathbf{n} : \partial\Omega \rightarrow V$ es el campo vectorial unitario normal a $\partial\Omega$.

Formulación variacional.

Sea $\omega : \mathcal{E} \rightarrow \mathfrak{R}$ una función de prueba, multiplicando la ecuación (1.1) por ω e integrando sobre todo el dominio Ω tenemos:

$$\int_{\Omega} \omega(\text{Div}S + \mathbf{b}_0) \, d\Omega = \int_{\Omega} (\omega \text{Div}S + \omega \mathbf{b}_0) \, d\Omega = 0 ;$$

al aplicar uno de los teoremas sobre derivada [9, p. 30] al primer término del integrando se obtiene:

$$\int_{\Omega} ((\text{Div}(\omega S) - S\nabla\omega) + \omega \mathbf{b}_0) \, d\Omega = \int_{\Omega} \text{Div}(\omega S) \, d\Omega - \int_{\Omega} (S\nabla\omega - \omega \mathbf{b}_0) \, d\Omega = 0 .$$

Ahora, empleando el teorema de la divergencia [9, p. 37] en la primera integral, tenemos:

$$\int_{\partial\Omega} \omega S\mathbf{n} \, d\partial\Omega - \int_{\Omega} (S\nabla\omega - \omega \mathbf{b}_0) \, d\Omega = 0 .$$

Aquí se introducen las condiciones de frontera naturales del problema. Sean $\hat{\mathbf{s}} : \partial\Omega_1 \rightarrow V$ el campo vectorial de fuerza superficial sobre la frontera $\partial\Omega_1$ y $\mathbf{n} : \partial\Omega \rightarrow V$ el campo vectorial unitario normal a $\partial\Omega$. Así pues:

$$S\mathbf{n} = \hat{\mathbf{s}}$$

son condiciones Neumann del modelo físico, con lo que llegamos a la expresión:

$$\int_{\partial\Omega_1} \omega \hat{\mathbf{s}} \, d\partial\Omega - \int_{\Omega} (S\nabla\omega - \omega \mathbf{b}_0) \, d\Omega = 0, \quad (2)$$

a la cual llamaremos formulación variacional del modelo físico (1).

Para materiales isotrópicos, elásticos, lineales y homogéneos, el campo tensorial de esfuerzo de Piola-Kirchhoff S puede representarse por la siguiente expresión:

$$S = 2\mu E + \lambda \text{tr}E I$$

donde μ y λ son constantes conocidas como coeficientes de Lamé y E es el campo tensorial de deformación infinitesimal¹.

Dado un marco de coordenadas cartesianas en el espacio de trabajo de la elastodinámica bidimensional, podemos conocer las coordenadas de los campos tensoriales E y S de la siguiente manera:

$$E = \frac{1}{2} (\nabla u + \nabla^T u) = \frac{1}{2} \begin{pmatrix} 2 \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} & 2 \frac{\partial v}{\partial y} \end{pmatrix}. \quad (3)$$

Entonces:

$$S = \mu \begin{pmatrix} 2 \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} & 2 \frac{\partial v}{\partial y} \end{pmatrix} + \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$S = \begin{pmatrix} (2\mu + \lambda) \frac{\partial u}{\partial x} + \lambda \frac{\partial v}{\partial y} & \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\ \mu \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) & (2\mu + \lambda) \frac{\partial v}{\partial y} + \lambda \frac{\partial u}{\partial x} \end{pmatrix}. \quad (4)$$

Formulación de elemento finito.

Supóngase que el campo vectorial de desplazamientos u es aproximado por dos campos escalares $u, v : \Omega \rightarrow \mathfrak{R}$ tales que:

$$u = u e_1 + v e_2$$

donde

$$u = \sum_{j=1}^n u_j \varphi_j \quad y \quad v = \sum_{j=1}^n v_j \varphi_j. \quad (5)$$

¹ 1.5 Elasticidad lineal. Derivación de la teoría lineal, p. 10.

Aquí, $u_j, v_j \in \mathfrak{R}$ son los valores de las funciones u y v en el punto $(x_j, y_j) \in \Omega$, $\varphi_i : \Omega \rightarrow \mathfrak{R}$ son funciones globales de interpolación definidas sobre una discretización Ω' de Ω , y el conjunto $\{e_1, e_2\}$ forma la base canónica en \mathfrak{R}^2 .

La expresión (2) es válida para toda función de prueba ω , particularmente para las funciones de interpolación φ_i^2 . Sustituyendo en la formulación variacional (2), tenemos:

$$\int_{\partial\Omega_1} \varphi_i \delta d\partial\Omega - \int_{\Omega} (S \nabla \varphi_i - \varphi_i b_0) d\Omega = 0,$$

expresión que es equivalente a:

$$\int_{\partial\Omega_1} \varphi_i \delta d\partial\Omega + \int_{\Omega} \varphi_i b_0 d\Omega - \int_{\Omega} S \nabla \varphi_i d\Omega = 0, \quad (6)$$

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

Por otro lado, al sustituir las ecuaciones (5) en la ecuación (4) resulta:

$$S = \begin{pmatrix} (2\mu + \lambda) \frac{\partial}{\partial x} \sum_{j=1}^n u_j \varphi_j + \lambda \frac{\partial}{\partial y} \sum_{j=1}^n v_j \varphi_j & \mu \left(\frac{\partial}{\partial y} \sum_{j=1}^n u_j \varphi_j + \frac{\partial}{\partial x} \sum_{j=1}^n v_j \varphi_j \right) \\ \mu \left(\frac{\partial}{\partial x} \sum_{j=1}^n v_j \varphi_j + \frac{\partial}{\partial y} \sum_{j=1}^n u_j \varphi_j \right) & (2\mu + \lambda) \frac{\partial}{\partial y} \sum_{j=1}^n v_j \varphi_j + \lambda \frac{\partial}{\partial x} \sum_{j=1}^n u_j \varphi_j \end{pmatrix}.$$

Por la linealidad de la derivada y el hecho de que u_j y v_j son escalares, tenemos:

$$S = \begin{pmatrix} (2\mu + \lambda) \sum_{j=1}^n u_j \frac{\partial \varphi_j}{\partial x} + \lambda \sum_{j=1}^n v_j \frac{\partial \varphi_j}{\partial y} & \mu \left(\sum_{j=1}^n u_j \frac{\partial \varphi_j}{\partial y} + \sum_{j=1}^n v_j \frac{\partial \varphi_j}{\partial x} \right) \\ \mu \left(\sum_{j=1}^n v_j \frac{\partial \varphi_j}{\partial x} + \sum_{j=1}^n u_j \frac{\partial \varphi_j}{\partial y} \right) & (2\mu + \lambda) \sum_{j=1}^n v_j \frac{\partial \varphi_j}{\partial y} + \lambda \sum_{j=1}^n u_j \frac{\partial \varphi_j}{\partial x} \end{pmatrix},$$

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

Ahora, puesto que φ_i es una función de valor escalar, su gradiente está dado por:

$$\nabla \varphi_i = \begin{pmatrix} \frac{\partial \varphi_i}{\partial x} \\ \frac{\partial \varphi_i}{\partial y} \end{pmatrix}.$$

² 2.4 Planteamientos teóricos generales para la aplicación del método de elemento finito. Método de Ritz, p. 25.

Aplicando el campo tensorial S sobre este campo vectorial $\nabla\varphi_i$, resulta:

$$S\nabla\varphi_i = \left(\begin{array}{l} \left[(2\mu + \lambda) \sum_{j=1}^n u_j \frac{\partial\varphi_j}{\partial x} + \lambda \sum_{j=1}^n v_j \frac{\partial\varphi_j}{\partial y} \right] \frac{\partial\varphi_i}{\partial x} + \mu \left(\sum_{j=1}^n u_j \frac{\partial\varphi_j}{\partial y} + \sum_{j=1}^n v_j \frac{\partial\varphi_j}{\partial x} \right) \frac{\partial\varphi_i}{\partial y} \\ \mu \left(\sum_{j=1}^n v_j \frac{\partial\varphi_j}{\partial x} + \sum_{j=1}^n u_j \frac{\partial\varphi_j}{\partial y} \right) \frac{\partial\varphi_i}{\partial x} + \left[(2\mu + \lambda) \sum_{j=1}^n v_j \frac{\partial\varphi_j}{\partial y} + \lambda \sum_{j=1}^n u_j \frac{\partial\varphi_j}{\partial x} \right] \frac{\partial\varphi_i}{\partial y} \end{array} \right),$$

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

$$S\nabla\varphi_i = \left(\begin{array}{l} \sum_{j=1}^n \left((2\mu + \lambda) \frac{\partial\varphi_j}{\partial x} \frac{\partial\varphi_i}{\partial x} + \mu \frac{\partial\varphi_j}{\partial y} \frac{\partial\varphi_i}{\partial y} \right) u_j + \sum_{j=1}^n \left(\lambda \frac{\partial\varphi_j}{\partial y} \frac{\partial\varphi_i}{\partial x} + \mu \frac{\partial\varphi_j}{\partial x} \frac{\partial\varphi_i}{\partial y} \right) v_j \\ \sum_{j=1}^n \left(\lambda \frac{\partial\varphi_j}{\partial x} \frac{\partial\varphi_i}{\partial y} + \mu \frac{\partial\varphi_j}{\partial y} \frac{\partial\varphi_i}{\partial x} \right) u_j + \sum_{j=1}^n \left((2\mu + \lambda) \frac{\partial\varphi_j}{\partial y} \frac{\partial\varphi_i}{\partial y} + \mu \frac{\partial\varphi_j}{\partial x} \frac{\partial\varphi_i}{\partial x} \right) v_j \end{array} \right),$$

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

$$S\nabla\varphi_i = \sum_{j=1}^n \left(\begin{array}{cc} (2\mu + \lambda) \frac{\partial\varphi_j}{\partial x} \frac{\partial\varphi_i}{\partial x} + \mu \frac{\partial\varphi_j}{\partial y} \frac{\partial\varphi_i}{\partial y} & \lambda \frac{\partial\varphi_j}{\partial y} \frac{\partial\varphi_i}{\partial x} + \mu \frac{\partial\varphi_j}{\partial x} \frac{\partial\varphi_i}{\partial y} \\ \lambda \frac{\partial\varphi_j}{\partial x} \frac{\partial\varphi_i}{\partial y} + \mu \frac{\partial\varphi_j}{\partial y} \frac{\partial\varphi_i}{\partial x} & (2\mu + \lambda) \frac{\partial\varphi_j}{\partial y} \frac{\partial\varphi_i}{\partial y} + \mu \frac{\partial\varphi_j}{\partial x} \frac{\partial\varphi_i}{\partial x} \end{array} \right) \begin{pmatrix} u_j \\ v_j \end{pmatrix}, \quad (7)$$

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

Ahora, sustituyendo la ecuación (7) en la ecuación (6) tenemos:

$$\int_{\partial\Omega_i} \varphi_i \delta d\Omega + \int_{\Omega} \varphi_i b_o d\Omega -$$

$$\int_{\Omega} \sum_{j=1}^n \left(\begin{array}{cc} (2\mu + \lambda) \frac{\partial\varphi_j}{\partial x} \frac{\partial\varphi_i}{\partial x} + \mu \frac{\partial\varphi_j}{\partial y} \frac{\partial\varphi_i}{\partial y} & \lambda \frac{\partial\varphi_j}{\partial y} \frac{\partial\varphi_i}{\partial x} + \mu \frac{\partial\varphi_j}{\partial x} \frac{\partial\varphi_i}{\partial y} \\ \lambda \frac{\partial\varphi_j}{\partial x} \frac{\partial\varphi_i}{\partial y} + \mu \frac{\partial\varphi_j}{\partial y} \frac{\partial\varphi_i}{\partial x} & (2\mu + \lambda) \frac{\partial\varphi_j}{\partial y} \frac{\partial\varphi_i}{\partial y} + \mu \frac{\partial\varphi_j}{\partial x} \frac{\partial\varphi_i}{\partial x} \end{array} \right) \begin{pmatrix} u_j \\ v_j \end{pmatrix} d\Omega = 0;$$

puesto que u_j y v_j son escalares, la expresión anterior queda de la siguiente forma:

$$\int_{\Omega} \sum_{j=1}^n \left(\begin{array}{cc} (2\mu + \lambda) \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial x} + \mu \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial y} & \lambda \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial x} + \mu \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial y} \\ \lambda \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial y} + \mu \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial x} & (2\mu + \lambda) \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial y} + \mu \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial x} \end{array} \right) d\Omega \begin{pmatrix} u_j \\ v_j \end{pmatrix} = \quad (8)$$

$$\int_{\partial\Omega_1} \varphi_i s d\partial\Omega + \int_{\Omega} \varphi_i b_0 d\Omega ,$$

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

a la que llamaremos formulación de elemento finito del modelo físico (1).

Modelo numérico equivalente.

La ecuación (8) representa, para cada i , un conjunto de 2 ecuaciones lineales algebraicas con $2n$ incógnitas ($u_j, v_j, j = 1, 2, \dots, n$). Así pues, el modelo numérico equivalente global se define por el conjunto de $2n$ ecuaciones lineales algebraicas con $2n$ incógnitas, cuya representación matricial es:

$$\mathbf{K} \mathbf{u} = \mathbf{f} \quad (9)$$

donde:

$$K_{ij} = \int_{\Omega} \sum_{j=1}^n \left(\begin{array}{cc} (2\mu + \lambda) \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial x} + \mu \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial y} & \lambda \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial x} + \mu \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial y} \\ \lambda \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial y} + \mu \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial x} & (2\mu + \lambda) \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial y} + \mu \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial x} \end{array} \right) d\Omega ,$$

$$f_i = \int_{\partial\Omega_1} \varphi_i s d\partial\Omega + \int_{\Omega} \varphi_i b_0 d\Omega , \quad (10)$$

$$\mathbf{u} = \begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ \vdots \\ u_n \\ v_n \end{pmatrix} .$$

El problema de resolver el modelo numérico equivalente (9) es encontrar los elementos de la matriz columna \mathbf{u} , los cuales son los escalares de las combinaciones lineales (5) que aproximan a los campos escalares u y v , los que a su vez, aproximan al campo vectorial de desplazamientos \mathbf{u} . Físicamente, estos valores representan los desplazamientos en los puntos $(x_j, y_j) \in \Omega$ debidos a la geometría del cuerpo, así como a las condiciones iniciales y de carga presentes en el mismo.

Si la matriz \mathbf{K} es no singular, la solución de (9) está dada por:

$$\mathbf{u} = \mathbf{K}^{-1} \mathbf{f}.$$

3.2 PROBLEMA DE ELASTODINÁMICA LINEAL BIDIMENSIONAL.

Aproximación en el tiempo.

En problemas dependientes del tiempo se asume como hipótesis que el campo vectorial de desplazamientos \mathbf{u} es la composición de dos funciones, una que depende sólo del tiempo y una que depende sólo del espacio³, esto es:

$$\mathbf{u}(\mathbf{p}, t) = \mathbf{u}(t) \circ \mathbf{u}(\mathbf{p}) \quad \mathbf{p} \in \Omega,$$

donde $\circ(f, g) = f \circ g$ es la función composición $\mathcal{U} \rightarrow \mathcal{W}$ para las funciones $f: \mathcal{V} \rightarrow \mathcal{W}$, $g: \mathcal{U} \rightarrow \mathcal{V}$. Para nuestro caso consideraremos la función composición como la multiplicación de funciones:

$$\mathbf{u}(\mathbf{p}, t) = \mathbf{u}(t) \mathbf{u}(\mathbf{p}). \quad (11)$$

De esta manera resulta que

$$\frac{\partial^n}{\partial t^n} \mathbf{u}(\mathbf{p}, t) = \frac{\partial^n \mathbf{u}(t)}{\partial t^n} \mathbf{u}(\mathbf{p})$$

lo que implica que en la formulación de elemento finito se requieran dos aproximaciones: una para el problema espacial y la otra para el problema temporal o dinámico. Considérese que el campo vectorial de desplazamientos \mathbf{u} es aproximado de manera similar que en la ecuación (5), pero considerando ahora los campos escalares $u, v: \Omega \times \mathfrak{R} \rightarrow \mathfrak{R}$, esto es:

$$u(\mathbf{p}, t) = u(t)u(\mathbf{p}) = \sum_{j=1}^n u_j(t) \varphi_j(\mathbf{p})$$

y

(12)

³ Esta hipótesis no está respaldada más que por el hecho de que en los casos que se han resuelto de esta manera, la solución obtenida es satisfactoria. Si esta hipótesis no funcionara satisfactoriamente, se escogería otra forma de la ecuación (11) [4, p. 384].

$$v(\mathbf{p}, t) = v(t)v(\mathbf{p}) = \sum_{j=1}^n v_j(t) \varphi_j(\mathbf{p}) ,$$

donde $u_j(t), v_j(t) \in \mathfrak{R}$ son los valores de las funciones u y v en el punto $(x_j, y_j) \in \Omega$ al tiempo t , y $\varphi_j : \Omega \rightarrow \mathfrak{R}$ son funciones globales de interpolación definidas sobre una discretización Ω' de Ω . De aquí se puede ver que en la formulación de elemento finito del problema dinámico, las incógnitas del problema son los valores $u_j(t)$ y $v_j(t)$ para cada tiempo t .

Definición.

El problema de encontrar el **proceso elástico** $[u, E, S]$ correspondiente a un cuerpo elástico que ocupa una región Ω del espacio, sujeto a un sistema de fuerzas dinámicas $(\hat{\mathbf{s}}, \mathbf{b}_0)$, está definido por el problema asociado de elastodinámica por medio del modelo físico en ecuaciones diferenciales parciales (13), conocido como formulación fuerte del problema.

Formulación fuerte.

Sea $\text{Div} : L(V, V) \rightarrow V$ el operador divergencia entre campos materiales:

$$\begin{aligned} \text{Div } S + \mathbf{b}_0 &= \rho_0 \ddot{\mathbf{u}} \text{ en } \Omega, \\ S\mathbf{n} &= \hat{\mathbf{s}} \text{ en } \partial\Omega_1, \\ \mathbf{u} &= \mathbf{u}_0 \text{ en } \partial\Omega_2, \\ \dot{\mathbf{u}}(\mathbf{p}, 0) &= \mathbf{v}_0(\mathbf{p}), \end{aligned} \tag{13}$$

donde

- $\hat{\mathbf{s}} : \partial\Omega_1 \times \mathcal{T} \rightarrow V$ es el campo vectorial de fuerza superficial,
- $\mathbf{b}_0 : \Omega \times \mathfrak{R} \rightarrow V$ es el campo vectorial de fuerza de cuerpo,
- $\rho_0 : \Omega \rightarrow \mathfrak{R}^+$ es el campo de densidad de referencia,
- $\mathbf{u} : \Omega \times \mathfrak{R} \rightarrow V$ es el campo vectorial de desplazamientos,
- $\dot{\mathbf{u}} : \Omega \times \mathfrak{R} \rightarrow V$ es el campo vectorial de velocidad,
- $\ddot{\mathbf{u}} : \Omega \times \mathfrak{R} \rightarrow V$ es el campo vectorial de aceleración.

Puesto que el problema estático ya fue analizado en la sección anterior, de aquí en adelante sólo analizaremos el lado derecho de la ecuación (13). En el lado izquierdo sólo debe hacerse la consideración de que ahora el campo vectorial de desplazamientos es también función del tiempo. Los siguientes desarrollos son paralelos a los correspondientes de la sección anterior.

Formulación variacional.

Sea $\omega : \mathcal{E} \rightarrow \mathfrak{R}$ una función de prueba, multiplicando sólo el lado derecho de la ecuación (13.1) por ω , en donde se aplica la hipótesis (11) y se integra sobre todo el dominio Ω tenemos:

$$\int_{\Omega} \omega \rho_0 \ddot{u}(p, t) d\Omega = \int_{\Omega} \omega \rho_0 \ddot{u}(t) u(p) d\Omega, \quad (14)$$

expresión que corresponde a la formulación variacional (2) de la sección anterior.

Formulación de elemento finito.

Si se aproxima el campo vectorial de desplazamientos \mathbf{u} por las ecuaciones (12) y se sustituye ω por alguna de las funciones globales de interpolación en la ecuación (14), tomado en cuenta el método de Ritz, tenemos:

$$\begin{aligned} \int_{\Omega} \omega \rho_0 \begin{pmatrix} \ddot{u}(t) u(p) \\ \ddot{v}(t) v(p) \end{pmatrix} d\Omega &= \int_{\Omega} \varphi_i(p) \rho_0 \begin{pmatrix} \sum_{j=1}^n \ddot{u}_j(t) \varphi_j(p) \\ \sum_{j=1}^n \ddot{v}_j(t) \varphi_j(p) \end{pmatrix} d\Omega & \quad i = 1, 2, \dots, n; \\ &= \int_{\Omega} \varphi_i(p) \rho_0 \begin{pmatrix} \sum_{j=1}^n \varphi_j(p) & 0 \\ 0 & \sum_{j=1}^n \varphi_j(p) \end{pmatrix} \begin{pmatrix} \ddot{u}_j(t) \\ \ddot{v}_j(t) \end{pmatrix} d\Omega \\ & \quad i = 1, 2, \dots, n. \end{aligned}$$

Ahora, puesto que $u_j(t)$ y $v_j(t)$ son independientes del espacio, así lo son sus derivadas. Entonces es posible llegar a la siguiente expresión:

$$\int_{\Omega} \omega \rho_0 \begin{pmatrix} \ddot{u}(t) u(p) \\ \ddot{v}(t) v(p) \end{pmatrix} d\Omega = \int_{\Omega} \sum_{j=1}^n \rho_0 \varphi_i(p) \varphi_j(p) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} d\Omega \begin{pmatrix} \ddot{u}_j(t) \\ \ddot{v}_j(t) \end{pmatrix} \quad (15)$$

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

la cual corresponde a la formulación de elemento finito (8) de la sección anterior.

Modelo numérico equivalente.

El lado derecho de la ecuación (21) puede ser sintetizado en forma matricial como sigue:

$$\mathbf{M} \ddot{\mathbf{u}}(t) = \int_{\Omega} \sum_{j=1}^n \rho_0 \varphi_i(p) \varphi_j(p) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} d\Omega \begin{pmatrix} \ddot{u}_j(t) \\ \ddot{v}_j(t) \end{pmatrix} \quad (16)$$

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

Para cada i , la matriz \mathbf{M} y el vector $\ddot{\mathbf{u}}(t)$ definen 2 combinaciones lineales algebraicas con $2n$ incógnitas ($\ddot{u}_j(t)$ y $\ddot{v}_j(t)$), $j = 1, 2, \dots, n$, por lo que en total se tienen $2n$ combinaciones lineales con $2n$ incógnitas. Este resultado concuerda con el obtenido para el caso estático en la sección anterior por la ecuación (8).

Así pues, el modelo numérico equivalente del problema elastodinámico lineal bidimensional queda definido por el sistema de ecuaciones lineales:

$$\mathbf{M} \ddot{\mathbf{u}}(t) + \mathbf{K} \mathbf{u}(t) = \mathbf{f}(t), \quad (17)$$

donde:

$$M_{ij} = \int_{\Omega} \sum_{j=1}^n \rho_0 \varphi_i \varphi_j \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} d\Omega,$$

$$K_{ij} = \int_{\Omega} \sum_{j=1}^n \left(\begin{array}{cc} (2\mu + \lambda) \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial x} + \mu \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial y} & \lambda \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial x} + \mu \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial y} \\ \lambda \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial y} + \mu \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial x} & (2\mu + \lambda) \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial y} + \mu \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial x} \end{array} \right) d\Omega,$$

$$f_i(t) = \int_{\Omega_1} \varphi_i s(p, t) d\Omega + \int_{\Omega} \varphi_i b_0(p, t) d\Omega,$$

$$\mathbf{u}(t) = \begin{pmatrix} u_1(t) \\ v_1(t) \\ u_2(t) \\ v_2(t) \\ \vdots \\ \vdots \\ u_n(t) \\ v_n(t) \end{pmatrix} \quad \text{y} \quad \mathbf{u}(t) = \begin{pmatrix} u_1(t) \\ v_1(t) \\ u_2(t) \\ v_2(t) \\ \vdots \\ \vdots \\ u_n(t) \\ v_n(t) \end{pmatrix}.$$

Esquemas de aproximación para la derivada temporal.

Una vez que ha sido aproximado el problema espacial es necesario establecer un esquema de aproximación a la segunda derivada temporal del campo vectorial de desplazamientos \mathbf{u} y resolver la ecuación (17), la cual debe ser resuelta para los desplazamientos puntuales en cada paso del tiempo. Para nuestro problema específico vamos a considerar un esquema de diferencias finitas con tres puntos de recurrencia⁴, particularmente el método de Crank-Nicolson. Para su implantación se debe llevar la ecuación (17) a un conjunto de dos ecuaciones diferenciales de primer orden:

$$\begin{aligned} \mathbf{M} \dot{\mathbf{v}}(t) + \mathbf{K} \mathbf{u}(t) &= \mathbf{f}(t) \\ \dot{\mathbf{u}}(t) &= \mathbf{v}(t) \end{aligned} \quad (18)$$

Considérense las siguientes aproximaciones:

$$\begin{aligned} \dot{\mathbf{v}}(t) &= \frac{d\mathbf{v}(t)}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\mathbf{v}(t + \Delta t) - \mathbf{v}(t)}{\Delta t} \cong \frac{\mathbf{v}(t + \Delta t) - \mathbf{v}(t)}{\Delta t} = \frac{\mathbf{v}_i - \mathbf{v}_{i-1}}{\Delta t}, \\ \mathbf{v}(t) &= \frac{\mathbf{v}_i + \mathbf{v}_{i-1}}{2}, \quad \mathbf{u}(t) = \frac{\mathbf{u}_i + \mathbf{u}_{i-1}}{2}, \quad \mathbf{f}(t) = \frac{\mathbf{f}_i + \mathbf{f}_{i-1}}{2}. \end{aligned}$$

Entonces, por la ecuación (18.2) tenemos que:

$$\frac{\mathbf{u}_i - \mathbf{u}_{i-1}}{\Delta t} = \frac{\mathbf{v}_i + \mathbf{v}_{i-1}}{2}. \quad (19)$$

Sustituyendo lo anterior en la ecuación (24.1) resulta:

$$\mathbf{M} \frac{\mathbf{v}_i - \mathbf{v}_{i-1}}{\Delta t} + \mathbf{K} \frac{\mathbf{u}_i + \mathbf{u}_{i-1}}{2} = \frac{\mathbf{f}_i + \mathbf{f}_{i-1}}{2} \quad (20)$$

Las componentes de la matriz columna $\mathbf{v}_i = \mathbf{v}(t + \Delta t)$ representan las velocidades puntuales al final de cada paso de tiempo Δt , las cuales son desconocidas al inicio de la solución. Sin embargo, es posible eliminar ésta utilizando la ecuación (19), esto es:

$$\mathbf{v}_i = 2 \frac{\mathbf{u}_i - \mathbf{u}_{i-1}}{\Delta t} - \mathbf{v}_{i-1}. \quad (21)$$

Sustituyendo el resultado anterior en la ecuación (20):

⁴ Apéndice A-2: *Esquemas de aproximación para la derivada temporal*, p. A-6.

$$M \left[2 \frac{u_i - u_{i-1} - \Delta t v_{i-1}}{\Delta t^2} \right] + K \frac{u_i + u_{i-1}}{2} = \frac{f_i + f_{i-1}}{2},$$

expresión que es equivalente a:

$$M \left[2 \frac{u_i - u_{i-1} - \Delta t v_{i-1}}{\Delta t} \right] + K \Delta t \frac{u_i + u_{i-1}}{2} = \Delta t \frac{f_i + f_{i-1}}{2}.$$

Finalmente, agrupando términos semejantes:

$$\left(\frac{2}{\Delta t} M + \frac{\Delta t}{2} K \right) u_i - \left(\frac{2}{\Delta t} M - \frac{\Delta t}{2} K \right) u_{i-1} - 2M v_{i-1} = \frac{\Delta t}{2} f_i + \frac{\Delta t}{2} f_{i-1} \quad (22)$$

o

$$\left(\frac{2}{\Delta t} M + \frac{\Delta t}{2} K \right) u(t + \Delta t) - \left(\frac{2}{\Delta t} M - \frac{\Delta t}{2} K \right) u(t) - 2M v(t) = \frac{\Delta t}{2} f(t + \Delta t) + \frac{\Delta t}{2} f(t).$$

Con la ecuación (22) queda definido un esquema recursivo de solución desde la matriz columna inicial $u_i = u_1 = u(\Delta t)$, para la cual tenemos:

$u_{i-1} = u(0)$: solución del problema estático,

$v_{i-1} = v(0)$: condiciones iniciales del problema,

$f_{i-1} = f(0)$: fuerzas puntuales iniciales,

$f_i = f(\Delta t)$: fuerzas puntuales al tiempo Δt .

(23)

Las expresiones en (23), que constituyen el proceso de inicio o arranque, requieren de la solución del caso estático para empezar a trabajar.

El problema de resolver la ecuación (22) es encontrar los elementos $u_j(t)$ y $v_j(t)$ de la matriz columna $u(t)$ para cada paso del tiempo Δt , tales valores son los escalares de las combinaciones lineales (12). Físicamente, estos valores representan los desplazamientos del punto $(x_j, y_j) \in \Omega_t$ en cada paso de tiempo Δt , debidos a la geometría del cuerpo, así como a las condiciones iniciales y de carga presentes en éste.

De la ecuación (22), sean:

$$\tilde{\mathbf{K}} = \left(\frac{2}{\Delta t} \mathbf{M} + \frac{\Delta t}{2} \mathbf{K} \right),$$

y

$$\tilde{\mathbf{f}}(t) = \left(\frac{2}{\Delta t} \mathbf{M} - \frac{\Delta t}{2} \mathbf{K} \right) \mathbf{u}(t) + 2\mathbf{M}\mathbf{v}(t) + \frac{\Delta t}{2} \mathbf{f}(t + \Delta t) + \frac{\Delta t}{2} \mathbf{f}(t), \quad (24)$$

si la matriz $\tilde{\mathbf{K}}$ es no singular, la solución del modelo numérico equivalente (22) está dada por:

$$\mathbf{u}(t + \Delta t) = \tilde{\mathbf{K}}^{-1} \tilde{\mathbf{f}}(t), \quad t \geq 0. \quad (25)$$

Capítulo 4:

Caso de Estudio: Vibración de una viga.

4.1 DEFINICIÓN DEL PROBLEMA.

Planteamiento.

Una viga de acero con un extremo fijo, de longitud L , ancho a , y espesor b , soporta un motor eléctrico de masa m que gira a una velocidad de N rpm, en su extremo libre. La fuerza de excitación que el motor ejerce sobre la viga es de la forma $F(t) = F_0 \sin \omega t$. Establézcanse los modelos numéricos para determinar la evolución dinámica del campo de desplazamientos desarrollados en la viga durante un período de la vibración natural asociada al problema, $t \in [0, \tau_n]$.



fig. 4-1: Caso de Estudio: Vibración de una viga.

Consideraciones.

El cuerpo descrito en el planteamiento está definido en un espacio de tres dimensiones. Sin embargo, el cuerpo cumple con la definición y las restricciones de cuerpo plano [6, pp. 59]. La solución no será exacta puesto que las condiciones de frontera sobre las superficies que definen el cuerpo plano, no son satisfechas rigurosamente por las ecuaciones de campo [6, pp. 60-63]. Sin embargo, para regiones del cuerpo suficientemente alejadas de las fronteras del cuerpo, la solución es aceptable.

Discretización del dominio.

Sea

$$\Omega = \{ (p_1, p_2) \in \mathbb{R}^2 \mid p_1 \in [0, L], p_2 \in [0, a] \} \subset \mathcal{E}$$

una región del espacio euclidiano que defina el dominio del problema. Considérese sobre éste la discretización formada por un conjunto finito de triángulos rectángulos de tres nodos, llamados elementos finitos geométricos, construidos a partir del elemento finito de referencia definido por:

$$\text{EFR} = \{ (x, y) \in \mathbb{R}^2 \mid y \leq 1 - x, 0 \leq x \leq 1, 0 \leq y \leq 1 \}.$$

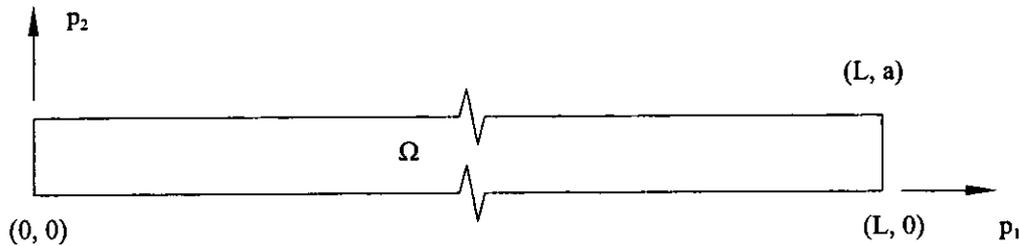


fig 4-2: Dominio bidimensional del problema.

Para ilustrar la discretización del dominio, se propone una malla formada por 22 elementos finitos triangulares y 24 nodos globales. Puesto que cada nodo tiene dos grados de libertad

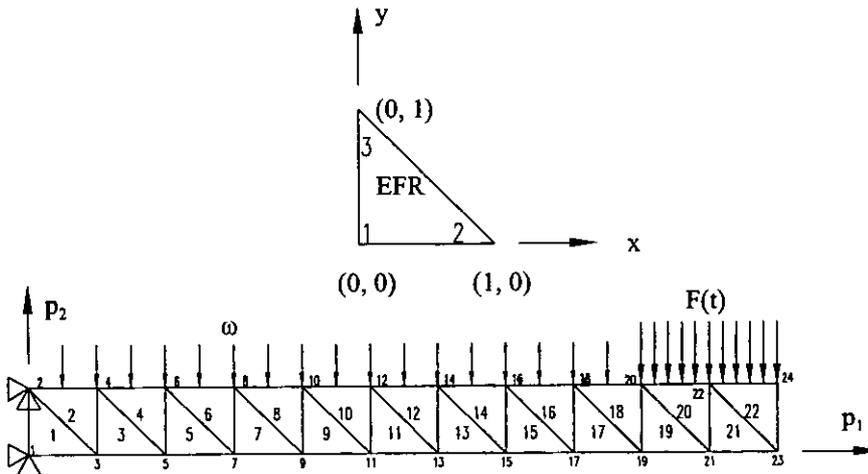


fig. 4-3: Discretización del dominio.

(desplazamientos en dirección p_1 y p_2 en cada nodo), y dos de ellos tienen restringido el movimiento, el problema completo es de 44 grados de libertad. La numeración global de elementos y nodos es como se muestra en la fig. (4-3).

4.2 ANÁLISIS DE ELEMENTO FINITO.

Funciones locales de interpolación.

De acuerdo con lo discutido sobre el tema de interpolación¹, dada una discretización de una región regular del espacio euclidiano bidimensional, como la que se ha propuesto en este problema, sobre el i -ésimo elemento finito geométrico con vértices (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , las funciones de interpolación $\varphi_{i1}(x, y)$, $\varphi_{i2}(x, y)$, $\varphi_{i3}(x, y)$, cuyo soporte es el i -ésimo triángulo, definidas como:

$$\begin{aligned} \text{a) } \varphi_{i1}(x_1, y_1) &= 1, & \varphi_{i1}(x_2, y_2) &= 0, & \varphi_{i1}(x_3, y_3) &= 0, \\ \text{b) } \varphi_{i2}(x_1, y_1) &= 0, & \varphi_{i2}(x_2, y_2) &= 1, & \varphi_{i2}(x_3, y_3) &= 0, \\ \text{c) } \varphi_{i3}(x_1, y_1) &= 0, & \varphi_{i3}(x_2, y_2) &= 0, & \varphi_{i3}(x_3, y_3) &= 1; \end{aligned} \quad (1)$$

forman una base mínima del conjunto de polinomios de grado uno y forma:

$$F = \{p : \mathfrak{R} \times \mathfrak{R} \rightarrow \mathfrak{R} \mid p(x, y) = Ax + By + C, A, B, C \in \mathfrak{R}\}.$$

Al resolver el sistema de ecuaciones (1) resulta que:

$$\begin{aligned} \varphi_{i1}(x, y) &= \frac{1}{A} \left[(y_3 - y_2)x + (x_2 - x_3)y + x_3y_2 - x_2y_3 \right], \\ \varphi_{i2}(x, y) &= \frac{1}{A} \left[(y_1 - y_3)x + (x_3 - x_1)y + x_1y_3 - x_3y_1 \right], \\ \varphi_{i3}(x, y) &= \frac{1}{A} \left[(y_2 - y_1)x + (x_1 - x_2)y + x_2y_1 - x_1y_2 \right], \end{aligned} \quad (2)$$

donde $A = x_2y_1 + x_3y_2 + x_1y_3 - (x_3y_1 + x_1y_2 + x_2y_3) \neq 0$.

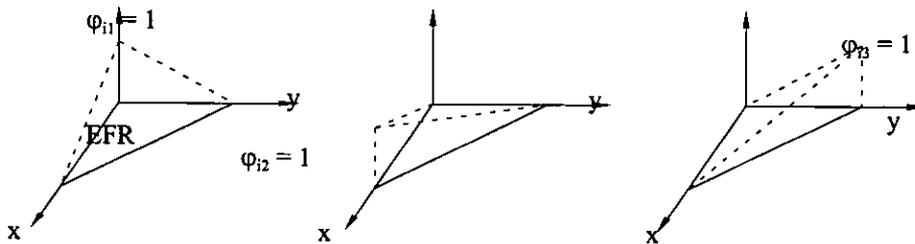


fig. 4-4: Funciones locales de interpolación.

Para nuestro elemento finito de referencia EFR, los vértices (x_1, y_1) , (x_2, y_2) , (x_3, y_3) tienen coordenadas $(0, 0)$, $(1, 0)$ y $(0, 1)$, respectivamente. Entonces, las funciones locales de interpolación son:

$$\varphi_1(x, y) = 1 - x - y,$$

¹ 1.9 Aproximaciones e interpolación, p. 20.

$$\varphi_2(x, y) = x \quad , \quad (3)$$

$$\varphi_3(x, y) = y \quad .$$

Funciones de deformación.

La discretización del dominio Ω mostrada en la fig. 4-3 se logra mediante una función de deformación del elemento finito de referencia EFR a cada elemento finito geométrico Ω_i , compuesta por una secuencia traslación-rotación-elongación². La función de deformación para cada elemento finito geométrico $f_i : \text{EFR} \rightarrow \Omega_i$, está dada por³:

$$f_i \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} q_{i1} + \lambda_{i1} \cos\theta_i x - \lambda_{i1} \text{sen}\theta_i y \\ q_{i2} + \lambda_{i2} \text{sen}\theta_i x + \lambda_{i2} \cos\theta_i y \end{pmatrix} \quad , \quad (4)$$

$$i = 1, 2, \dots, k.$$

donde, para el i-ésimo elemento finito:

- (q_{i1}, q_{i2}) son las coordenadas del punto fijo q de la deformación, que en este caso coincide con el vértice recto del elemento finito;
- θ_i es la cantidad de rotación angular alrededor de q;
- $\lambda_{i1}, \lambda_{i2}$ son las extensiones en dirección x e y desde q, respectivamente, después de la rotación del elemento finito de referencia EFR;
- $k = 1, 2, \dots, 22$ es el número total de elementos finitos geométricos de la discretización.

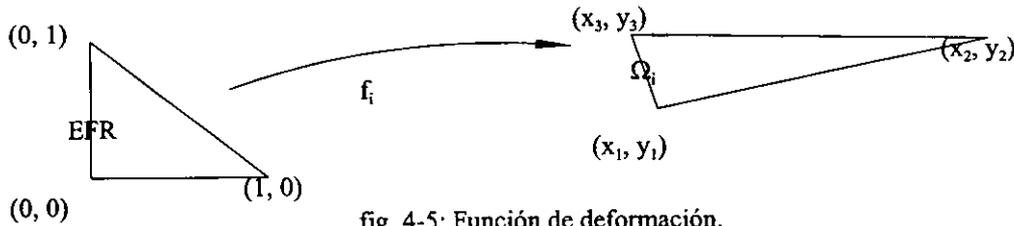


fig. 4-5: Función de deformación.

Debido a que en la discretización propuesta las cantidades de rotación θ_i son 0 o π , las funciones de deformación tienen la forma general:

$$f_i \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} q_{i1} + (-1)^{i+1} \lambda_{i1} x \\ q_{i2} + (-1)^{i+1} \lambda_{i2} y \end{pmatrix} \quad (5)$$

²1.1 Cinemática. Cuerpos y deformaciones, lema 1.1. Deformaciones homogéneas, p. 1.

³Apéndice A-1: Deformaciones básicas en coordenadas cartesianas, p. A-1.

Funciones globales de interpolación.

En la construcción de las funciones globales de interpolación se emplean las funciones locales de interpolación del elemento finito de referencia EFR y la función de deformación correspondiente al subdominio en el cual el EFR es deformado, como se muestra en la fig. 4-6.

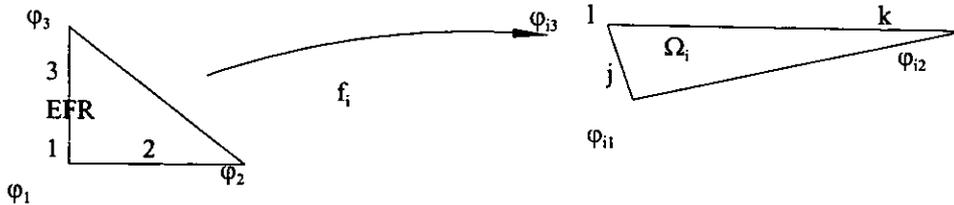


fig. 4-6: Funciones globales de interpolación.

Una forma de representar las funciones globales de interpolación es la siguiente: dados los nodos locales 1, 2 y 3 del *i*-ésimo elemento finito de la discretización, las funciones globales de interpolación quedan representadas por un doble subíndice de acuerdo con la siguiente tabla:

Elemento finito	Nodo local (EFR)	Función local de interpolación	Nodo global Ω_i	Función global de interpolación.
Ω_i	1	φ_1	j	$\varphi_j = \varphi_{i1}$
	2	φ_2	k	$\varphi_k = \varphi_{i2}$
	3	φ_3	l	$\varphi_l = \varphi_{i3}$

Tabla 4-1: Relación entre las funciones de interpolación locales y globales.

Puesto que las funciones locales de interpolación están definidas sobre el elemento finito de referencia EFR:

$$\varphi_j : \text{EFR} \rightarrow \mathfrak{R}, \quad j = 1, 2, 3;$$

las funciones globales de interpolación deben estar definidas sobre cada uno de los subdominios del problema:

$$\varphi_{ij} : \Omega_i \rightarrow \mathfrak{R}.$$

Entonces las funciones globales de interpolación quedan definidas como:

$$\varphi_{ij} = \varphi_j \circ f_i^{-1}. \tag{6}$$

Sea $f_i : \text{EFR} \rightarrow \Omega_i$, la *i*-ésima función de deformación, su inversa, $f_i^{-1} : \Omega_i \rightarrow \text{EFR}$ es tal que, si:

$$f_i \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} q_{i1} + (-1)^{i+1} \lambda_{i1} x \\ q_{i2} + (-1)^{i+1} \lambda_{i2} y \end{pmatrix} = p \in \Omega_i$$

y

$$f_i^{-1}(p) = f_i^{-1} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} \in \text{EFR} ,$$

entonces

$$q_{i1} + (-1)^{i+1} \lambda_{i1} x = p_1 \quad q_{i2} + (-1)^{i+1} \lambda_{i2} y = p_2 .$$

Por lo tanto:

$$x = \frac{p_1 - q_{i1}}{(-1)^{i+1} \lambda_{i1}} \quad e \quad y = \frac{p_2 - q_{i2}}{(-1)^{i+1} \lambda_{i2}} ,$$

así que:

$$f_i^{-1} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} \frac{p_1 - q_{i1}}{(-1)^{i+1} \lambda_{i1}} \\ \frac{p_2 - q_{i2}}{(-1)^{i+1} \lambda_{i2}} \end{pmatrix} , \quad (7)$$

donde p_1 y p_2 son las direcciones del marco de coordenadas cartesianas definido sobre el dominio Ω , con la misma base canónica con la que se definió el marco de coordenadas x - y sobre el elemento finito de referencia EFR.

Considerando lo anterior, las funciones globales de interpolación $\varphi_{ij}: \Omega_i \rightarrow \mathfrak{R}$ tienen la forma:

$$\varphi_{i1}(p_1, p_2) = \varphi_1 \left(f_i^{-1} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \right) = \varphi_1 \left(\begin{pmatrix} \frac{p_1 - q_{i1}}{(-1)^{i+1} \lambda_{i1}} \\ \frac{p_2 - q_{i2}}{(-1)^{i+1} \lambda_{i2}} \end{pmatrix} \right) = 1 - \frac{p_1 - q_{i1}}{(-1)^{i+1} \lambda_{i1}} - \frac{p_2 - q_{i2}}{(-1)^{i+1} \lambda_{i2}} ,$$

$$\varphi_{i2}(p_1, p_2) = \varphi_2 \left(f_i^{-1} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \right) = \varphi_2 \left(\begin{pmatrix} \frac{p_1 - q_{i1}}{(-1)^{i+1} \lambda_{i1}} \\ \frac{p_2 - q_{i2}}{(-1)^{i+1} \lambda_{i2}} \end{pmatrix} \right) = \frac{p_1 - q_{i1}}{(-1)^{i+1} \lambda_{i1}} , \quad (8)$$

$$\varphi_{i3}(p_1, p_2) = \varphi_3 \left(f_i^{-1} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \right) = \varphi_3 \left(\begin{pmatrix} \frac{p_1 - q_{i1}}{(-1)^{i+1} \lambda_{i1}} \\ \frac{p_2 - q_{i2}}{(-1)^{i+1} \lambda_{i2}} \end{pmatrix} \right) = \frac{p_2 - q_{i2}}{(-1)^{i+1} \lambda_{i2}},$$

$$i = 1, 2, \dots, 22.$$

Otra forma de obtener las funciones globales de interpolación es directamente de su definición general, dada por las ecuaciones (2), en función de las coordenadas nodales de cada elemento finito.

Con un ejemplo mostraremos que las ecuaciones (2) y (8) son equivalentes. Sea Ω_{16} un subdominio de la discretización de Ω , sus coordenadas nodales y parámetros de deformación son, para $L = 5.5$ y $a = 0.5$:

$$\begin{aligned} (x_1, y_1) &= (4, 0.5), & (x_2, y_2) &= (3.5, 0.5), & (x_3, y_3) &= (4, 0), \\ (q_{161}, q_{162}) &= (4, 0.5), & \text{y} & & (\lambda_{161}, \lambda_{162}) &= (0.5, 0.5); \end{aligned}$$

sus funciones globales de interpolación, dadas por (2) son:

$$\varphi_{161}(x, y) = -8 + x + y,$$

$$\varphi_{162}(x, y) = 8 - 2x,$$

$$\varphi_{163}(x, y) = 1 - y$$

y por (8):

$$\varphi_{161}(p_1, p_2) = -8 + p_1 + p_2,$$

$$\varphi_{162}(p_1, p_2) = 8 - 2p_1,$$

$$\varphi_{163}(p_1, p_2) = 1 - p_2,$$

las cuales son equivalentes, excepto por el “nombre” asignado a las direcciones del marco de referencia, el cual es el mismo en ambos casos.

El procedimiento de construcción de funciones globales de interpolación propuesto en esta sección, planteado desde el punto de vista de mecánica del medio continuo, nos lleva a los mismos resultados obtenidos al construir las funciones globales de interpolación desde su

definición original (2). La ventaja de usar la definición (2) para las funciones globales de interpolación es que no necesariamente se debe conocer el tipo de deformación causada al elemento finito de referencia para construir la discretización del dominio. En cambio, con el procedimiento de esta sección es necesario conocer los parámetros de deformación, lo que podría representar algún problema para discretizaciones no uniformes del dominio.

4.3 MODELOS NUMÉRICOS DE ELEMENTO FINITO.

Modelo de la matriz \mathbf{K} .

La matriz \mathbf{K} del modelo numérico equivalente del problema elastodinámico bidimensional, fue definida en el capítulo 3 por la ecuación (10.1) como:

$$K_{ij} = \int_{\Omega} \sum_{i=1}^n \left(\begin{array}{cc} (2\mu + \lambda) \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial x} + \mu \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial y} & \lambda \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial x} + \mu \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial y} \\ \lambda \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial y} + \mu \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial x} & (2\mu + \lambda) \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial y} + \mu \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial x} \end{array} \right) d\Omega ,$$

donde φ_j son las funciones globales de interpolación y n es el número total de nodos de la discretización del dominio Ω . Debido a esa discretización, la expresión anterior es equivalente a la siguiente:

$$K_{ij} = \sum_{k=1}^h \int_{\Omega_k} \sum_{i=1}^n \left(\begin{array}{cc} (2\mu + \lambda) \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial x} + \mu \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial y} & \lambda \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial x} + \mu \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial y} \\ \lambda \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial y} + \mu \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial x} & (2\mu + \lambda) \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial y} + \mu \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial x} \end{array} \right) d\Omega_k , \quad (9)$$

donde Ω_k representa el k -ésimo subdominio o elemento finito geométrico que compone la discretización del dominio Ω y h es el número total de subdominios.

Las componentes de la matriz \mathbf{K} están formadas por los productos de las derivadas parciales de las funciones globales de interpolación (8), las cuales están dadas por las siguientes expresiones:

$$\frac{\partial \varphi_{i1}}{\partial p_1} = -\frac{1}{(-1)^{i+1} \lambda_{i1}} , \quad \frac{\partial \varphi_{i1}}{\partial p_2} = -\frac{1}{(-1)^{i+1} \lambda_{i2}} ,$$

$$\begin{aligned} \frac{\partial \varphi_{i2}}{\partial p_1} &= \frac{1}{(-1)^{i+1} \lambda_{i1}} , & \frac{\partial \varphi_{i2}}{\partial p_2} &= \frac{1}{(-1)^{i+1} \lambda_{i2}} , \\ \frac{\partial \varphi_{i3}}{\partial p_1} &= \frac{1}{(-1)^{i+1} \lambda_{i1}} , & \frac{\partial \varphi_{i3}}{\partial p_2} &= \frac{1}{(-1)^{i+1} \lambda_{i2}} , \end{aligned} \quad i = 1, 2, \dots, n;$$

o en función de las coordenadas globales, de las ecuaciones (2) se tiene que:

$$\begin{aligned} \frac{\partial \varphi_{i1}}{\partial x} &= \frac{y_3 - y_2}{A} , & \frac{\partial \varphi_{i1}}{\partial y} &= \frac{x_2 - x_3}{A} , \\ \frac{\partial \varphi_{i2}}{\partial x} &= \frac{y_1 - y_3}{A} , & \frac{\partial \varphi_{i2}}{\partial y} &= \frac{x_3 - x_1}{A} , \\ \frac{\partial \varphi_{i3}}{\partial x} &= \frac{y_2 - y_1}{A} , & \frac{\partial \varphi_{i3}}{\partial y} &= \frac{x_1 - x_2}{A} ; \end{aligned}$$

donde $A = x_2y_1 + x_3y_2 + x_1y_3 - (x_3y_1 + x_1y_2 + x_2y_3) \neq 0$ e $i = 1, 2, \dots, n$.

Para cuerpos isotrópicos y homogéneos, como es nuestro caso, los coeficientes de Lamé μ y λ son cantidades adimensionales que describen el comportamiento mecánico del material; al coeficiente μ se le conoce como módulo de rigidez y también se representa por el símbolo G . Generalmente estos coeficientes se encuentran en tablas [2, p. 699] en función de otros coeficientes mayormente conocidos como son el módulo de elasticidad E y el módulo de Poisson ν . Las ecuaciones para obtener los coeficientes de Lamé en función de los coeficientes E y ν son las siguientes [6, p. 46]:

$$\mu = \frac{E}{2(1+\nu)} \quad \text{y} \quad \lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} = \frac{\mu(2\mu - E)}{E - 3\mu} .$$

Entonces, puesto que las derivadas de las funciones de interpolación y los coeficientes de Lamé son constantes, el integrando de la ecuación (9) es constante, esto es:

$$K_{ij} = \sum_{k=1}^h \sum_{\substack{j=1 \\ i=1}}^n \left(\begin{array}{cc} (2\mu + \lambda) \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial x} + \mu \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial y} & \lambda \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial x} + \mu \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial y} \\ \lambda \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial y} + \mu \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial x} & (2\mu + \lambda) \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial y} + \mu \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial x} \end{array} \right) \int_{\Omega_k} d\Omega_k$$

$$= \sum_{k=1}^h \sum_{j=1}^n \sum_{i=1}^n \left(\begin{array}{cc} (2\mu + \lambda) \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial x} + \mu \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial y} & \lambda \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial x} + \mu \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial y} \\ \lambda \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial y} + \mu \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial x} & (2\mu + \lambda) \frac{\partial \varphi_j}{\partial y} \frac{\partial \varphi_i}{\partial y} + \mu \frac{\partial \varphi_j}{\partial x} \frac{\partial \varphi_i}{\partial x} \end{array} \right) V_k \quad (10)$$

donde $V_k = \frac{1}{2} Ab$ es el volumen del k-ésimo elemento finito geométrico.

Modelo de la matriz \mathbf{M} .

La matriz \mathbf{M} quedó definida por la ecuación (16) del capítulo 3 como:

$$M_{ij} = \int_{\Omega} \sum_{j=1}^n \rho_0 \varphi_i(\mathbf{p}) \varphi_j(\mathbf{p}) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} d\Omega .$$

Otra vez, debido a la discretización del dominio Ω , la expresión anterior es equivalente a:

$$M_{ij} = \sum_{k=1}^h \int_{\Omega_k} \sum_{j=1}^n \rho_0 \varphi_i(\mathbf{p}) \varphi_j(\mathbf{p}) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} d\Omega_k . \quad (11)$$

Propiedades de las matrices de elemento finito.

Simetría, bandeado y dominancia diagonal.

Debido a que las coordenadas de las derivadas parciales de las funciones globales de interpolación son valores reales y la conmutatividad es válida entre números reales, la matriz \mathbf{K} de la ecuación (10.1) del capítulo 3 resulta ser simétrica. Por otro lado, puesto que las funciones globales de interpolación son campos de valor real y la conmutatividad es válida en el campo de los números reales, la matriz \mathbf{M} de la ecuación (17.2) del capítulo 3 también es simétrica.

Las matrices \mathbf{K} y \mathbf{M} presentan además una construcción bandeda y son diagonalmente dominantes o definidas positivas⁴.

Cálculo del vector $\mathbf{f}(\mathbf{t})$.

Las componentes del vector $\mathbf{f}(\mathbf{t})$ quedan definidas por la expresión:

$$f_i(\mathbf{t}) = \int_{\Omega} \varphi_i \hat{s} d\Omega + \int_{\Omega} \varphi_i b_0 d\Omega ,$$

⁴ Estas propiedades son discutidas en el apéndice A-4: *Solución de ecuaciones lineales simultáneas*, p. A-11.

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

donde

$\hat{s} : \partial\Omega \times \mathcal{R} \rightarrow V$ es el campo vectorial de densidad de carga superficial y
 $b_0 : \Omega \times \mathcal{R} \rightarrow V$ es el campo vectorial de densidad de fuerza de cuerpo.

Nuevamente, debido a la discretización de Ω , tenemos:

$$f_i(t) = \sum_{k=1}^h \left(\int_{\partial\Omega_k} \varphi_i \hat{s} d\partial\Omega_k + \int_{\Omega_k} \varphi_i b_0 d\Omega_k \right), \quad (12)$$

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

El campo vectorial \hat{s} es debido a la acción de fuerzas externas sobre la superficie del cuerpo. En nuestro caso, este campo vectorial es válido solamente en los elementos finitos sobre los cuales se encuentra la carga (fig. 4-3) y su valor se debe a la fuerza de excitación que causa el motor y el propio peso del mismo. Esto es:

$$\hat{s}_k(x, t) = -\frac{1}{b e_c} \begin{pmatrix} 0 \\ mg + F_0 \text{sen } \omega t \end{pmatrix} x \quad \begin{bmatrix} \text{N} \\ \text{m} \end{bmatrix} \quad x \in \partial\Omega_k$$

donde b es el ancho de la viga, e_c es el número de elementos cargados, m es la masa del motor y $g = 9.81 \text{ m/seg}^2$ es la constante de gravedad.

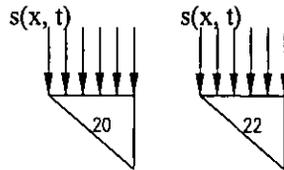


fig. 4-7: Cargas externas.

El campo vectorial b_0 es debido a la acción de fuerzas internas generadas por algún campo de naturaleza magnética, eléctrica, gravitatoria, etc. En nuestro caso este campo es válido sobre todo el dominio Ω y estará representado por el peso específico del material, con lo cual se considera el peso del cuerpo debido al campo de fuerza gravitatorio:

$$b_{0k}(x, t) = \frac{1}{b} \begin{pmatrix} 0 \\ -\gamma \end{pmatrix} \quad \begin{bmatrix} \text{N} \\ \text{m}^2 \end{bmatrix} \quad x \in \Omega_k$$

$$k = 1, 2, \dots, n;$$

Integración numérica⁵.

La densidad del cuerpo ρ_0 es una función independiente del espacio, más no así las funciones globales de interpolación. Esto significa que en el cálculo de la matriz \mathbf{M} se debe implementar la integración numérica del producto de las funciones globales de interpolación sobre dominios triangulares:

$$\int_{\Omega_k} \varphi_i(\mathbf{p})\varphi_j(\mathbf{p})d\Omega_k .$$

Asimismo, en el cálculo de las componentes del vector $\mathbf{f}(t)$ también se requiere de la integración numérica de funciones polinomiales sobre dominios triangulares. Por la forma en que hemos establecidos los campos vectoriales $\hat{\mathbf{s}}$ y \mathbf{b}_0 , las integrales que se requieren en este caso son:

$$\int_{\Omega_k} \varphi_i \bullet \mathbf{x} d\Omega_k \quad \text{y} \quad \int_{\Omega_k} \varphi_i d\Omega_k .$$

4.4 DISCRETIZACIÓN EN EL TIEMPO.

Frecuencia natural de vibración.

En el análisis dinámico de un cuerpo elástico es importante conocer su frecuencia natural para determinar el tiempo mínimo de análisis. El cuerpo con sus condiciones de frontera y sus propiedades geométricas y elásticas constituye un sistema que posee una frecuencia natural de vibración, ω_n , la cual puede ser obtenida de la siguiente manera:

$$\omega_n = 2\pi\sqrt{k/m} = 1/\tau_n \tag{13}$$

donde k es la rigidez equivalente, m es la masa del sistema y τ_n es el período natural de vibración.

Estabilidad del esquema de aproximación.

La estabilidad de los esquemas de aproximación depende en gran medida de una discretización adecuada del tiempo, de ello depende el éxito de un análisis dinámico. Una discretización del tiempo recomendada [13, pp. 233-245] es:

$$\Delta t = \frac{\tau_n}{\pi} \tag{14}$$

⁵ En el apéndice A-3: *Integración numérica sobre dominios triangulares*, p. A-8, se encuentran las expresiones resultantes de la integración de las funciones polinomiales necesarias para calcular los elementos de la matriz \mathbf{M} y el vector $\mathbf{f}(t)$.

donde τ_n es el período natural de vibración más pequeño del cuerpo elástico.

En una situación hipotética de que la vibración del cuerpo elástico estuviera caracterizada por una onda senoidal de la forma:

$$f(t) = A \text{ sen } \omega t ,$$

el período de vibración estaría dado por:

$$\tau = \frac{2\pi}{\omega} .$$

De acuerdo con la fig. (4-8), el paso máximo de tiempo recomendado para tomar las variaciones máximas y mínimas de la fuerza de excitación debe ser:

$$\Delta t = \frac{2\pi}{(m-1)\omega} = \frac{\tau}{m-1} \quad (15)$$

donde m es el número deseado de puntos de discretización del tiempo y debe pertenecer a la serie $\{5, 9, 13, 17, 21, \dots, 2k-1, 2k+3, \dots\}$, para comprender los puntos donde una función senoidal tiene máximos, mínimos y ceros.

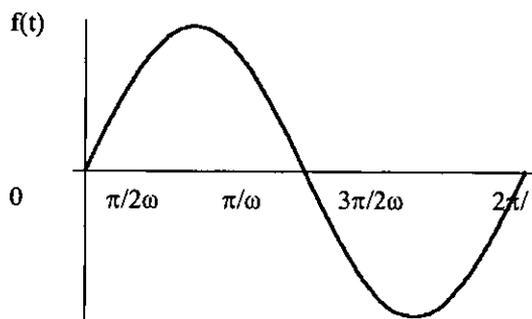


fig. 4-8: Paso de tiempo máximo recomendado.

Por lo tanto, una adecuada discretización del tiempo estará dada por la ecuación (15), la cual incluye la recomendada en la ecuación (14), pero no necesariamente la vibración del cuerpo elástico debe estar caracterizada por una onda senoidal.

$$\mathbf{K} \mathbf{u}(0) = \mathbf{f}(0),$$

cuyo resultado $\mathbf{u}(0)$ y las condiciones iniciales $\mathbf{v}(0)$, servirán para resolver el problema dinámico en cada paso del tiempo. Como se estableció en el capítulo anterior, el modelo numérico equivalente del problema dinámico está dado por las ecuaciones (21) y (22) del capítulo 3:

$$\left(\frac{2}{\Delta t} \mathbf{M} + \frac{\Delta t}{2} \mathbf{K}\right) \mathbf{u}(t + \Delta t) - \left(\frac{2}{\Delta t} \mathbf{M} - \frac{\Delta t}{2} \mathbf{K}\right) \mathbf{u}(t) - 2\mathbf{M}\mathbf{v}(t) = \frac{\Delta t}{2} \mathbf{f}(t + \Delta t) + \frac{\Delta t}{2} \mathbf{f}(t)$$

y

$$\mathbf{v}(t) = 2 \frac{\mathbf{u}(t) - \mathbf{u}(t - \Delta t)}{\Delta t} - \mathbf{v}(t - \Delta t).$$

De ahí que la solución para cada tiempo haya sido establecida por las ecuaciones (24) y (25) del capítulo 3 como:

$$\mathbf{u}(t + \Delta t) = \left(\frac{2}{\Delta t} \mathbf{M} + \frac{\Delta t}{2} \mathbf{K}\right)^{-1} \left(\left(\frac{2}{\Delta t} \mathbf{M} - \frac{\Delta t}{2} \mathbf{K}\right) \mathbf{u}(t) + 2\mathbf{M}\mathbf{v}(t) + \frac{\Delta t}{2} \mathbf{f}(t + \Delta t) + \frac{\Delta t}{2} \mathbf{f}(t) \right)$$

$t \geq 0,$

donde las componentes de las matrices columna $\mathbf{u}(t)$, $\mathbf{v}(t)$, $\mathbf{f}(t)$ y $\mathbf{f}(t+\Delta t)$ quedan determinados de acuerdo con la siguiente tabla:

t	$\mathbf{u}(t)$	$\mathbf{v}(t)$	$\mathbf{f}(t)$	$\mathbf{f}(t+\Delta t)$
$0^{(a)}$	$\mathbf{u}(0) = \mathbf{K}^{-1} \mathbf{f}(0)$	0	$\mathbf{f}(0)$	$\mathbf{f}(0)$
$\Delta t^{(b)}$	$\mathbf{u}(0)$	$0^{(c)}$	$\mathbf{f}(0)$	$\mathbf{f}(\Delta t)$
$2\Delta t$	$\mathbf{u}(\Delta t)$	$2 \frac{\mathbf{u}(\Delta t) - \mathbf{u}(0)}{\Delta t} - \mathbf{v}(0)$	$\mathbf{f}(\Delta t)$	$\mathbf{f}(2\Delta t)$
$3\Delta t$	$\mathbf{u}(2\Delta t)$	$2 \frac{\mathbf{u}(2\Delta t) - \mathbf{u}(\Delta t)}{\Delta t} - \mathbf{v}(\Delta t)$	$\mathbf{f}(2\Delta t)$	$\mathbf{f}(3\Delta t)$
\vdots	\vdots	\vdots	\vdots	\vdots
$j\Delta t$	$\mathbf{u}((j-1)\Delta t)$	$2 \frac{\mathbf{u}((j-1)\Delta t) - \mathbf{u}((j-2)\Delta t)}{\Delta t} - \mathbf{v}((j-2)\Delta t)$	$\mathbf{f}((j-1)\Delta t)$	$\mathbf{f}(j\Delta t)$

Tabla 4-2: Solución del problema dinámico: (a) Caso estático, (b) arranque, (c) condición inicial.

Algoritmo de solución.

Nótese que el modelo numérico equivalente para el caso dinámico es de la forma:

$$\tilde{\mathbf{K}}(t)\mathbf{u}(t) = \tilde{\mathbf{f}}(t)$$

donde la matriz $\tilde{\mathbf{K}}(t)$ es una matriz equivalente compuesta por las matrices \mathbf{K} y \mathbf{M} , y el vector $\tilde{\mathbf{f}}(t)$ depende de las matrices \mathbf{K} y \mathbf{M} , así como de los vectores, $\mathbf{u}(t)$, $\mathbf{v}(t)$, y $\mathbf{f}(t)$. El algoritmo para determinar la solución de nuestro problema en el caso estático y para cada paso del tiempo es el siguiente:

- A. Caso estático, $t = 0$.
 - A.1. Construir la matriz \mathbf{K} .
 - A.2. Construir el vector $\mathbf{f}(0)$.
 - A.3. Modificar la matriz \mathbf{K} y el vector $\mathbf{f}(0)$ por imposición de restricciones en los desplazamientos.
 - A.4. Resolver para $\mathbf{u}(0)$.
- B. Caso dinámico, $t = j\Delta t, j \geq 1$.
 - B.1. Construir la matriz $\tilde{\mathbf{K}}(t)$.
 - B.2. Construir el vector $\tilde{\mathbf{f}}(t)$.
 - B.3. Modificar la matriz $\tilde{\mathbf{K}}(t)$ y el vector $\tilde{\mathbf{f}}(t)$ por imposición de restricciones en los desplazamientos.
 - B.4. Resolver para $\mathbf{u}(t + \Delta t)$.
 - B.5. Volver a B.1.

4.6 MANEJO DE LOS DESPLAZAMIENTOS.

Interpolación de los desplazamientos.

Una vez que las variables primarias han sido determinadas, en este caso los desplazamientos nodales, éstas deben ser manejadas de alguna manera para determinar otro tipo de información interesante acerca del problema, que en nuestro caso son: interpolación de los desplazamientos en puntos del dominio no nodales, distribución de las deformaciones y esfuerzos en el dominio.

Definición 4.1. Campo de desplazamientos asociado a cada elemento.

Sea $\mathbf{u}(t)$ el campo vectorial de desplazamientos que aproxima a la solución de un problema en EDP al tiempo t , definido sobre un dominio Ω cuya discretización Ω' está compuesta de un conjunto finito de subdominios triangulares. Para el i -ésimo subdominio de la discretización Ω'_i definido por el conjunto de coordenadas $C = \{ (x_j, y_j), (x_k, y_k), (x_l, y_l) \}$, el conjunto de desplazamientos $\mathbf{u}_i(t) = \{ (u_j(t), v_j(t)), (u_k(t), v_k(t)), (u_l(t), v_l(t)) \}$ correspondiente a C , lo llamaremos **campo de desplazamientos asociado** al subdominio Ω_i , en el tiempo t .

Lema 4.1. Interpolación de los desplazamientos.

Sea $\mathbf{u}_i(t) = \{ (u_j(t), v_j(t)), (u_k(t), v_k(t)), (u_l(t), v_l(t)) \}$ el campo de desplazamientos asociado al subdominio Ω_i en el tiempo t definido por el conjunto de coordenadas $C = \{ (x_j, y_j), (x_k, y_k), (x_l, y_l) \}$. Dado un punto $p \in \Omega_i$ de coordenadas (x, y) , los desplazamientos $u(x, y, t)$ y $v(x, y, t)$ correspondientes a p están dados por:

$$u(x, y, t) = u_j(t) \varphi_j(x, y) + u_k(t) \varphi_k(x, y) + u_l(t) \varphi_l(x, y)$$

y

$$v(x, y, t) = v_j(t) \varphi_j(x, y) + v_k(t) \varphi_k(x, y) + v_l(t) \varphi_l(x, y)$$

donde $\varphi_j, \varphi_k, \varphi_l : \Omega_i \rightarrow \mathfrak{R}$ son funciones globales de interpolación con soporte en Ω_i .

Deformaciones, esfuerzos y esfuerzos principales.

La tarea de la mecánica, definida como la relación entre cuerpos, sistemas de fuerzas y movimientos, es la predicción del comportamiento del cuerpo debido a esa relación, para evaluar las posibilidades de falla. Para lograr este objetivo se han desarrollado los conceptos de esfuerzo y deformación y varios criterios de falla. Los análisis de esfuerzos y deformaciones son independientes pero ambos conceptos están relacionados por las propiedades mecánicas del material.

La deformación es una cantidad puramente geométrica definida como el movimiento relativo entre puntos del cuerpo [6, pp. 28-29]. La deformación en la vecindad de un punto del cuerpo queda determinada por un criterio de deformación, el cual indica si hubo o no movimiento relativo entre otros puntos del cuerpo y el punto en cuestión.

Por otro lado, el esfuerzo en un punto es función de la posición del punto con respecto a un sistema de coordenadas y la orientación de un plano que pasa por dicho punto [6, p. 4]. Entonces existe una infinidad de esfuerzos en un punto, debido a la infinidad de planos que pasan por éste. El conjunto de esfuerzos en un punto se conoce como estado de esfuerzo en dicho punto, y para cada estado de esfuerzo existen esfuerzos que no dependen del sistema de coordenadas, llamados esfuerzos principales [6, pp. 14-16], [14, pp. 17-36], [24, pp. 17-27], sino del campo tensorial de elasticidad del cuerpo, del sistema de fuerzas definidas sobre el cuerpo, la representación analítica del dominio y las condiciones de frontera.

Lema 4.2. Deformaciones.

Sea $\mathbf{u}_i(t) = \{ (u_j(t), v_j(t)), (u_k(t), v_k(t)), (u_l(t), v_l(t)) \}$ el campo de desplazamientos asociado al subdominio Ω_i en el tiempo t definido por el conjunto de coordenadas $C = \{ (x_j, y_j), (x_k, y_k), (x_l, y_l) \}$. El campo de deformación asociado a Ω_i en el tiempo t está dado por la expresión:

$$E_k(t) = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} \\ \varepsilon_{21} & \varepsilon_{22} \end{bmatrix}$$

donde:

$$\varepsilon_{11} = \frac{\partial u}{\partial x} = \frac{2u_k(t) - u_j(t) - u_i(t)}{2(x_k - x_j)},$$

$$\varepsilon_{12} = \varepsilon_{21} = \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) = \frac{2u_k(t) - u_j(t) - u_i(t)}{4(y_1 - y_j)} + \frac{2v_1(t) - v_j(t) - v_k(t)}{4(x_k - x_j)},$$

$$\varepsilon_{22} = \frac{\partial v}{\partial y} = \frac{2v_1(t) - v_j(t) - v_k(t)}{2(y_1 - y_j)}.$$

Lema 4.3. Esfuerzos.

Dado un cuerpo elástico Ω con módulos de Lamé μ y λ , sea $E_i(t)$ el campo de deformación asociado a Ω_i en el tiempo t . El campo de esfuerzo asociado a Ω_i en el tiempo t está dado por la expresión:

$$S_k(t) = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}$$

donde:

$$\sigma_{11} = (2\mu + \lambda) \varepsilon_{11} + \lambda \varepsilon_{22},$$

$$\sigma_{12} = \sigma_{21} = 2\mu \varepsilon_{12},$$

$$\sigma_{22} = (2\mu + \lambda) \varepsilon_{22} + \lambda \varepsilon_{11}.$$

Observación.

Las deformaciones y esfuerzos, más que asociarlos a un punto del cuerpo como su definición lo establece, se han asociado a un subdominio de la discretización de éste. La solución numérica obtenida del análisis nos permite construir expresión polinomial de las funciones de desplazamientos u y v ; los valores de deformación y esfuerzo en un punto específico están determinados por la función derivada de las funciones de desplazamiento en ese punto. Sin embargo, en los lemas 4.2 y 4.3 sobre deformaciones y esfuerzos se hace una aproximación de sus derivadas por un esquema en diferencias, considerando de esta manera la vecindad del punto la definida por el subdominio analizado. El punto debe ser representativo del subdominio (por

decir, su centro geométrico), de manera que para una mayor aproximación de estos valores a la solución real se requiera un refinamiento de la discretización del dominio.

Lema 4.4. Esfuerzos principales.

Dado un cuerpo elástico Ω con módulos de Lamé μ y λ , sea $S_i(t)$ el campo de esfuerzo asociado a Ω_i en el tiempo t . Los esfuerzos principales asociados a Ω_i en el tiempo t están dados por la expresión:

$$\sigma_1, \sigma_2 = \frac{\sigma_{11} + \sigma_{22}}{2} \pm \left(\left(\frac{\sigma_{11} - \sigma_{22}}{2} \right)^2 + \sigma_{21}^2 \right)^{1/2}$$

Criterios de falla.

Una forma de determinar la falla del material es la evaluación del estado de esfuerzo en cada punto del cuerpo, existen varios criterios de falla que dependen del tipo de material y del tipo de estado de esfuerzo presente. Para materiales dúctiles hay dos criterios de falla: cortante máximo o de Tresca y máxima energía de distorsión o de von Mises, ambos calculados a partir de los esfuerzos principales. La zona segura de ambos criterios está descrita por un hexágono y una elipse, respectivamente. Estas figuras geométricas coinciden en los vértices del hexágono pero en otros puntos el criterio de Tresca es más conservador. De esta manera, mientras el estado de esfuerzo no exceda el esfuerzo de Tresca o de von Mises, se puede decir que el cuerpo es seguro [2, pp. 365-367], [16, pp. 115].

Lema 4.5. Esfuerzos de von Mises.

Dado un cuerpo elástico Ω con módulos de Lamé μ y λ , sean σ_1, σ_2 los esfuerzos principales asociados a Ω_i en el tiempo t . Los esfuerzos de von Mises asociados a Ω_i en el tiempo t están dados por la expresión:

$$\sigma_{vM} = \left(\sigma_1^2 - \sigma_1 \sigma_2 + \sigma_2^2 \right)^{1/2} .$$

Capítulo 5:

Algoritmos de programación.

5.1 INTRODUCCIÓN.

Los pasos involucrados en el método de elemento finito son sistemáticos, como se pudo constatar en los planteamientos hechos en el capítulo 3, por lo que pueden implementarse mediante un lenguaje de programación de alto nivel para ser ejecutados con la ayuda de una computadora. De hecho, el método de elemento finito sólo es efectivo y confiable en sus resultados con la ayuda de computadoras por la gran cantidad de cálculos implicados.

El método de elemento finito ha demostrado ser una técnica muy eficiente en la solución numérica de ecuaciones diferenciales. Dado un modelo físico en EDP, la implementación del método para su solución se resume en tres pasos principales: construcción del modelo de elemento finito, construcción y solución del sistema de ecuaciones equivalente, y finalmente, interpretación de resultados.

Un programa general de análisis de elemento finito se constituye de tres módulos básicos: pre-procesador, procesador y post-procesador [18, pp. 258-257], cuyas funciones son discutidas en las secciones siguientes.

5.2 PRE-PROCESADOR.

En este módulo se lleva a cabo la adquisición de datos. Aquí son leídos y/o generados los datos con los cuales se construye el modelo de elemento finito. Estos datos tienen que ver con las siguientes características del problema:

1. Geometría: dimensiones del dominio.
2. Propiedades físicas del material: coeficientes de las ecuaciones diferenciales.
3. Propiedades de la malla: densidad de la malla, número total de nodos y elementos, generación de coordenadas globales.
4. Condiciones de frontera: número de nodos con restricción de movimiento, tipo de restricción de movimiento (dirección); número de nodos y/o elementos cargados, tipo de carga (estática, dinámica, concentrada, distribuida).

5. Discretización de la carga.
6. Verificación de datos: mostrar la información general proporcionada, mostrar las coordenadas globales, dibujar el modelo de elemento finito.
7. Información adicional: tipo de problema (fenómeno físico), tipo de análisis (estático, dinámico).

5.3 PROCESADOR.

En este módulo son ejecutados todos los cálculos necesarios para resolver los modelos numéricos equivalentes, derivados de la aplicación del método de elemento finito a un problema en EDP. Para un problema dinámico, como es nuestro caso, es necesario resolver en primer lugar el problema estático y posteriormente el problema dinámico, los cuales están caracterizados por las ecuaciones:

$$\mathbf{K} \mathbf{u}(0) = \mathbf{f}(0) \quad \text{y} \quad \tilde{\mathbf{K}} \mathbf{u}(t) = \tilde{\mathbf{f}}(t) ,$$

respectivamente; pero antes de esto son necesarios algunos cálculos iniciales. La estructura del módulo procesador es la que se describe enseguida:

A. Cálculos iniciales.

8. Cálculo de las matrices \mathbf{K} y \mathbf{M} : asociación de nodos globales, coordenadas globales y elementos finitos, cálculo de gradientes de funciones de interpolación, integración numérica, verificación de las propiedades de las matrices (simetría, bandeado, dominancia diagonal).
9. Cálculo de la matriz equivalente $\tilde{\mathbf{K}}$.
10. Cálculo del vector $\mathbf{f}(0)$: integración numérica, carga estática.
11. Cálculo del vector $\mathbf{v}(0)$ por condiciones iniciales.
12. Modificación de las matrices \mathbf{K} y $\tilde{\mathbf{K}}$, y los vectores $\mathbf{f}(0)$ y $\mathbf{v}(0)$ por aplicación de restricciones en los desplazamientos.

B. Solución del caso estático.

13. Resolver para $\mathbf{u}(0)$ con el método de Gauss-Seidel.

C. Solución del caso dinámico (proceso iterativo).

14. Cálculo del vector $\mathbf{v}(t)$: esquema de recursión.
15. Cálculo del vector $\tilde{\mathbf{f}}(t)$: esquema de recursión.
16. Modificación del vector $\tilde{\mathbf{f}}(t)$ por aplicación de restricciones en los desplazamientos.
17. Resolver para $\mathbf{u}(t + \Delta t)$ con el método de Gauss-Seidel.
18. Volver a 14.

5.4 POST-PROCESADOR.

La tarea principal de este módulo es mostrar los resultados del análisis en formato numérico y gráfico. Para lograr esto, el módulo debe manejar las variables principales (resultados obtenidos en el módulo anterior) para determinar cualquier otra variable secundaria que se desee. La salida gráfica de los resultados nos da una idea cualitativa del análisis, nos permite localizar las áreas donde se presentan los valores máximos o mínimos de nuestras variables. Por otro lado, los resultados numéricos nos permiten cuantificar tales variables.

La estructura del módulo post-procesador estaría constituida por los elementos:

19. Interpolación: Cálculo de la variable primaria en puntos diferentes a los nodos.
20. Variables secundarias: Cálculo de las deformaciones y esfuerzos.
21. Salida numérica y gráfica de los resultados obtenidos por medio de archivos de texto y de imágenes.

5.5 DIAGRAMA DE BLOQUES.

Un buen principio para la creación de aplicaciones computacionales grandes y complejas, es la creación de éstos por medio de módulos o bloques de funcionamiento independiente [18, pp. 258-257], [13, pp. 474-498, 505-510] y [29, pp. 453-474], esto facilita labores de detección y corrección de errores, modificaciones y ampliaciones posteriores.

Bajo este principio, se han diseñado los diagramas de bloques y algoritmos de programación para implementar los modelos numéricos del problema particular del capítulo 4 y encontrar una solución numérica aproximada. Una característica importante de estos módulos es el manejo de archivos de texto como dispositivos de salida, esto permite manejar la información generada en uno de los módulos por el resto de ellos, además de hacer un uso eficiente de la memoria de la máquina, recurso muy limitado en estos programas.

Cabe mencionar que, si bien el módulo pre-procesador está limitado a discretizar un dominio rectangular, su salida es un conjunto de archivos donde se almacena toda la información acerca de la construcción de la malla; esto permite al módulo procesador trabajar sobre ellos para realizar los cálculos necesarios, sin la menor preocupación acerca de la naturaleza del dominio. Esto quiere decir que si discretizamos un dominio irregular y alimentamos al procesador la información en el formato correspondiente, podremos resolver el problema.

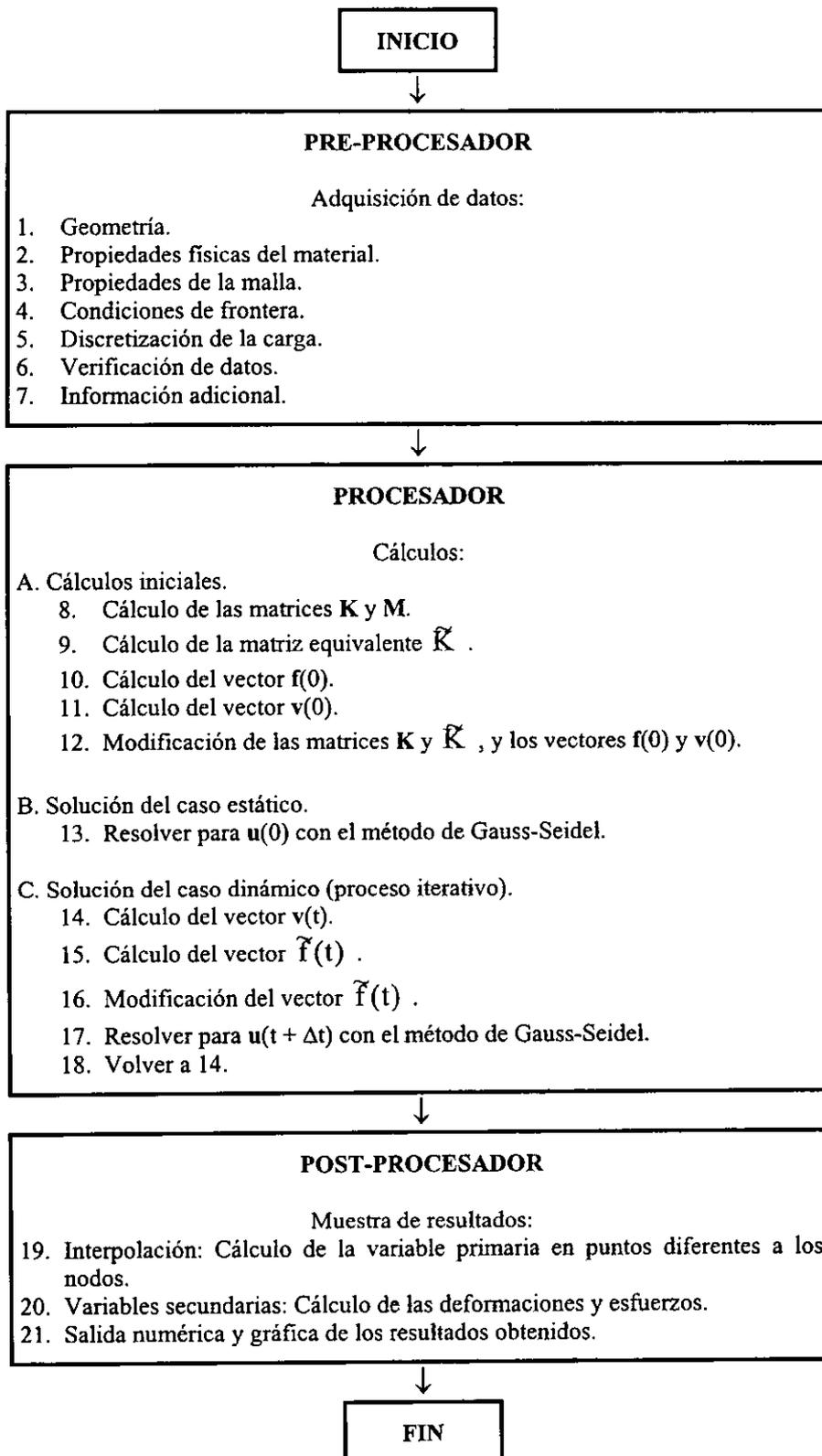
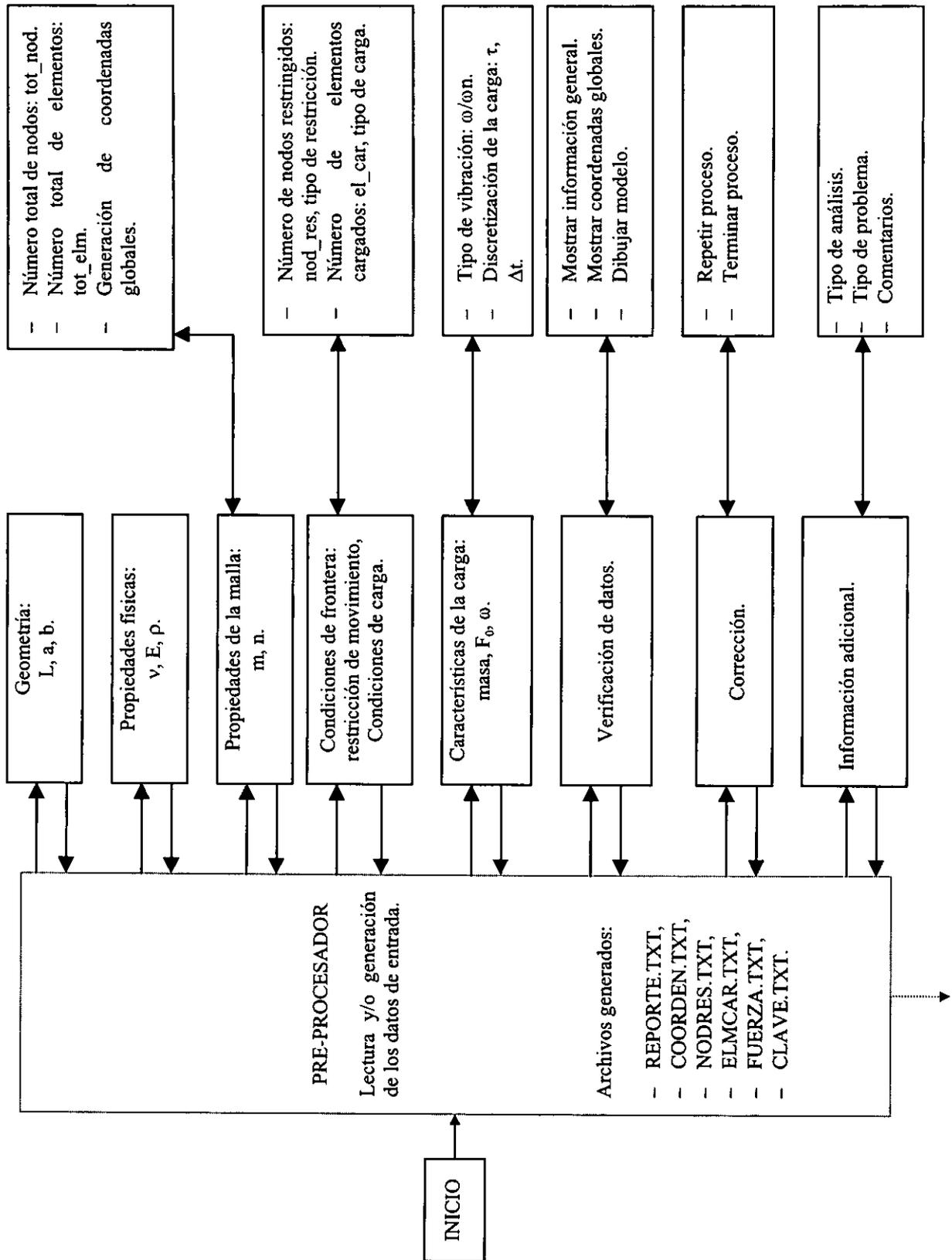
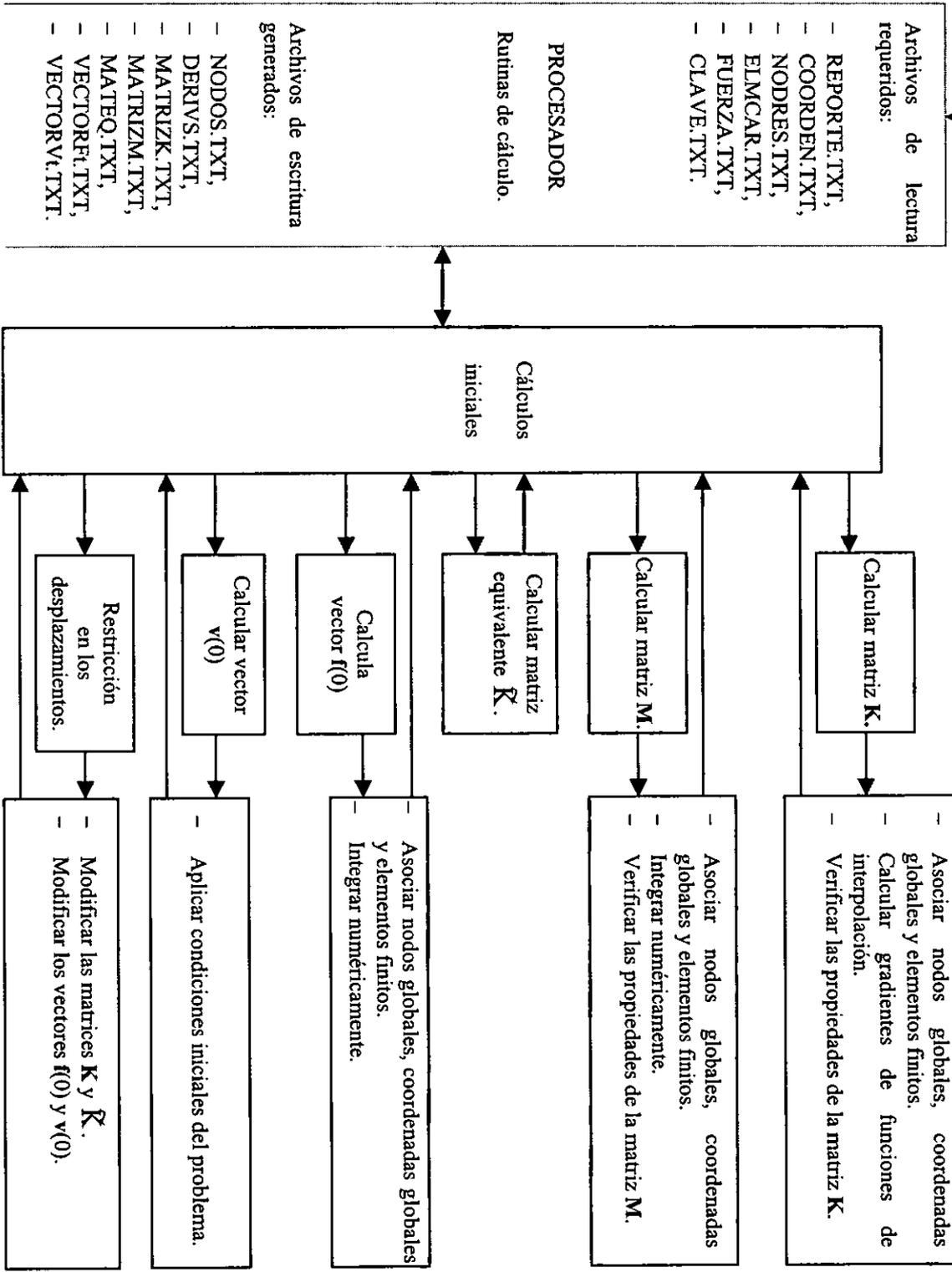
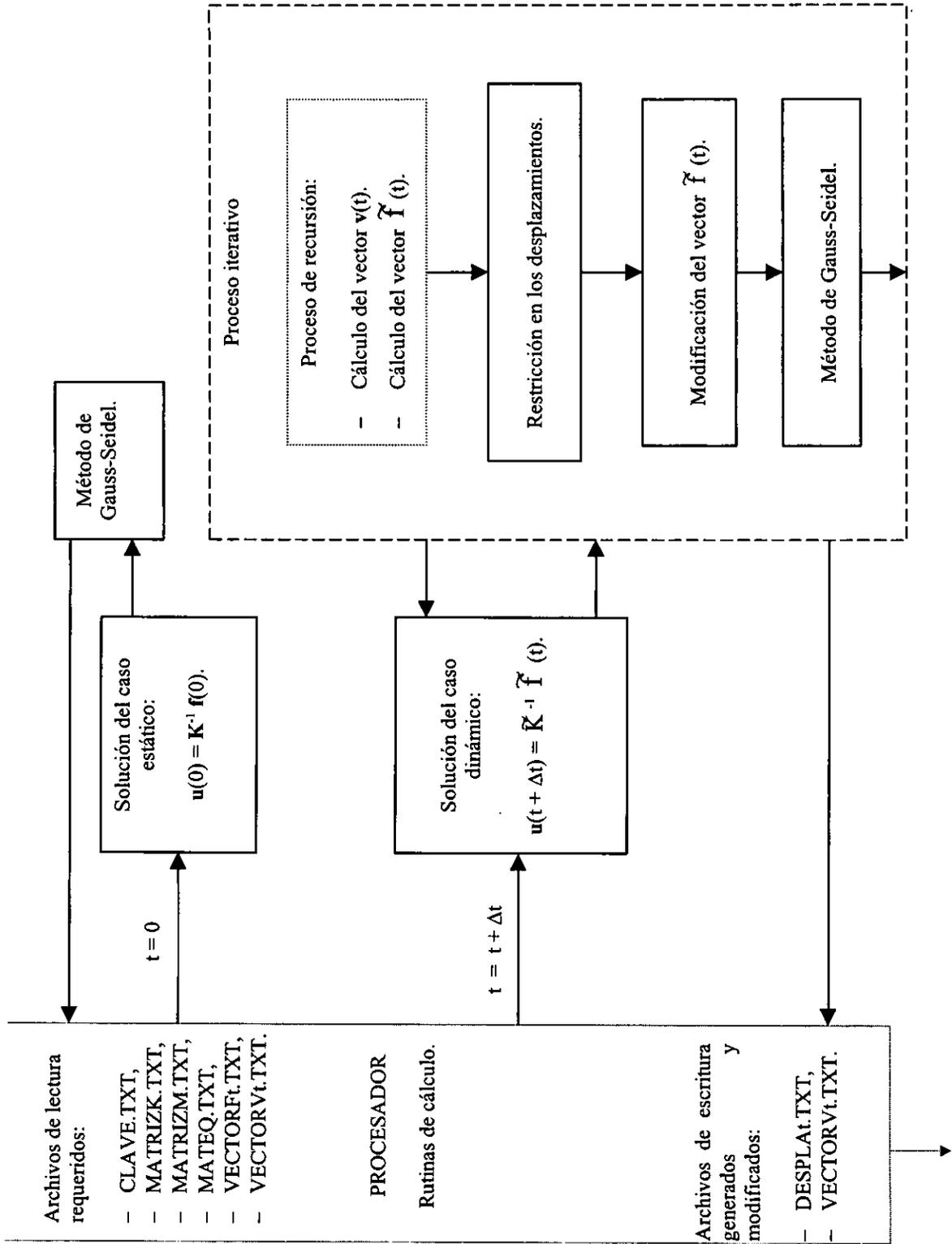
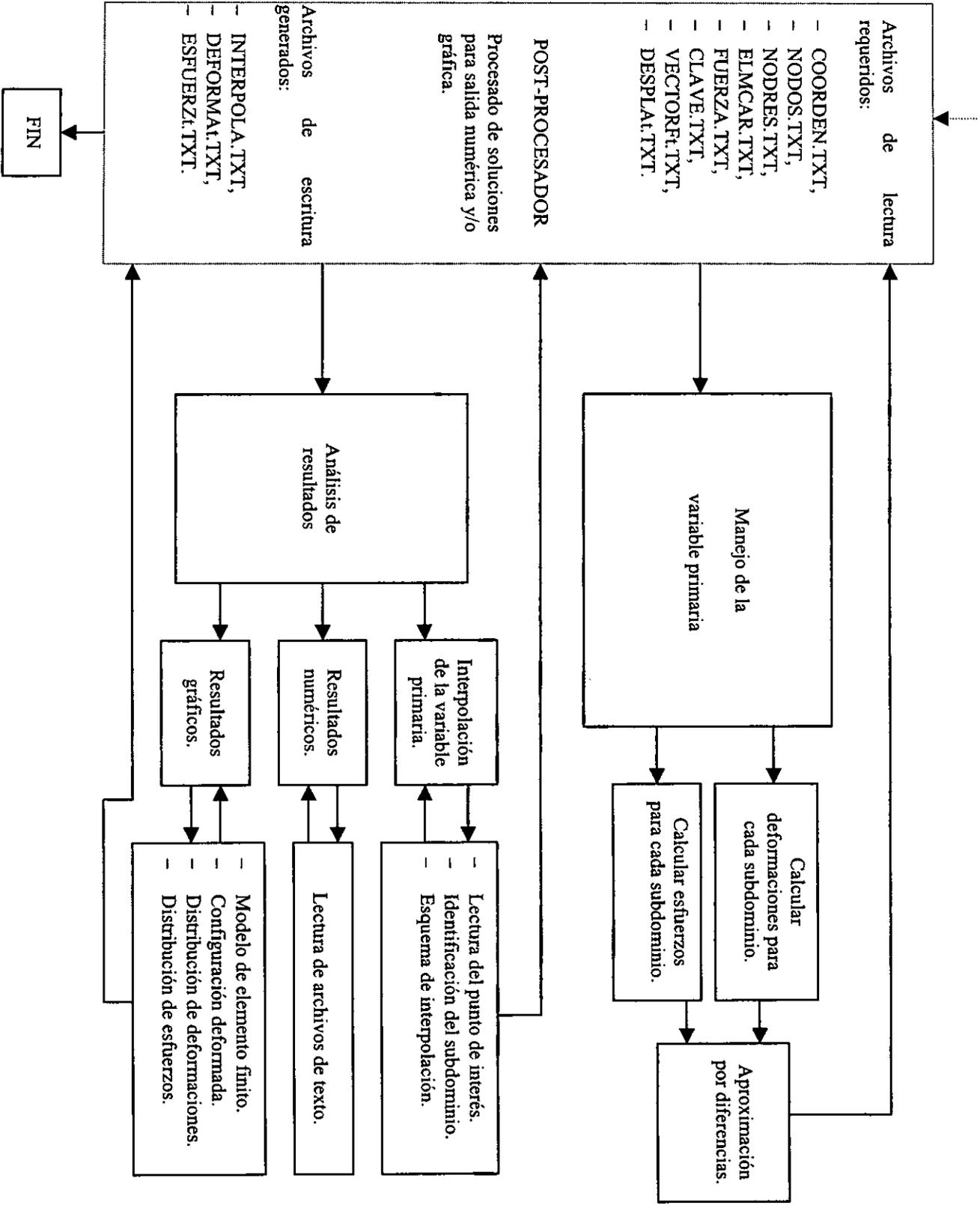


fig. 5-1: Pre-procesador, procesador y post-procesador.









5.6 ALGORITMOS DE PROGRAMACIÓN.

En esta sección se presentan los algoritmos de programación para implementar un análisis por el método de elemento finito del problema definido en el capítulo anterior. En el apéndice A-8: *Listado de programas*, se puede encontrar el código fuente correspondiente a estos algoritmos.

Pre-procesador.

a) Archivos de trabajo:

```
FILE *reporte, *coord, *nodres, *elmcar,*fuerza, *clave;
```

b) Variables locales:

```
char
    ver                // Verificación de datos.
    cor                // Corrección de datos.
int
    m, n, tot_el, tot_nod // Propiedades de la malla.
    nod_res, el_carg     // Condiciones de frontera.
    Puntos              // Discretización del tiempo.
float
    l, a, b            // Dimensiones del dominio.
    nu, E, densidad    // Propiedades físicas del cuerpo.
    masa, base, F0, w  // Características de la carga.
    wn, tau, paso      // Parámetros de vibración.
```

c) Apertura de archivos de trabajo:

```
reporte = fopen(ruta_reporte,"w"); // Reporte
coord = fopen(ruta_coord,"w"); // Coordenadas globales
nodres = fopen(ruta_nodres,"w"); // Nodos restringidos
elmcar = fopen(ruta_elmcar,"w"); // Elementos cargados
fuerza = fopen(ruta_fuerza,"w"); // Fuerza discretizada
clave = fopen(ruta_clave,"w"); // Datos clave del análisis
```

d) Programa principal:

```
main(){
    Presentación y descripción del problema();
    do {
        Lee geometría del dominio (&l, &a, &b);
        Lee propiedades del cuerpo (&nu, &E, &densidad);
        Calcula coordenadas_globales(&l, &a, &m, &n, &tot_el, &tot_nod);
        Cierra archivo COORD.TXT;
        Lee condiciones de frontera(&tot_el, &tot_nod, &nod_res, &el_carg);
        Cierra archivo NODRES.TXT;
        Cierra archivo ELMCAR.TXT;
        Lee características de la carga(&masa, &base, &F0, &w);
        Identifica tipo de vibracion(&l, &a, &b, &base, &E, &densidad, &masa, &w, &wn);
        Discretiza fuerza(&F0, &w, &wn, &tau, &paso);
```

```

Cierra archivo FUERZA.TXT;
Cierra archivo REPORTE.TXT;
Guarda datos claves(&l, &a, &b, &nu, &E, &densidad, &m, &n, &tot_el, &tot_nod, &nod_res, &el_carg,
&puntos, &tau, &paso, &masa);
Cierra archivo CLAVE.TXT;
do {
    ver = verificacion();
    switch (ver) {
        case '1' : Muestra información general();
                    break;
        case '2' : Muestra coordenadas globales(&tot_el);
                    break;
        case '3' : break;
    } // switch (ver) { ...
} while ( ver != '4');
cor = correccion();
switch (cor) {
    case '1' : Repite proceso();
                break;
    case '2' : Termina proceso;
                break;
    } // switch (cor) { ...
} while ( cor == '1');
} // Termina programa principal

```

Las funciones definidas en el módulo pre-procesador cumplen con los requerimientos establecidos en el diagrama del programa general. Toda la información adquirida en este módulo es almacenada en archivos de texto para su uso posterior, ya sea por este mismo programa en la verificación o por los siguientes módulos. La función más importante en este módulo es la encargada de calcular las coordenadas globales nodales de la discretización, puesto que son parámetros empleados en la construcción de las funciones de interpolación. En el apéndice A-7: *Ejemplos de discretización*, se muestran algunos ejemplos del cálculo de coordenadas globales, correspondientes a algunas discretizaciones de dominios rectangulares.

Procesador.

a) Archivos de trabajo:

```

FILE *clave, *nodres, *elmar, *fuerza, *nodos, *coord, *deriv, *matrizK, *matK, *matKr, *matKt, *matrizM,
    *matM, *mateq, *mateqr, *mateqt, *vectorvt, *vectorFt, *vectorf0, *desplat, *vectorFt_eq, *vectorFt_eqr,
    *reporte;

```

b) Variables locales:

```

int m, n, tot_el, tot_nod, nod_res, el_carg, puntos, t;
float l, a, b, nu, E, densidad, tau, paso, masa;

```

c) Programa principal:

```

main() {

```

```

Lee datos claves(&l, &a, &b, &nu, &E, &densidad, &m, &n, &tot_el, &tot_nod,
                &nod_res, &el_carg, &puntos, &tau, &paso, &masa);

// Cálculos iniciales
asocia_nodos_con_elementos_finitos (&n, &tot_el);
calcula_derivadas_de_funciones_de_interpolacion(&tot_el, &b);
calcula_matriz_K(&nu, &E, &tot_el, &tot_nod);
calcula_matriz_M(&densidad, &tot_el, &tot_nod);
calcula_matriz_equiv(&paso, &tot_nod);
calcula_vector_f0(&puntos, &el_carg, &tot_nod, &tot_el, &masa,
                 &densidad, &b, &paso);
calcula_vector_v0(&tot_nod);

// Reducción de matrices por restricciones nodales
reduce_mat_K(&tot_nod, &nod_res);
reduce_mat_eq(&tot_nod, &nod_res);
reduce_vec_f0(&tot_nod, &nod_res);
reduce_vec_v0(&tot_nod, &nod_res);

// Solución del caso estático
Gauss_Seidel_est(&tot_nod, &nod_res);

// Solución del caso dinámico
for (t = 0; t < puntos; t++) { // Solución para cada paso del tiempo
    calcula_vector_vt();
    calcula_vector_fteq();
    reduce_vec_fteq(&tot_nod, &nod_res);
    Gauss_Seidel_din(&tot_nod, &nod_res);
}
} // Termina programa principal

```

Las funciones más interesantes de este módulo son las encargadas de calcular las matrices **K** y **M** y el vector **f(t)**. Para estos cálculos se requiere principalmente de las coordenadas globales nodales de la discretización, calculadas en el módulo anterior. Otras de las funciones interesantes son aquéllas concernientes a la reducción de matrices por restricciones nodales. El resto de las funciones, como solución de sistemas de ecuaciones, multiplicación de matrices, etc. es tomado de literatura relacionada con el tema [20].

Post-procesador.

a) Archivos de trabajo:

FILE *clave, *nodres, *elmar, *fuerza, *nodos, *coord, *desplat, *deforma, *esfuerzo, *reporte;

d) Variables locales:

```

char op; // Opción de cálculo
int m, n, tot_el, tot_nod, nod_res, el_carg, puntos, k;
float l, a, b, nu, E, densidad, tau, paso, masa;

```

e) Programa principal:

```

main() {
  Lee datos claves(&l, &a, &b, &nu, &E, &densidad, &m, &n, &tot_el, &tot_nod,
    &nod_res, &el_carg, &puntos, &tau, &paso, &masa);

  // Cálculos
  for (k = 1; k <= tot_el; k++) {
    // Cálculo para cada subdominio
    Lee coordenadas globales(&tot_el, &tot_nod, &nod_res, &el_carg);
    Lee desplazamientos nodales(&tot_nod, &nod_res);
    Calcula deformaciones();
    Calcula esfuerzos();
  }
  do {
    op = opcion();
    switch (op) {
      case '1' : interpolación();
                break;
      case '2' : dibuja malla de elemento finito(&tot_el, &tot_nod);
                break;
      case '3' : dibuja configuración deformada(&tot_el, &tot_nod);
                break;
      case '4' : dibuja distribución de deformaciones();
                break;
      case '5' : dibuja distribución de esfuerzos ();
                break;
      case '6' : break;
    } // switch (op) { ...
  } while ( op != '6');
} // Termina programa principal

```

5.7 EJEMPLO DE APLICACIÓN: DEFORMACIÓN DE UNA VIGA.

Para validar los planteamientos teóricos y modelos numéricos desarrollados en los capítulos 3 y 4, en esta sección se verificará la solución para el caso estático. Para ello se determinará el campo de desplazamientos de la viga en cantiliver con la solución analítica dada por la mecánica de materiales, conocida como ecuación diferencial de la curva elástica, cuya derivación se encuentra en el apéndice A-5: *Soluciones analíticas*. Esta solución analítica se comparará con la obtenida por los programas diseñados en secciones anteriores para su verificación.

Solución numérica, método de elemento finito.

Al final del apéndice A-7: *Ejemplos de discretización* se agrega el listado del archivo REPORTE.TXT, el cual es creado por el programa pre-procesador para almacenar la información general del análisis adquirida a través de su ejecución; al ser corrido el programa procesador se almacena información en el mismo archivo acerca de los cálculos iniciales y solución del caso estático.

Solución numérica, método de elemento finito.

Al final del apéndice A-7: *Ejemplos de discretización* se agrega el listado del archivo REPORTE.TXT, el cual es creado por el programa pre-procesador para almacenar la información general del análisis adquirida a través de su ejecución; al ser corrido el programa procesador se almacena información en el mismo archivo acerca de los cálculos iniciales y solución del caso estático.

La fig. (5-1) muestra tres gráficas del modelo de elemento finito para el problema analizado. De izquierda a derecha, en las dos primeras se ilustra la numeración global de los nodos y elementos de la discretización, respectivamente. El orden establecido para la numeración de los elementos permite que el bandedo de las matrices de elemento finito sea más pronunciado, este orden de numeración puede apreciarse mejor en las discretizaciones del apéndice A-7: *Ejemplos de discretización*. En la tercera gráfica se muestran las cargas a que está sujeta la viga, existe una carga uniformemente distribuida que corresponde a las fuerzas de cuerpo debidas al propio peso

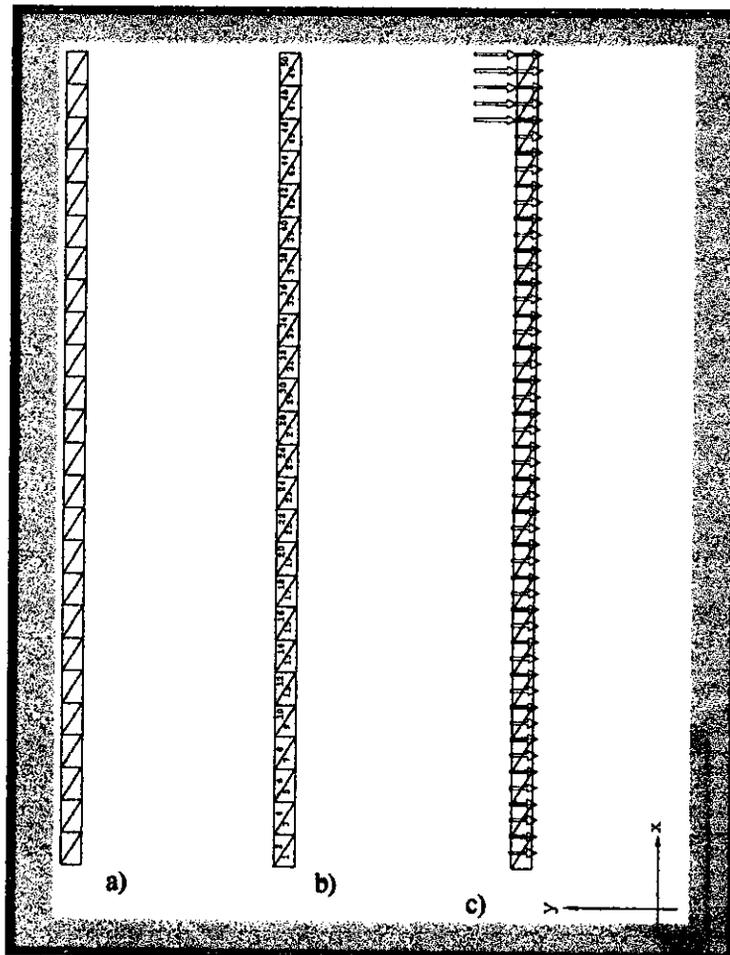
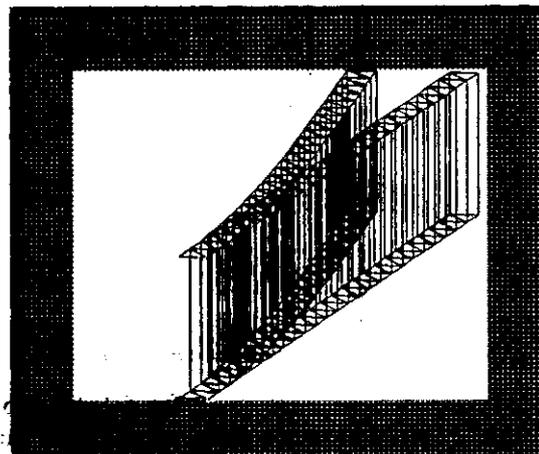
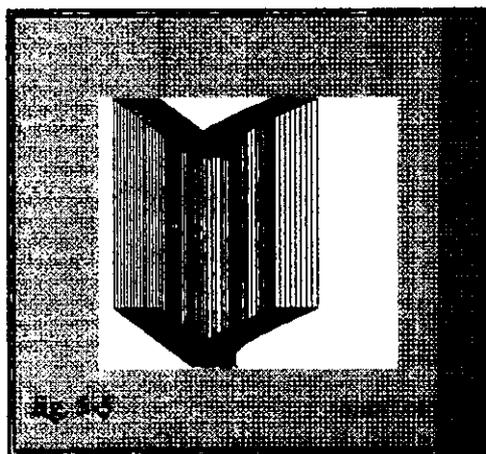
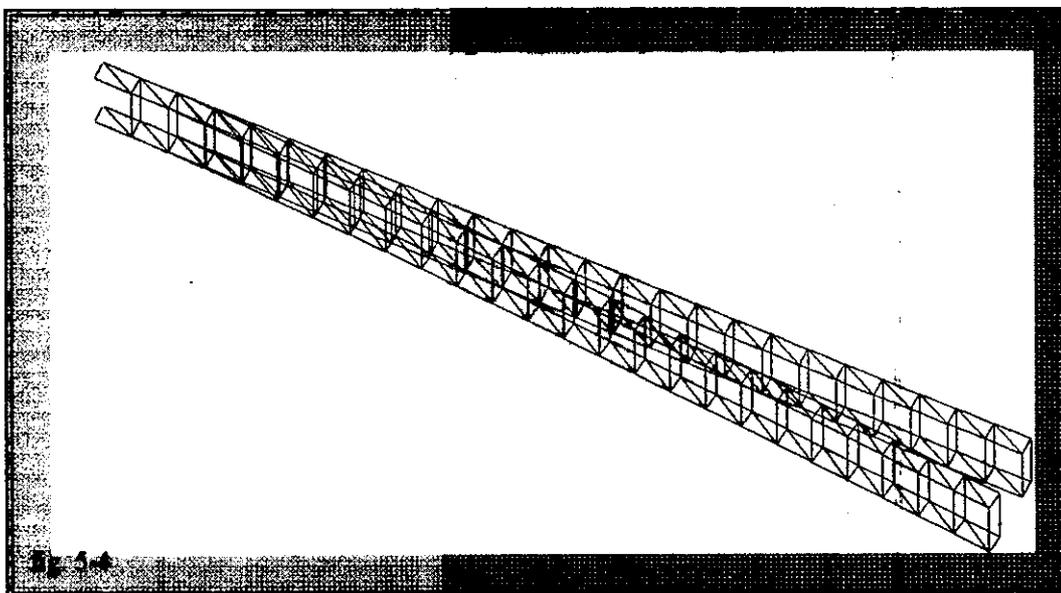
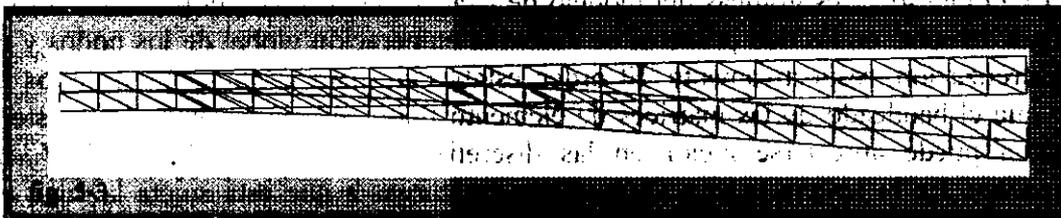
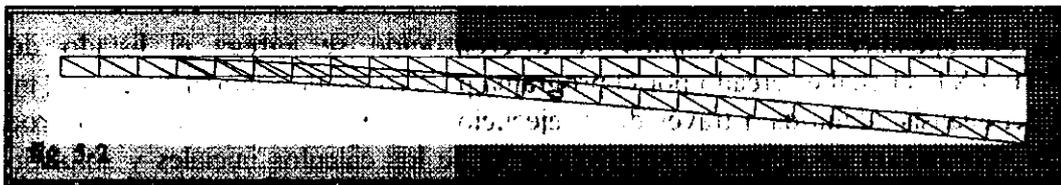
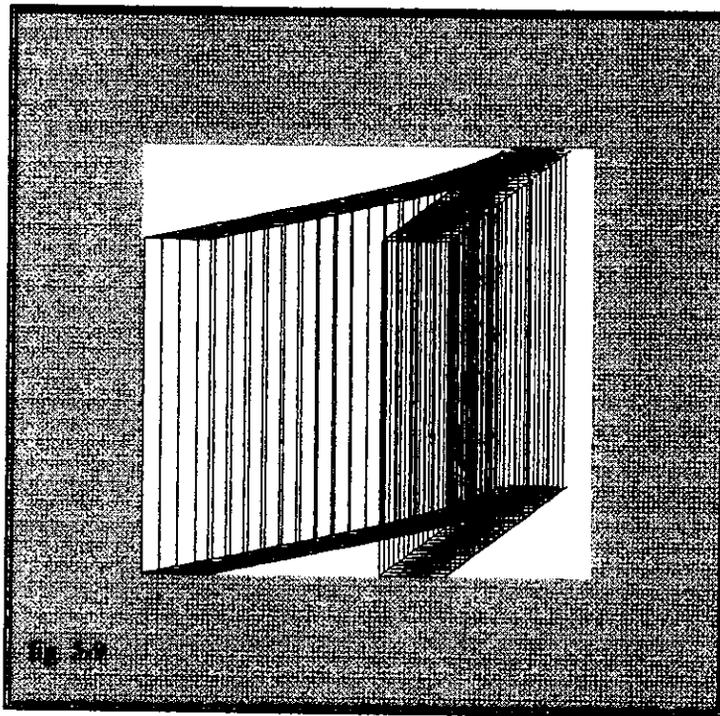
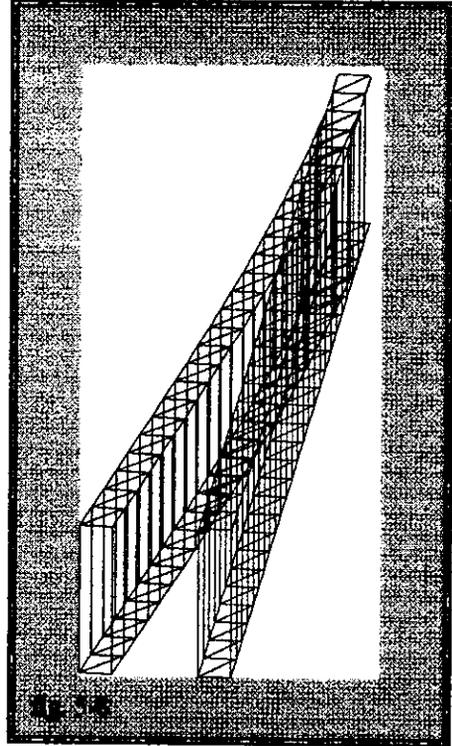
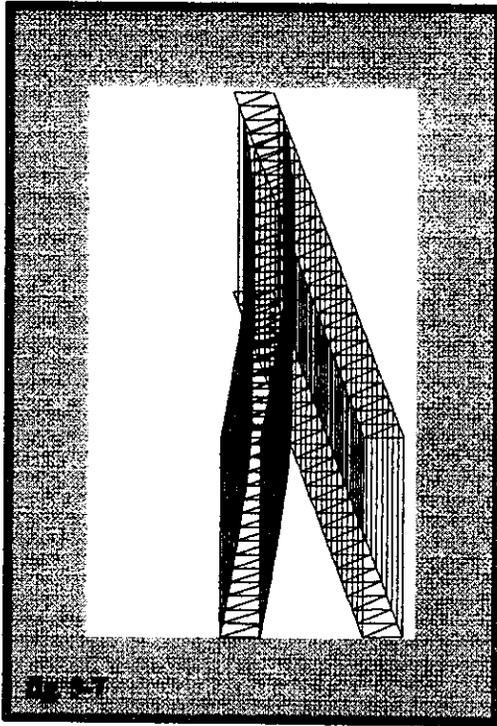


fig. 5-1: Modelo de elemento finito:

a) Numeración global de nodos, b) Numeración global de elementos, c) Cargas.





figs. 5-2 a 5-9: Diferentes vistas de la configuración deformada.

de la viga. En el extremo libre se agrega el peso del motor como fuerza de superficie, y en el análisis dinámico se contemplaría la variación de la fuerza de excitación. En las tres gráficas se indican también las condiciones de apoyo.

Comparación de soluciones y verificación de resultados.

La configuración deformada ilustrada en las figs. (5-2) a (5-9) corresponde a lo que por sentido común se esperaba: la deflexión de la viga va aumentando a lo largo de su longitud a medida que nos acercamos a su extremo libre. En la tabla 5.1 se listan los desplazamientos obtenidos con la solución analítica para dos condiciones de carga: considerando sólo el peso de la viga e incluyendo el peso del motor. Hay que mencionar que estos desplazamientos se asocian a puntos ubicados sobre el plano neutro de la viga.

L (m)	Sólo peso de la viga (mm)	Incluye peso del motor (mm)
x	y_1	y_2
0.00	0.00	0.00
0.20	-0.11	-0.12
0.40	-0.44	-0.48
0.60	-0.96	-1.06
0.80	-1.66	-1.84
1.00	-2.53	-2.80
1.20	-3.54	-3.93
1.40	-4.68	-5.21
1.60	-5.95	-6.62
1.80	-7.32	-8.16
2.00	-8.79	-9.81
2.20	-10.34	-11.56
2.40	-11.96	-13.39
2.60	-13.65	-15.30
2.80	-15.39	-17.27
3.00	-17.18	-19.29
3.20	-19.00	-21.37
3.40	-20.85	-23.48
3.60	-22.72	-25.62
3.80	-24.62	-27.79
4.00	-26.52	-29.98
4.20	-28.44	-32.18
4.40	-30.36	-34.39
4.60	-32.29	-36.61
4.80	-34.22	-38.83
5.00	-36.14	-41.05

Tabla 5.1: Solución analítica de la vibración transversal de vigas por la ecuación diferencial de la curva elástica.

Para los mismos casos de carga, en la tabla 5.2 se listan los resultados obtenidos numéricamente, con la diferencia de que el análisis numérico también proporciona desplazamientos en dirección longitudinal de la viga. Se agrega una columna para cuantificar la diferencia entre ambas soluciones, considerando que el desplazamiento entre dos nodos de un mismo plano vertical es

idéntico (salvo algunos casos) entre sí y por consecuencia, al de un nodo hipotético ubicado sobre el plano neutro de la viga, cuyos resultados se toman de la tabla 5.1.

$\Delta x = 0.2$ (m)	Sólo peso de la viga (mm)			Incluye carga del motor (mm)		
	método de elemento finito ^(a)		error ^(b)	método de elemento finito		Error
nodo	u_x	v_x	%	u_x	v_x	%
1	0.00	0.00		0.00	0.00	
2	0.00	0.00	-	0.00	0.00	-
3	-0.05	-0.12		-0.06	-0.12	
4	0.05	-0.12	-9.09	0.06	-0.12	-
5	-0.1	-0.44		-0.11	-0.46	
6	0.1	-0.44	0.00	0.11	-0.46	4.17
7	-0.15	-0.95		-0.15	-1.00	
8	0.15	-0.95	1.04	0.16	-1.00	5.66
9	-0.19	-1.63		-0.20	-1.73	
10	0.19	-1.63	1.81	0.20	-1.73	5.98
11	-0.22	-2.46		-0.24	-2.62	
12	0.23	-2.47	2.77	0.25	-2.62	6.43
13	-0.26	-3.45		-0.27	-3.67	
14	0.27	-3.45	2.54	0.28	-3.67	6.62
15	-0.29	-4.56		-0.31	-4.86	
16	0.3	-4.56	2.56	0.32	-4.86	6.72
17	-0.31	-5.78		-0.34	-6.17	
18	0.33	-5.78	2.86	0.35	-6.17	6.80
19	-0.34	-7.12		-0.36	-7.60	
20	0.35	-7.12	2.73	0.38	-7.60	6.86
21	-0.36	-8.54		-0.39	-9.13	
22	0.37	-8.54	2.84	0.40	-9.13	6.93
23	-0.38	-10.04		-0.41	-10.75	
24	0.39	-10.04	2.90	0.42	-10.75	7.01
25	-0.39	-11.62		-0.42	-12.45	
26	0.41	-11.62	2.84	0.44	-12.46	7.02
27	-0.41	-13.26		-0.44	-14.23	
28	0.43	-13.26	2.86	0.46	-14.23	6.99
29	-0.42	-14.95		-0.45	-16.06	
30	0.44	-14.95	2.86	0.47	-16.06	7.01
31	-0.43	-16.69		-0.46	-17.94	
32	0.45	-16.69	2.85	0.49	-17.94	7.00
33	-0.44	-18.46		-0.47	-19.86	
34	0.46	-18.46	2.84	0.50	-19.86	7.07
35	-0.44	-20.26		-0.48	-21.83	
36	0.47	-20.27	2.83	0.51	-21.83	7.03
37	-0.45	-22.09		-0.49	-23.82	
38	0.47	-22.09	2.77	0.51	-23.82	7.03
39	-0.45	-23.94		-0.49	-25.83	
40	0.48	-23.94	2.76	0.52	-25.83	7.05
41	-0.45	-25.80		-0.50	-27.86	
42	0.48	-25.80	2.71	0.52	-27.86	7.07
43	-0.46	-27.67		-0.50	-29.90	
44	0.48	-27.67	2.71	0.53	-29.90	7.09
45	-0.46	-29.55		-0.50	-31.96	
46	0.48	-29.55	2.67	0.53	-31.96	7.07

47	-0.46	-31.43		-0.50	-34.01	
48	0.48	-31.43	2.66	0.53	-34.01	7.10
49	-0.46	-33.32		-0.50	-36.07	
50	0.48	-33.32	2.63	0.53	-36.07	7.11
51	-0.46	-35.2		-0.50	-38.13	
52	0.48	-35.2	2.60	0.53	-38.13	7.11

Tabla 5.2: Solución de viga en cantiliver para una discretización de 50 elementos y 52 nodos: (a) resultados del programa desarrollado en este trabajo; (b) el error es comparado con respecto a la solución analítica de la tabla 5.1.

Analizando los resultados registrados en las tablas 5.1 y 5.2 se observa que de acuerdo con la numeración de los nodos, los desplazamientos longitudinales indican que los elementos han girado con respecto al eje neutro en dirección perpendicular al plano analizado. Esto quiere decir que superficies verticales sobre la viga permanecen planas antes y después de aplicada la deformación¹.

Con respecto a los desplazamientos transversales se observan diferentes comportamientos para cada caso de carga:

- Cuando se analiza sólo el peso de la viga, el promedio de la variación en las soluciones es de 2.79%, excepto en los nodos cercanos al empotramiento de la viga, donde hay una variación máxima del 9.09%, para luego bajar y oscilar alrededor del valor ya mencionado.
- Para la condición que incluye el peso del motor, la variación va en aumento a medida que se aleja del empotramiento, aunque se aprecia una razón de crecimiento menor con respecto de la presentada cerca del empotramiento.

En términos generales se puede decir que la solución numérica obtenida por el método de elemento finito presenta una semejanza cualitativa aceptable con respecto a la solución analítica de la mecánica de materiales, al menos en la primera condición de carga. Con respecto a la segunda condición de carga sí hay una diferencia mayor, aunque ambas soluciones muestran la misma tendencia de crecimiento que va de acuerdo con el fenómeno físico. Así pues, la validación de los modelos numéricos y la verificación de la solución para el caso estático quedan justificadas por el análisis anterior. En las siguientes figuras puede apreciar gráficamente el comportamiento de ambas soluciones.

¹ De hecho, esta es una hipótesis de la mecánica de materiales, empleada en la deducción de la ecuación de esfuerzos en vigas.

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

Solución analítica

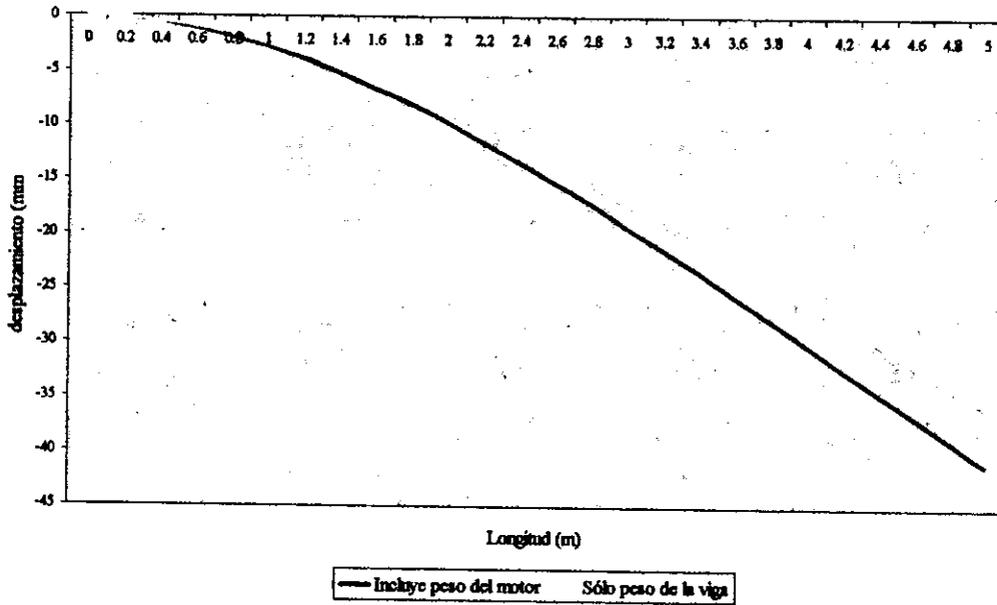


fig. 5-10: Solución analítica.

Solución numérica

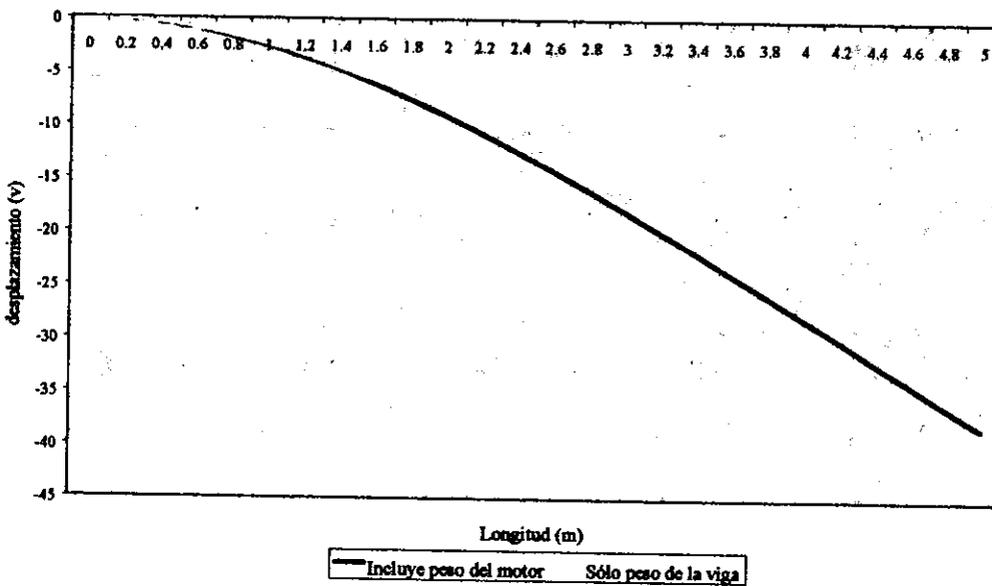


fig. 5-11: Solución numérica.

Variación de la solución

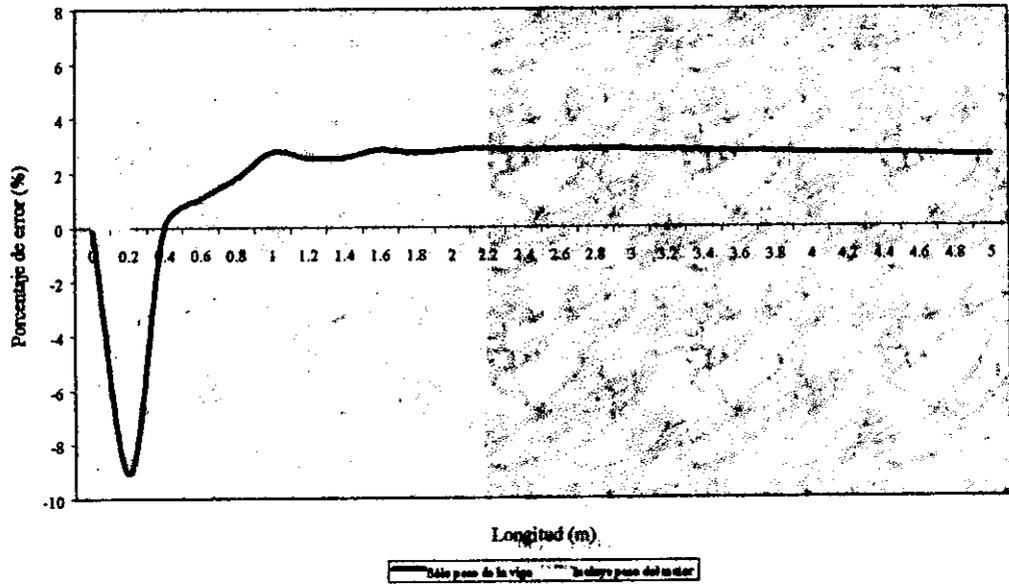


fig. 5-12: Variación de la solución.

aproximación de la solución

aproximación de la solución

Conclusiones.

La deducción de las ecuaciones de campo de la elasticidad lineal, a partir de la presentación axiomática de la mecánica del medio continuo, fue un trabajo importante por el hecho de conocer todo el proceso de deducción, el cual permite identificar puntos clave para el desarrollo de otros tipos de problemas diferenciados por aspectos tales como leyes constitutivas, linealidad-no linealidad, pequeñas-grandes deformaciones, entre otros. Cabe señalar que las ecuaciones obtenidas, al ser contrastadas con las expresiones deducidas con un enfoque práctico, [16, pp. 120-170], [21, pp. 199-200] y [22, p. 74], resultaron ser las mismas.

El uso de las matemáticas relacionadas con la aplicación del método de elemento finito, permite ver su alcance como técnica de interpolación a la solución de ecuaciones diferenciales, sin importar el modelo físico en cuestión [28, p. 16]. Con esto se identifica claramente el concepto de modelación matemática [3] por un lado, y por otro lado la herramienta utilizada para la solución del modelo. Al respecto cabe resaltar que a lo largo de este trabajo se evitó el uso de expresiones como: matriz de rigidez, matriz de masas y vector de cargas. Estas expresiones muy difundidas en la literatura deben su origen a las primeras aplicaciones del método en análisis estructural y a los principios de ingeniería usados en su implementación tales como: energía de deformación y energía potencial mínima [7, pp. 124-127], [26, pp. 461-470, 480-484], y a las interpretaciones prácticas de éste [26, pp. 484-490], [28, pp. 9-11, 122].

El modelo numérico de elasticidad dinámica lineal, presentado en este trabajo como aportación principal, fue obtenido a partir de la teoría general, y aunque muestra rasgos similares a otros modelos de la literatura [14], [18], [21], [24], tiene sus características únicas. Algunas ventajas de la formulación presentada con respecto a la encontrada en la literatura son:

- Posee una formulación matemática sólida.
- Su formulación global facilita la implementación computacional dejando el problema de ensamble de matrices, derivado de la formulación local (de elementos), al diseño de algoritmos de construcción de las matrices.
- Su construcción permite establecer una relación numérica-experimental entre el proceso de modelado y la medición física de variables. La instrumentación experimental puede dar información en la frontera, la cual, al ser considerada en el modelo numérico, ayuda a conocer el estado al interior del cuerpo.

Los aspectos estrictamente computacionales como verificación de datos de entrada y verificación de resultados parciales en los módulos de pre-procesamiento y procesamiento, así como el manejo numérico y gráfico de variables primarias en el módulo de post-procesamiento, fueron

tomados muy en cuenta en el diseño de los algoritmos de programación. Aunque hay que decir que el autor no tiene una preparación profesional en el campo, por lo que quizá algunos de esos puntos no fueron resueltos de manera óptima. Sin embargo, el sistema ofrece la posibilidad de, sin alteraciones fundamentales, proceder a su mejora y optimización.

Una aplicación muy práctica de este modelo numérico, y que está relacionada con la tercer ventaja señalada arriba es la siguiente:

- la medición local de deformaciones o desplazamientos en la frontera de los cuerpos como datos de entrada al modelo. La aplicación del método de elemento finito permitirá conocer variables del mismo tipo o de otro inclusive, utilizando técnicas de interpolación y las ecuaciones constitutivas del material. Es importante aclarar que este aspecto ya había sido señalado antes [27, pp. 1-2, 36], pero en el presente trabajo se establecen más claramente los puntos del proceso de modelación donde pueden ser incluidos estos datos, así como su manejo posterior.

Este trabajo sienta las bases para la modelación numérica del problema elastodinámico lineal. Se pretende que en desarrollos posteriores se realice la modelación de otro tipo de problemas con los mismos marcos teóricos utilizados aquí, como pueden ser de tipo eléctrico, magnético, térmico, entre otros. Los programas desarrollados poseen características que no han sido consideradas en otros trabajos, a su vez, esos trabajos incluyeron aspectos no considerados ahora [19], por lo que sería interesante una complementación para dar continuidad a este tipo de trabajos.

La documentación e implantación del análisis numérico, experimental y analítico, así como la inclusión de la informática, podrá servir para la creación de un laboratorio en el que se puedan validar y verificar los modelos. Esto sólo es posible con el trabajo en equipo, en el que es importante la conjunción de diversas áreas de la ciencia, esto es, el trabajo interdisciplinario en colectivo.

Bibliografía.

- [1] AUBIN, Jean Pierre, et al; Applied Nonlinear Analysis; John Wiley & Sons; USA; 1994.
- [2] BEER, Ferdinand P. et al; Mecánica de Materiales; Segunda Edición; McGraw-Hill Interamericana; 1993; Colombia.
- [3] CARRERA, Jorge; Methodological conceptualization of mathematical modelling; *Mathl. Comput. Modelling* Vol. 16, No. 12, pp. 19-28; Pergamon Press; Great Britain; 1992.
- [4] CARRERA, Jorge; Introducción al Método de Elemento Finito y a sus Fundamentos Matemáticos; DEPMI-UNAM; 1997.
- [5] CIARLET, P. G., et al; Handbook of Numerical Analysis, Volume II: Finite Element Method (Part 1); Elsevier Science Publishers B. V.; North-Holland; 1991.
- [6] DALLY, James W. et al; Experimental Stress Analysis; Third Edition; McGraw-Hill International Editions; 1991; Singapore.
- [7] DIETER, George; Engineering Design. A materials and processing Approach; Mc Graw-Hill Inc.; Japan; 1983.
- [8] FENNER, D. N.; Engineering Stress Analysis. A Finite Element Approach with FORTRAN 77 Software; Ellis-Horwood Ltd; England; 1987.
- [9] GURTIN, Morton; An Introduction to Continuum Mechanics; Academic Press; New York; 1981.
- [10] GRIFFEL, D. H.; Applied Functional Analysis; Ellis-Horwood Ltd; England; 1981.
- [11] HUTSON, V.; et al; Applications of Functional Analysis and Operator Theory; Academic Press; London; 1980.
- [12] HEARN, Donald, et al; Gráficas por Computadora: Prentice-Hall Hispanoamericana, S. A.; México; 1988.
- [13] HUEBNER, Kenneth H.; The Finite Element Method for Engineers; Second Edition; John Wiley & Sons; New York; 1982.
- [14] KNIGHT, Charles E. The Finite Element Method in Mechanical Design; PWS-KENT Publishing Company; Boston, Massachusetts; 1993.
- [15] PERRY, Greg; C by Example, Special Edition; Que Co.; USA; 1993.
- [16] POPOV, Egor P.; Introducción a la Mecánica de Sólidos; Ed. Limusa; México; 1983.
- [17] RAO, Singiresu S.; Mechanical Vibrations; Second Edition; Addison-Wesley Publishing Company; USA; 1990.
- [18] REDDY, J. N.; An Introduction to the Finite Element Method; McGraw-Hill Publishing Company; USA; 1984.
- [19] ROJANO, Abraham; Modelación de una carcasa de transmisión con elemento finito; Tesis DEPMI-UNAM; México; 1992.
- [20] ROJANI, Kamal B.; Programming in C with numerical methods for engineers; Prentice-Hall, New Jersey, 1996.
- [21] ROSS, C. T. F.; Finite Element Methods in Engineering Science; Ellis Horwood Limited; England; 1990.
- [22] SÁNCHEZ, José Luis; Mecánica del Medio Continuo; Textos Universitarios UNAM; México; 1972.

- [23] SCHILDT, Herbert; C: The Complete Reference; Mc Graw-Hill; Berkeley, Ca; 1987.
- [24] SPYRAKOS, Constantine C.; Finite Element Modeling in Engineering Practice. Includes Examples with ALGOR; West Virginia University Press; Morgantown, W. V.; 1994.
- [25] STASA, Frank L.; Applied Finite Element Analysis for Engineers; Holt, Rinehart and Winston Series in Mechanical Design; New York; 1985.
- [26] TIMOSHENKO, S. P.; et al; Mecánica de Materiales; UTEHA; México; 1979.
- [27] VÁZQUEZ, Rubén; Modelación mecánica de un disco sujeto a compresión diametral; Tesis DEPMI-UNAM; México; 1996.
- [28] VELA, L.; Aplicación del Método de los Elementos Finitos en el Diseño Mecánico; Tesis Universidad Autónoma de Zacatecas; 1996.
- [29] ZIENKIEWICZ, O. C., et al; El Método de los Elementos Finitos. Formulación básica y problemas lineales, Volumen 1; 4ta. Edición; McGraw-Hill/Interamericana de España; Barcelona; 1994.
- [30] ZIENKIEWICZ, O. C., et al; El Método de los Elementos Finitos, Volumen 2; 4ta. Edición; McGraw-Hill/Interamericana de España; Barcelona; 1995.

Apéndice.

A.1 DEFORMACIONES EN COORDENADAS CARTESIANAS.

Deformaciones básicas.

Sea $\mathcal{B} \subset \mathcal{E}$ un cuerpo sujeto a una deformación homogénea f . Dado un marco de coordenadas cartesianas formado por la base canónica $\{e_1, e_2\}$ en \mathcal{R}^2 y un punto $o \in \mathcal{E}$ llamado origen, las coordenadas de un punto $p \in \mathcal{B}$ con respecto al marco de coordenadas cartesianas están dadas por el par (p_1, p_2) . Las nuevas coordenadas del punto p después de la deformación, dadas por el par $(f(p_1), f(p_2))$, son obtenidas según la naturaleza de f de la siguiente manera [12, p. 115]:

Traslación. Sea f una traslación. Las coordenadas del punto $f(p)$ están dadas por:

$$f(p) = p + u \quad (1)$$

$$f \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} + \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} p_1 + u_1 \\ p_2 + u_2 \end{pmatrix},$$

donde u es un vector constante cuyas componentes u_1 y u_2 son las cantidades de traslación del punto p en las direcciones e_1 y e_2 , respectivamente.

Rotación. Sea f una rotación alrededor de un punto fijo q . Las coordenadas del punto $f(p)$ están dadas por:

$$f(p) = q + R(p - q) \quad (2)$$

$$f \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} + \begin{pmatrix} \cos\theta & -\text{sen}\theta \\ \text{sen}\theta & \cos\theta \end{pmatrix} \begin{pmatrix} p_1 - q_1 \\ p_2 - q_2 \end{pmatrix}$$

donde R es una rotación que se especifica con un ángulo de rotación $\theta \in \mathcal{R}$, el cual determina la cantidad de rotación de cada punto $p \in \mathcal{B}$ con respecto a otro punto fijo q , el cual permanece inalterado después de la deformación y puede o no ser parte de \mathcal{B} . La convención de rotación es que si $\theta > 0$ la rotación es en sentido antihorario, mientras que si $\theta < 0$ la rotación es en el sentido opuesto.

Elongación. Sea f una elongación desde un punto fijo q . Las coordenadas del punto $f(p)$ están dadas por:

$$f(p) = q + U(p - q)$$

(3)

$$f \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} + \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} p_1 - q_1 \\ p_2 - q_2 \end{pmatrix}$$

donde U es una elongación. El control de la deformación se logra mediante la elección de un punto fijo q , que permanece inalterado después de la deformación y puede o no ser parte de \mathbf{B} . Además, las cantidades $\lambda_1, \lambda_2 \in \mathfrak{R}$ son las extensiones en las direcciones e_1 y e_2 , respectivamente. De acuerdo con el valor de las extensiones se tiene que si:

- a) $\lambda_i > 0$, el cuerpo \mathbf{B} es aumentado de tamaño en la dirección i ,
- b) $\lambda_i < 0$, el cuerpo \mathbf{B} es reducido de tamaño en la dirección i ,
- c) $\lambda_1 = \lambda_2$, el cuerpo \mathbf{B} es extendido uniformemente. Particularmente, si $\lambda_1 = \lambda_2 = 1$, el cuerpo no se altera.

Deformaciones compuestas.

Debido al lema 1.1 de deformaciones homogéneas¹, cualquier deformación homogénea puede ser representada por una secuencia traslación-rotación-elongación, sin importar el orden de ejecución.

Sea f una deformación homogénea de un cuerpo \mathbf{B} compuesta por la secuencia traslación-rotación-elongación. La nueva región del espacio euclidiano \mathcal{E} que ocupa \mathbf{B} después de la deformación está dada por:

$$f_U(\mathbf{B}) = f_R(f_T(\mathbf{B})) \quad (4)$$

donde:

f_T es una traslación del cuerpo \mathbf{B} ,

f_R es una rotación del cuerpo trasladado $f_T(\mathbf{B})$,

f_U es una elongación del cuerpo rotado $f_R(f_T(\mathbf{B}))$.

Deformación compuesta. Sea f una deformación homogénea con punto fijo q , compuesta por la secuencia traslación-rotación-elongación. Sea $p \in \mathbf{B}$, las coordenadas del punto $f(p)$ en cada deformación básica están dadas por:

$$f_T(p) = p + u,$$

¹ 1.1 Cinemática, p. 1.

$$f_R(f_T(p)) = q + R((p + u) - q),$$

$$f_U(f_R(f_T(p))) = q + U(q + R((p + u) - q)).$$

El punto fijo q es el mismo tanto para la elongación como para la rotación. Puesto que se ha establecido que $f = f_U \circ f_R \circ f_T$, entonces:

$$f(p) = f_U(f_R(f_T(p))) = q + U(R((p + u) - q)). \quad (5)$$

Expresando la forma de la elongación U , de la rotación R y del campo vectorial de desplazamiento u , así como las coordenadas de los puntos p y q , tenemos:

$$f \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} + \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} \cos\theta & -\text{sen}\theta \\ \text{sen}\theta & \cos\theta \end{pmatrix} \begin{pmatrix} (p_1 + u_1) - q_1 \\ (p_2 + u_2) - q_2 \end{pmatrix},$$

$$f \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} + \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} (p_1 + u_1 - q_1) \cos\theta - (p_2 + u_2 - q_2) \text{sen}\theta \\ (p_1 + u_1 - q_1) \text{sen}\theta + (p_2 + u_2 - q_2) \cos\theta \end{pmatrix},$$

$$f \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} + \begin{pmatrix} \lambda_1 [(p_1 + u_1 - q_1) \cos\theta - (p_2 + u_2 - q_2) \text{sen}\theta] \\ \lambda_2 [(p_1 + u_1 - q_1) \text{sen}\theta + (p_2 + u_2 - q_2) \cos\theta] \end{pmatrix};$$

reacomodando términos:

$$f \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} q_1 + \lambda_1 [(u_1 - q_1) \cos\theta - (u_2 - q_2) \text{sen}\theta] + \lambda_1 \cos\theta p_1 - \lambda_1 \text{sen}\theta p_2 \\ q_2 + \lambda_2 [(u_1 - q_1) \text{sen}\theta + (u_2 - q_2) \cos\theta] + \lambda_2 \text{sen}\theta p_1 + \lambda_2 \cos\theta p_2 \end{pmatrix}. \quad (6)$$

De manera sintetizada, la función de deformación f tiene la siguiente forma:

$$f \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} c_1 + b_1 p_1 + a_1 p_2 \\ c_2 + b_2 p_1 + a_2 p_2 \end{pmatrix}, \quad (7)$$

donde los coeficientes a , b , c , dependen de las cantidades de traslación, rotación y elongación de la deformación compuesta. Cualquier cuerpo **B** podrá ser deformado en la secuencia traslación-rotación-elongación mediante la función f , de acuerdo a los parámetros de cada deformación homogénea básica.

Si en particular, el origen $o \in \mathcal{E}$ es trasladado al punto fijo de la rotación y elongación, las componentes del vector de desplazamiento u coinciden con las coordenadas del punto fijo q y la deformación compuesta se vuelve invariante ante la traslación:

$$f \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} q_1 + \lambda_1 \cos\theta p_1 - \lambda_1 \operatorname{sen}\theta p_2 \\ q_2 + \lambda_2 \operatorname{sen}\theta p_1 + \lambda_2 \cos\theta p_2 \end{pmatrix}. \quad (8)$$

La expresión anterior no depende de la traslación, solamente de las cantidades de rotación y elongación, y puede ser sintetizada de la misma forma que en (7).

Como una aplicación de la ecuación (8), en la siguiente tabla se presentan los parámetros necesarios para construir las funciones de deformación de cada elemento finito geométrico de la discretización propuesta en el capítulo 4 de este trabajo, para $L = 5.5$ y $a = 0.5$.

Elemento finito i	Punto fijo		Extensiones		Rotación θ_i (rad)
	q_{i1}	q_{i2}	λ_{i1}	λ_{i2}	
1	0	0	0.5	0.5	0
2	0.5	0.5	"	"	π
3	0.5	0	"	"	0
4	1	0.5	"	"	π
5	1	0	"	"	0
6	1.5	0.5	"	"	π
7	1.5	0	"	"	0
8	2	0.5	"	"	π
9	2	0	"	"	0
10	2.5	0.5	"	"	π
11	2.5	0	"	"	0
12	3	0.5	"	"	π
13	3	0	"	"	0
14	3.5	0.5	"	"	π
15	3.5	0	"	"	0
16	4	0.5	"	"	π
17	4	0	"	"	0
18	4.5	0.5	"	"	π
19	4.5	0	"	"	0
20	5	0.5	"	"	π
21	5	0	"	"	0
22	5.5	0.5	"	"	π

Tabla A.1-1: Parámetros de las funciones de deformación.

Con los datos de la tabla anterior, las funciones de deformación son las siguientes:

$$f_1 \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} x \\ y \end{pmatrix} \quad f_2 \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1-x \\ 1-y \end{pmatrix} \quad f_3 \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1+x \\ y \end{pmatrix}$$

$$f_4 \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 2-x \\ 1-y \end{pmatrix}$$

$$f_5 \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 2+x \\ y \end{pmatrix}$$

$$f_6 \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 3-x \\ 1-y \end{pmatrix}$$

$$f_7 \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 3+x \\ y \end{pmatrix}$$

$$f_8 \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 4-x \\ 1-y \end{pmatrix}$$

$$f_9 \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 4+x \\ y \end{pmatrix}$$

$$f_{10} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 5-x \\ 1-y \end{pmatrix}$$

$$f_{11} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 5+x \\ y \end{pmatrix}$$

$$f_{12} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 6-x \\ 1-y \end{pmatrix}$$

$$f_{13} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 6+x \\ y \end{pmatrix}$$

$$f_{14} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 7-x \\ 1-y \end{pmatrix}$$

$$f_{15} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 7+x \\ y \end{pmatrix}$$

$$f_{16} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 8-x \\ 1-y \end{pmatrix}$$

$$f_{17} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 8+x \\ y \end{pmatrix}$$

$$f_{18} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 9-x \\ 1-y \end{pmatrix}$$

$$f_{19} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 9+x \\ y \end{pmatrix}$$

$$f_{20} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 10-x \\ 1-y \end{pmatrix}$$

$$f_{21} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 10+x \\ y \end{pmatrix}$$

$$f_{22} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 11-x \\ 1-y \end{pmatrix} .$$

A.2 ESQUEMAS DE APROXIMACIÓN PARA LA DERIVADA TEMPORAL.

Cuando un cuerpo elástico está sujeto a cargas dinámicas se presentan dos tipos de fuerzas de cuerpo distribuidas que deben tomarse en cuenta: fuerzas de inercia, proporcionales a la aceleración y caracterizadas por el principio de D'Alembert, $F = -\rho \ddot{u}$; y fuerzas de fricción o amortiguamiento, difíciles de caracterizar en general pero aproximadas por una relación proporcional a la velocidad, $F = c \dot{u}$.

Una forma de resolver el problema dinámico es utilizando esquemas de recursión o aproximación en el tiempo, los cuales relacionan los valores de u , \dot{u} y \ddot{u} en un instante de tiempo dado t , con los valores de esas variables en un tiempo posterior $t + \Delta t$, donde Δt es un pequeño incremento en el tiempo. Los esquemas de aproximación comienzan la solución del problema con las condiciones iniciales en $t = 0$ y continúan paso a paso hasta que se alcanza la longitud deseada del tiempo de análisis [18, pp. 233-245].

Existen dos esquemas principales de solución basados en los métodos de diferencias finitas y método de elemento finito. Algunos esquemas de aproximación para derivadas temporales derivados de ambos métodos son [29, pp. 588-613]:

- a) Esquemas de dos puntos de recurrencia. Son aplicables a problemas que involucran sólo la primera derivada temporal.
 - a.1) Método de diferencias finitas:
 - a.1.1) Diferencias hacia adelante (método de Euler).
 - a.1.2) Diferencias hacia atrás.
 - a.1.3) Diferencias centrales (método de Crank-Nicolson).
 - a.2) Método de elemento finito:
 - a.1.1) Método de residuos ponderados (método de Galerkin) [30, pp. 385-419].

- b) Esquemas de tres puntos de recurrencia. Son aplicables a problemas que involucran la segunda derivada temporal. En este tipo de esquemas son necesarios los valores de la función en dos pasos de tiempo anteriores para determinar el valor de la función en el tiempo actual. Es en esta situación donde se introducen las condiciones iniciales y de frontera del problema.
 - b.1) Método de diferencias finitas:
 - b.1.1) Método de inicialización de Euler.
 - b.1.2) Método de inicialización de Crank-Nicolson.
 - b.2) Método de elemento finito:
 - b.1.1) Método de residuos ponderados (método de Galerkin).

En los esquemas de diferencias finitas, el método de Crank-Nicolson suele ser más preciso que el método de Euler.

A.3 INTEGRACIÓN NUMÉRICA SOBRE DOMINIOS TRIANGULARES.

Funciones polinomiales.

Sean $\varphi_i : \Omega \rightarrow \mathfrak{R}$ las funciones globales de interpolación de la forma:

$$\varphi_i(x, y) = a_i x + b_i y + c_i \quad (1)$$

con $a_i, b_i, c_i \in \mathfrak{R}$, donde Ω es una región regular. Entonces:

$$\begin{aligned} \varphi_i(x, y) \cdot \varphi_j(x, y) &= a_i a_j x^2 + b_i b_j y^2 + (a_i b_j + b_i a_j) xy + \\ &\quad (a_i c_j + c_i a_j) x + (b_i c_j + c_i b_j) y + c_i c_j, \\ &= A x^2 + B y^2 + C xy + D x + E y + F; \end{aligned} \quad (2)$$

y

$$\varphi_i(x, y) \cdot x = a_i x^2 + b_i xy + c_i x. \quad (3)$$

Dominios triangulares.

Sean los dominios triangulares definidos como sigue:

$$\Omega_1 = \left\{ (x, y) \mid y \leq y_2 + m(x - x_2); x_1 \leq x \leq x_2, y_1 \leq y \leq y_3, m = \frac{y_3 - y_2}{x_3 - x_2} \right\},$$

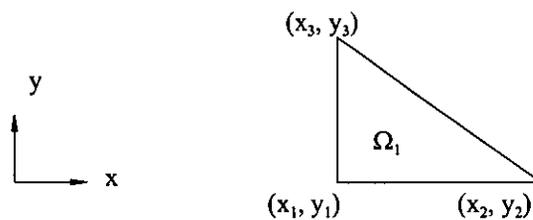


fig. A.3-1: Elemento finito triangular TIPO 1.

y

$$\Omega_2 = \left\{ (x, y) \mid y \geq y_3 + m(x - x_3); x_2 \leq x \leq x_1, y_3 \leq y \leq y_2, m = \frac{y_2 - y_3}{x_2 - x_3} \right\},$$

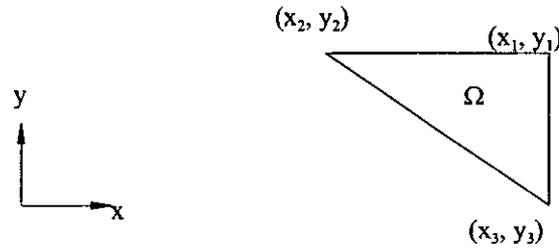


fig. A.3-2: Elemento finito triangular TIPO 2.

La integración numérica de las funciones polinomiales (1), (2) y (3) sobre los dominios triangulares Ω_1 y Ω_2 , la cual necesaria para calcular los elementos de la matriz \mathbf{M} y los elementos del vector $\mathbf{f}(t)$ de un análisis de elemento finito, queda determinada por las expresiones establecidas a continuación. Hay que señalar que la integración de la función polinomial (3) sólo se hará sobre la frontera del elemento finito TIPO 2, pues es ese el tipo de elemento cargado.

Integración numérica.

$$\int_{\Omega_1} \varphi_i(\Omega_1) d\Omega_1 = \int_{x_1}^{x_2} \int_{y_2}^y \varphi_i(x, y) dy dx$$

$$= -\frac{m}{6} (x_2 - x_1)^2 [a_i(2x_1 + x_2) + b_i(mx_1 - mx_2 + 3y_2) + 3c_i] .$$

$$\int_{\Omega_1} \varphi_i(\Omega_1) \cdot \varphi_j(\Omega_1) d\Omega_1 = \int_{x_1}^{x_2} \int_{y_2}^y \varphi_i(x, y) \cdot \varphi_j(x, y) dy dx$$

$$= \frac{m}{24} (x_2 - x_1)^2 \left[2A(3x_1 + x_2)(x_1 - x_2) - \right.$$

$$2B(m^2 \{x_1 - x_2\}^2 + 4my_2 \{x_1 - x_2\} + 6y_2^2) -$$

$$C(m\{3x_1 + x_2\}\{x_1 - x_2\} + 4y_2 \{2x_1 + x_2\}) -$$

$$\left. 4D(2x_1 + x_2) - 4E(m\{x_1 - x_2\} + 3y_2) - 12F \right] .$$

$$\int_{\Omega_2} \varphi_i(\Omega_2) d\Omega_2 = \int_{x_2}^{x_3} \int_y^{y_1} \varphi_i(x, y) dy dx$$

$$= \frac{(x_2 - x_3)}{6} \left[a_i (m\{2x_2 + x_3\}\{x_2 - x_3\} + 3\{x_2 + x_3\}\{y_3 - y_1\}) + \right. \\ b_i (m^2\{x_2 - x_3\}^2 + 3my_3\{x_2 - x_3\} + 3\{y_3^2 - y_1^2\}) + \\ \left. c_i (3m\{x_2 - x_3\} + 6\{y_3 - y_1\}) \right] .$$

$$\int_{\partial\Omega_2} \varphi_i(\partial\Omega_2) \bullet x d\partial\Omega_2 = \int_{x_2}^{x_3} \varphi_i(x, y) \bullet x dx = \frac{a_i(x_3^3 - x_2^3)}{3} + \frac{(b_i y_2 + c_i)(x_3^2 - x_2^2)}{2} .$$

$$\int_{\Omega_2} \varphi_i(\Omega_2) \bullet \varphi_j(\Omega_2) d\Omega_2 = \int_{x_2}^{x_3} \int_y^{y_1} \varphi_i(x, y) \bullet \varphi_j(x, y) dy dx$$

$$= \frac{(x_2 - x_3)}{24} \left[2A(m x_2 \{2x_2 + x_3\}\{x_2 - x_3\} + m\{x_2^3 - x_3^3\} + \right. \\ 4\{y_3 - y_1\}\{x_2^2 + x_2 x_3 + x_3^2\}) + \\ 2B(m^3\{x_2^3 - x_3^3\} + 3m\{2y_3^2 - m^2 x_2 x_3\}\{x_2 - x_3\} + \\ 4m^2 y_3\{x_2 - x_3\}^2 + 4\{y_3^3 - y_1^3\}) + \\ C(3m^2 x_2^2\{x_2 - x_3\} + m\{4y_3 - m x_3\}\{2x_2 + x_3\}\{x_2 - x_3\} + \\ 6\{y_3^2 - y_1^2\}\{x_2 + x_3\}) + \\ 4D(m\{2x_2 + x_3\}\{x_2 - x_3\} + 3\{y_3 - y_1\}\{x_2 + x_3\}) + \\ 4E(m^2\{x_2 - x_3\}^2 + 3my_3\{x_2 - x_3\} + 3\{y_3^2 - y_1^2\}) + \\ \left. 12F(m\{x_2 - x_3\} + 2\{y_3 - y_1\}) \right] .$$

A.4 SOLUCIÓN DE ECUACIONES LINEALES SIMULTÁNEAS.

Introducción.

Un sistema de ecuaciones simultáneas generalmente está dado en la forma:

$$\begin{array}{r}
 a_{11} x_1 + a_{12} x_2 + \dots + a_{1n} x_n = b_1 \\
 a_{21} x_1 + a_{22} x_2 + \dots + a_{2n} x_n = b_2 \\
 \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\
 a_{m1} x_1 + a_{m2} x_2 + \dots + a_{mn} x_n = b_m
 \end{array} \tag{1}$$

donde a_{ij} y b_i son coeficientes conocidos, y x_i son las incógnitas para las cuales las ecuaciones serán resueltas [20, pp. 872-913]. Puesto que cada x_i aparece en su primera potencia y no se multiplican entre sí, cada ecuación es lineal. De forma compacta, el sistema de ecuaciones puede representarse como

$$A_{m \times n} x_{n \times 1} = b_{m \times 1} \tag{2}$$

donde $A = [a_{ij}]$ representa el arreglo rectangular de los $m \times n$ coeficientes a_{ij} y se conoce como matriz de coeficientes, $x = [x_i]$ representa las n incógnitas x_i , y $b = [b_i]$ es el conjunto de los m coeficientes b_i . En general, existe un conjunto de x_i valores que al ser sustituidos en las ecuaciones (1) simultáneamente, son satisfechas todas ellas. Bajo ciertas circunstancias puede haber un número infinito de conjuntos de x_i valores que satisfagan las ecuaciones simultáneas, mientras que bajo otras circunstancias no hay un conjunto de x_i valores que satisfaga el sistema de ecuaciones.

Un sistema de ecuaciones puede ser *singular* o *no singular*, hecho caracterizado por el determinante de la matriz A . Si el determinante de la matriz es diferente de cero, el sistema es no singular. Sin embargo, algunas veces el sistema de ecuaciones puede estar cerca de la singularidad; esto sucede cuando el valor del determinante es muy pequeño. Tales sistemas de ecuaciones se conocen como *mal condicionados* y desde un punto de vista numérico computacional pueden llevar a resultados no confiables. Los sistemas mal condicionados se caracterizan por su sensibilidad a las condiciones iniciales, pues pequeños cambios en éstas pueden causar grandes cambios en el resultado.

Los métodos para la solución de ecuaciones lineales simultáneas pueden clasificarse en dos categorías: *métodos directos* y *métodos indirectos*. Los métodos directos se refieren a los procedimientos numéricos que proporcionarán una solución en un número finito de pasos; entre ellos podemos mencionar el método de eliminación de Gauss. Los métodos indirectos son procedimientos iterativos que pueden o no converger a una solución; entre ellos tenemos el método iterativo de Gauss-Seidel el cual será discutido más adelante.

Comparación de los métodos directos e indirectos.

De acuerdo con la experiencia obtenida en la aplicación de los métodos de solución de sistemas de ecuaciones lineales, se tienen las siguientes observaciones [20, pp. 913-914]:

a) **Convergencia.**

El método directo es un método realizado en un número finito de pasos, lo que significa que, teóricamente convergerá a la solución para una matriz no singular. Los métodos indirectos no siempre convergen, su eficiencia sólo queda garantizada para sistemas en los cuales la matriz de coeficientes es diagonalmente dominante.

b) **Rapidez.**

En los métodos directos se puede saber de antemano el número de operaciones requeridas para llegar a la solución. En los métodos indirectos, el número de iteraciones depende de los coeficientes de la matriz así como de la precisión deseada, y por lo tanto no es posible predecir el número de iteraciones necesarias.

c) **Eficiencia computacional.**

Un método directo necesita todos los coeficientes de la matriz A y los coeficientes de b almacenados en la memoria de la computadora, mientras que un método indirecto maneja una sola ecuación a la vez, esto permite que el método indirecto resuelva grandes sistemas de ecuaciones.

d) **Error de redondeo.**

En particular, el método de eliminación de Gauss es susceptible a errores de redondeo, los cuales generalmente son más pequeños en el método iterativo de Gauss-Seidel.

e) **Porosidad.**

La porosidad en sistemas de ecuaciones lineales significa que la mayoría de los elementos de la matriz A son ceros. Las matrices porosas son difíciles de resolver usando la eliminación de Gauss. Para sistemas porosos es preferible el método de Gauss-Seidel puesto que no altera los coeficientes de la matriz A .

Matrices de elemento finito.

Las matrices que resultan de un análisis por el método de elemento finito deben reproducir, o al menos aproximar, las propiedades del operador diferencial del modelo físico. Si el operador es simétrico, la matriz resultante es simétrica también [4, pp. 346-347].

Además, con una numeración global adecuada de nodos y elementos, se puede lograr que la matriz quede estructurada en forma de matriz bandeda, puesto que la intersección de los soportes de las funciones globales se da solamente en un número muy limitado de otras funciones globales, esto da como resultado que la mayoría de los elementos de la matriz sean ceros.

Otra característica importante de las matrices de elemento finito es que son diagonalmente dominantes, esto es:

$$|a_{ii}| > \sum_{\substack{i=1 \\ i \neq j}}^n |a_{ij}| .$$

Esto se debe a que la interacción de una función global consigo misma siempre involucra más elementos finitos, pues se utiliza todo el soporte de la función, mientras que la interacción de funciones globales diferentes involucra sólo unos cuantos de ellos. Una matriz diagonalmente dominante es, numéricamente, mejor comportada.

De acuerdo con las características que presentan las matrices de elemento finito, resulta obvio que los métodos de solución más recomendables para resolver los sistemas de ecuaciones derivados de un análisis de elemento finito son los métodos indirectos o iterativos.

El método de Gauss-Seidel.

Un procedimiento iterativo consiste en hacer una aproximación inicial para el conjunto de valores x_i y, usando ésta obtener mejores aproximaciones a la solución. La convergencia del procedimiento dada una aproximación inicial queda garantizada si la matriz A es diagonalmente dominante [20, pp. 900-907].

Aún si la matriz no es diagonalmente dominante, el método iterativo puede converger si los elementos más grandes de la matriz están localizados sobre la diagonal. Por lo tanto, el requerimiento de dominancia diagonal es una condición suficiente pero no necesaria y es posible encontrar una solución a sistemas que posean una diagonal fuerte.

La implementación computacional del método iterativo de Gauss-Seidel debe habilitar la posibilidad de obtener una solución. Para lograr esto, se debe verificar que la matriz A sea diagonalmente dominante. Si la matriz A no es diagonalmente dominante se debe realizar un cambio de renglones. Para cada elemento diagonal de un renglón no dominante diagonalmente, se busca el renglón que contiene el elemento más grande sobre la columna y se intercambia con el renglón que contiene el elemento sobre la diagonal.

A.5 SOLUCIONES ANALÍTICAS.

Caso estático: Ecuación diferencial de la curva elástica.

De mecánica de materiales se sabe que una viga prismática sometida a carga transversal se flexiona adoptando una forma regida por la ecuación diferencial de la curva elástica, siempre y cuando el material no exceda el rango elástico y sea aplicable el principio de Saint-Venant [2, p. 476].

Para determinar la deflexión en cualquier punto sobre el eje neutro de la viga, se emplea la ecuación diferencial ordinaria lineal y de segundo orden que caracteriza a la curva elástica o forma que adopta la viga deformada:

$$\frac{d^2y(x)}{dx^2} = \frac{M(x)}{EI} \quad (1)$$

donde:

$y(x)$: deflexión o desplazamiento de la viga.

$M(x)$: momento flexionante a lo largo de la viga.

E : módulo de elasticidad del material.

I : momento de inercia de la sección transversal con respecto al eje neutro.

Para el caso de una viga en cantiliver sujeta a una carga concentrada cerca de su extremo libre, como se muestra en la fig. (A.5-1), el momento flexionante tiene la siguiente expresión:

$$M(x) = \frac{1}{2}\gamma ab(x-L)^2 - F(x-L) \quad (2)$$

donde:

F es el peso del motor,

L : longitud total de la viga.

a, b : alto y ancho, dimensiones de la sección transversal de la viga.

γ : peso específico del material de la viga.



fig. A.5-1: Modelo simplificado: viga en cantiliver.

Sustituyendo la ecuación (2) en (1) e integrando sobre la longitud de la viga, obtenemos la función para la deflexión de la viga que satisface condiciones de frontera $y'(0) = y(0) = 0$:

$$y(x) = \frac{x^2}{24EI} [\gamma ab(x^2 - 4xL + 6L^2) - 4F(x - 3L)] \quad (3)$$

donde $I = \frac{1}{12}ba^3$ es el momento de inercia para una sección prismática como la nuestra.

Caso dinámico: Vibración de un cuerpo armónicamente excitado.

Desde el punto de vista de las vibraciones mecánicas, nuestro problema podría verse como un sistema dinámico de masa-resorte no amortiguado de un grado de libertad sujeto a una fuerza de excitación armónica. La ecuación diferencial de movimiento correspondiente es [17, pp. 129-136]:

$$m\ddot{x}(t) + kx(t) = F(t) = F_0 \text{ sen } \omega t, \quad (4)$$

donde k y m son la rigidez efectiva y la masa del sistema, respectivamente. Con la solución de la ecuación (4) sólo se predice la amplitud de vibración de la viga en su extremo libre.

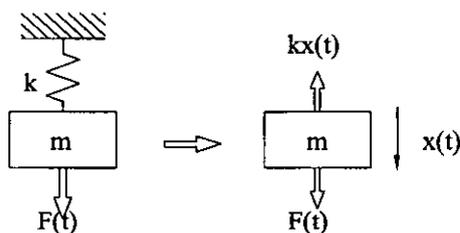


fig. A.5-2: Modelo simplificado: sistema masa-resorte no amortiguado.

La solución general de la ecuación (4) está dada por:

$$x(t) = x_0 \cos \omega_n t + \frac{v_0 - \omega X}{\omega_n} \text{ sen } \omega_n t + X \text{ sen } \omega t \quad (5)$$

donde:

$$X = \frac{F_0}{k - m\omega^2} = \frac{\delta_{st}}{1 - \left(\frac{\omega}{\omega_n}\right)^2},$$

$$\delta_{st} = F_0/k,$$

$x_0 = x(t=0) = -mg/k$ y $v_0 = \dot{x}(t=0)$ son las condiciones iniciales del problema,

$\omega_n = (k/m)^{1/2}$ se conoce como frecuencia natural del sistema,

ω es la frecuencia de la fuerza de excitación.

El valor de la rigidez efectiva k del sistema se obtiene de la ecuación (3), es el valor de F al hacer $y(L) = 1$. Entonces:

$$k = \frac{24EI - 3\gamma abL^4}{8L^3} . \tag{6}$$

La rigidez efectiva del sistema es debida al peso de la viga y a las propiedades elásticas del material. La masa m del sistema es solamente la del motor.

La razón ω/ω_n tiene una gran importancia puesto que la respuesta del sistema depende de su magnitud, presentándose tres casos diferentes descritos en la siguiente tabla.

CASO	RESPUESTA DINÁMICA
$0 < \omega/\omega_n < 1$	Se dice que la respuesta del sistema está en fase con la fuerza de excitación. La respuesta está dada por la ecuación (5), sin cambio.
$\omega/\omega_n > 1$	En este caso, se dice que la respuesta del sistema está π rad fuera de fase con la fuerza de excitación. Además, si $\omega/\omega_n \rightarrow \infty$, $X \rightarrow 0$, esto indica que la respuesta del sistema a una fuerza armónica de muy alta frecuencia es cercana a cero. La respuesta está dada por la ecuación (5) con el cambio $X = -X$.
$\omega/\omega_n = 1$	En este caso, llamado de resonancia, el valor de X se vuelve infinito por lo que la respuesta del sistema no puede ser determinada con la ecuación (5). En este caso se emplea la siguiente ecuación como respuesta:

Tabla A.5.1: Tipos de respuesta a la vibración.

$$x(t) = x_0 \cos \omega_n t + \frac{v_0}{\omega_n} \sin \omega_n t + \frac{\delta_{st} \omega_n t}{2} \cos \omega_n t . \tag{7}$$

De la ecuación anterior se ve que en resonancia, la amplitud de la respuesta $x(t)$ se incrementa indefinidamente con el tiempo.

A.6 MÉTODOS APROXIMADOS DE SOLUCIÓN.

Cada fenómeno natural, ya sea biológico, geológico o mecánico, puede describirse con ayuda de las leyes de la física en términos de ecuaciones diferenciales o integrales, las cuales relacionan varias variables del fenómeno [18, pp. 1-2].

La derivación de las ecuaciones que gobiernan cada fenómeno físico no es un trabajo tan difícil como lo es encontrar su solución por algún método exacto. En estos casos, los métodos aproximados son una buena alternativa para encontrar soluciones. Entre ellos podemos mencionar el método de diferencias finitas, métodos variacionales (como los métodos de Ritz, Galerkin, mínimos cuadrados, colocación, residuos ponderados) y el método de elemento finito.

El *método de diferencias finitas* consiste en aproximar la ecuación diferencial por cocientes de diferencias que involucran los valores de la solución en una malla discreta de puntos del dominio. Las ecuaciones discretas resultantes son resueltas al imponer las condiciones de frontera del problema. El método de diferencias finitas presenta las siguientes desventajas:

- inexactitud de las derivadas de la solución aproximada,
- dificultad para imponer condiciones de frontera sobre fronteras no rectas,
- dificultad para representar con exactitud dominios geoméricamente complejos,
- incapacidad para emplear mallas no uniformes y no rectangulares.

En la *solución variacional* de una ecuación diferencial, ésta es llevada a una forma variacional equivalente, y entonces se asume que la solución aproximada es una combinación lineal $\sum c_j \phi_j$ de funciones de aproximación dadas. Los coeficientes c_j se determinan de la forma variacional. La principal desventaja de estos métodos es que las funciones de aproximación para problemas con dominios arbitrarios son difíciles de construir, pues no existe un procedimiento sistemático para construirlas.

El *método de elemento finito* proporciona un procedimiento sistemático para la derivación de las funciones de aproximación, venciendo así la limitante de los métodos variacionales. El método de elemento finito posee dos características principales que lo hacen superior sobre otros métodos. Primero, un dominio geoméricamente complejo puede representarse como una colección de subdominios geoméricamente simples, llamados elementos finitos. Segundo, sobre cada elemento finito las funciones de aproximación se construyen a partir de la idea básica de que una función continua puede ser representada por una combinación lineal de polinomios algebraicos. Las funciones de aproximación se construyen usando conceptos básicos de interpolación, de hecho, éstas se conocen como funciones de interpolación. Los coeficientes involucrados en la combinación lineal vienen a ser los valores de la solución en un número finito de puntos, llamados nodos, sobre la frontera y el interior del dominio.

A.7 EJEMPLOS DE DISCRETIZACIÓN.

El cálculo de las coordenadas globales de la discretización triangular de un dominio rectangular, hecha por el pre-procesador, está basado en el número deseado de divisiones en direcciones horizontal y vertical. Sean m y n el número de renglones y columnas deseado, el número total de nodos y elementos está dado por las ecuaciones:

$$n^* = (m + 1)(n + 1) \text{ y } h = 2mn,$$

respectivamente. Para mostrar la generación de coordenadas por el módulo pre-procesador, enseguida se listan los resultados obtenidos para las discretizaciones de la fig. (A.7-1).

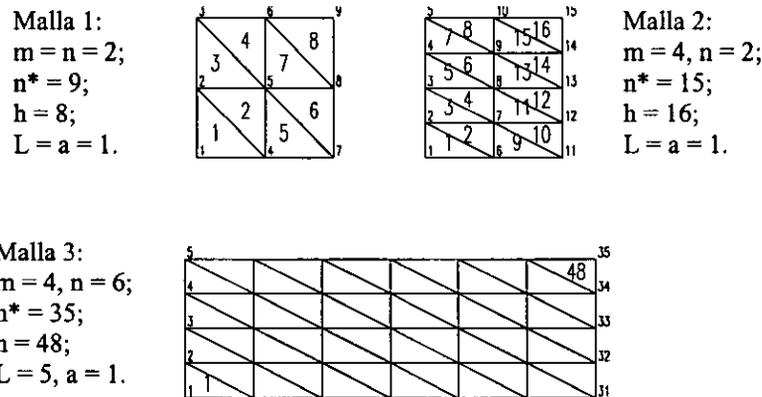


fig. A.7-1: Discretización de dominios triangulares.

Malla 1.

3. Generación de coordenadas nodales globales.
 Densidad de la malla: 2×2 .
 8 elementos, 9 nodos.

Elemento	x1	y1	x2	y2	x3	y3
1	0.0000	0.0000	0.5000	0.0000	0.0000	0.5000
2	0.5000	0.5000	0.0000	0.5000	0.5000	0.0000
3	0.0000	0.5000	0.5000	0.5000	0.0000	1.0000
4	0.5000	1.0000	0.0000	1.0000	0.5000	0.5000
5	0.5000	0.0000	1.0000	0.0000	0.5000	0.5000
6	1.0000	0.5000	0.5000	0.5000	1.0000	0.0000
7	0.5000	0.5000	1.0000	0.5000	0.5000	1.0000
8	1.0000	1.0000	0.5000	1.0000	1.0000	0.5000

Malla 2.

3. Generación de coordenadas nodales globales.
 Densidad de la malla: 2×4 .
 16 elementos, 15 nodos.

Elemento	x1	y1	x2	y2	x3	y3
1	0.0000	0.0000	0.5000	0.0000	0.0000	0.2500
2	0.5000	0.2500	0.0000	0.2500	0.5000	0.0000
3	0.0000	0.2500	0.5000	0.2500	0.0000	0.5000
4	0.5000	0.5000	0.0000	0.5000	0.5000	0.2500

5	0.0000	0.5000	0.5000	0.5000	0.0000	0.7500
6	0.5000	0.7500	0.0000	0.7500	0.5000	0.5000
7	0.0000	0.7500	0.5000	0.7500	0.0000	1.0000
8	0.5000	1.0000	0.0000	1.0000	0.5000	0.7500
9	0.5000	0.0000	1.0000	0.0000	0.5000	0.2500
10	1.0000	0.2500	0.5000	0.2500	1.0000	0.0000
11	0.5000	0.2500	1.0000	0.2500	0.5000	0.5000
12	1.0000	0.5000	0.5000	0.5000	1.0000	0.2500
13	0.5000	0.5000	1.0000	0.5000	0.5000	0.7500
14	1.0000	0.7500	0.5000	0.7500	1.0000	0.5000
15	0.5000	0.7500	1.0000	0.7500	0.5000	1.0000
16	1.0000	1.0000	0.5000	1.0000	1.0000	0.7500

Malla 3.

3. Generación de coordenadas nodales globales.
 Densidad de la malla: 6×4 .
 48 elementos, 35 nodos.

Elemento	x1	y1	x2	y2	x3	y3
1	0.0000	0.0000	0.8333	0.0000	0.0000	0.2500
2	0.8333	0.2500	0.0000	0.2500	0.8333	0.0000
3	0.0000	0.2500	0.8333	0.2500	0.0000	0.5000
4	0.8333	0.5000	0.0000	0.5000	0.8333	0.2500
5	0.0000	0.5000	0.8333	0.5000	0.0000	0.7500
6	0.8333	0.7500	0.0000	0.7500	0.8333	0.5000
7	0.0000	0.7500	0.8333	0.7500	0.0000	1.0000
8	0.8333	1.0000	0.0000	1.0000	0.8333	0.7500

9	0.8333	0.0000	1.6667	0.0000	0.8333	0.2500	29	2.5000	0.5000	3.3333	0.5000	2.5000	0.7500
10	1.6667	0.2500	0.8333	0.2500	1.6667	0.0000	30	3.3333	0.7500	2.5000	0.7500	3.3333	0.5000
11	0.8333	0.2500	1.6667	0.2500	0.8333	0.5000	31	2.5000	0.7500	3.3333	0.7500	2.5000	1.0000
12	1.6667	0.5000	0.8333	0.5000	1.6667	0.2500	32	3.3333	1.0000	2.5000	1.0000	3.3333	0.7500
13	0.8333	0.5000	1.6667	0.5000	0.8333	0.7500	33	3.3333	0.0000	4.1667	0.0000	3.3333	0.2500
14	1.6667	0.7500	0.8333	0.7500	1.6667	0.5000	34	4.1667	0.2500	3.3333	0.2500	4.1667	0.0000
15	0.8333	0.7500	1.6667	0.7500	0.8333	1.0000	35	3.3333	0.2500	4.1667	0.2500	3.3333	0.5000
16	1.6667	1.0000	0.8333	1.0000	1.6667	0.7500	36	4.1667	0.5000	3.3333	0.5000	4.1667	0.2500
17	1.6667	0.0000	2.5000	0.0000	1.6667	0.2500	37	3.3333	0.5000	4.1667	0.5000	3.3333	0.7500
18	2.5000	0.2500	1.6667	0.2500	2.5000	0.0000	38	4.1667	0.7500	3.3333	0.7500	4.1667	0.5000
19	1.6667	0.2500	2.5000	0.2500	1.6667	0.5000	39	3.3333	0.7500	4.1667	0.7500	3.3333	1.0000
20	2.5000	0.5000	1.6667	0.5000	2.5000	0.2500	40	4.1667	1.0000	3.3333	1.0000	4.1667	0.7500
21	1.6667	0.5000	2.5000	0.5000	1.6667	0.7500	41	4.1667	0.0000	5.0000	0.0000	4.1667	0.2500
22	2.5000	0.7500	1.6667	0.7500	2.5000	0.5000	42	5.0000	0.2500	4.1667	0.2500	5.0000	0.0000
23	1.6667	0.7500	2.5000	0.7500	1.6667	1.0000	43	4.1667	0.2500	5.0000	0.2500	4.1667	0.5000
24	2.5000	1.0000	1.6667	1.0000	2.5000	0.7500	44	5.0000	0.5000	4.1667	0.5000	5.0000	0.2500
25	2.5000	0.0000	3.3333	0.0000	2.5000	0.2500	45	4.1667	0.5000	5.0000	0.5000	4.1667	0.7500
26	3.3333	0.2500	2.5000	0.2500	3.3333	0.0000	46	5.0000	0.7500	4.1667	0.7500	5.0000	0.5000
27	2.5000	0.2500	3.3333	0.2500	2.5000	0.5000	47	4.1667	0.7500	5.0000	0.7500	4.1667	1.0000
28	3.3333	0.5000	2.5000	0.5000	3.3333	0.2500	48	5.0000	1.0000	4.1667	1.0000	5.0000	0.7500

Listado del análisis.

MODELACION NUMERICA DE LA EVOLUCION DINAMICA DE
ESFUERZOS EN SOLIDOS.

ANALISIS ELASTODINAMICO LINEAL BIDIMENSIONAL POR EL METODO
DE ELEMENTO FINITO DE UNA VIGA EN CANTILEVER SUJETA A UN
SISTEMA DE CARGA ESTATICA Y DINAMICA EN SU EXTREMO LIBRE.

LA FUERZA DE EXCITACION, PRODUCIDA POR UN MOTOR ELECTRICO,
ES DE LA FORMA $F(t) = F_0 \text{sen}(wt)$.

LA MALLA DE ELEMENTO FINITO ES DEL TIPO HOMOGENEA FORMADA
POR ELEMENTOS FINITOS GEOMETRICOS TRIANGULO-RECTANGULARES DE
TRES NODOS Y DOS GRADOS DE LIBERTAD POR NODO.

Ing. Luciano Vela Martinez.

DIVISION DE ESTUDIOS DE POSGRADO. DEPARTAMENTO DE MECANICA.
FACULTAD DE INGENIERIA, U N A M .

Agosto de 1998.

Titulo de analisis: Analisis para tesis. Agosto 20, 1998.

1. Dimensiones generales de la viga (metros). L = 5.00 a = 0.10 b = 0.50	10	1.0000	0.1000	0.8000	0.1000	1.0000	0.0000
	11	1.0000	0.0000	1.2000	0.0000	1.0000	0.1000
	12	1.2000	0.1000	1.0000	0.1000	1.2000	0.0000
2. Propiedades del material (nu, E, densidad). nu = 0.33 E = 19999995904.00 N/m2 densidad = 7860.00 kg/m3	13	1.2000	0.0000	1.4000	0.0000	1.2000	0.1000
	14	1.4000	0.1000	1.2000	0.1000	1.4000	0.0000
	15	1.4000	0.0000	1.6000	0.0000	1.4000	0.1000
	16	1.6000	0.1000	1.4000	0.1000	1.6000	0.0000
3. Generación de coordenadas nodales globales. Densidad de la malla: 25 x 1. 50 elementos, 52 nodos.	17	1.6000	0.0000	1.8000	0.0000	1.6000	0.1000
	18	1.8000	0.1000	1.6000	0.1000	1.8000	0.0000
	19	1.8000	0.0000	2.0000	0.0000	1.8000	0.1000
	20	2.0000	0.1000	1.8000	0.1000	2.0000	0.0000
Elemento	21	2.0000	0.0000	2.2000	0.0000	2.0000	0.1000
1	22	2.2000	0.1000	2.0000	0.1000	2.2000	0.0000
2	23	2.2000	0.0000	2.4000	0.0000	2.2000	0.1000
3	24	2.4000	0.1000	2.2000	0.1000	2.4000	0.0000
4	25	2.4000	0.0000	2.6000	0.0000	2.4000	0.1000
5	26	2.6000	0.1000	2.4000	0.1000	2.6000	0.0000
6	27	2.6000	0.0000	2.8000	0.0000	2.6000	0.1000
7	28	2.8000	0.1000	2.6000	0.1000	2.8000	0.0000
8	29	2.8000	0.0000	3.0000	0.0000	2.8000	0.1000
9	30	3.0000	0.1000	2.8000	0.1000	3.0000	0.0000

31	3.0000	0.0000	3.2000	0.0000	3.0000	0.1000	9	9	11	10
32	3.2000	0.1000	3.0000	0.1000	3.2000	0.0000	10	12	10	11
33	3.2000	0.0000	3.4000	0.0000	3.2000	0.1000	11	11	13	12
34	3.4000	0.1000	3.2000	0.1000	3.4000	0.0000	12	14	12	13
35	3.4000	0.0000	3.6000	0.0000	3.4000	0.1000	13	13	15	14
36	3.6000	0.1000	3.4000	0.1000	3.6000	0.0000	14	16	14	15
37	3.6000	0.0000	3.8000	0.0000	3.6000	0.1000	15	15	17	16
38	3.8000	0.1000	3.6000	0.1000	3.8000	0.0000	16	18	16	17
39	3.8000	0.0000	4.0000	0.0000	3.8000	0.1000	17	17	19	18
40	4.0000	0.1000	3.8000	0.1000	4.0000	0.0000	18	20	18	19
41	4.0000	0.0000	4.2000	0.0000	4.0000	0.1000	19	19	21	20
42	4.2000	0.1000	4.0000	0.1000	4.2000	0.0000	20	22	20	21
43	4.2000	0.0000	4.4000	0.0000	4.2000	0.1000	21	21	23	22
44	4.4000	0.1000	4.2000	0.1000	4.4000	0.0000	22	24	22	23
45	4.4000	0.0000	4.6000	0.0000	4.4000	0.1000	23	23	25	24
46	4.6000	0.1000	4.4000	0.1000	4.6000	0.0000	24	26	24	25
47	4.6000	0.0000	4.8000	0.0000	4.6000	0.1000	25	25	27	26
48	4.8000	0.1000	4.6000	0.1000	4.8000	0.0000	26	28	26	27
49	4.8000	0.0000	5.0000	0.0000	4.8000	0.1000	27	27	29	28
50	5.0000	0.1000	4.8000	0.1000	5.0000	0.0000	28	30	28	29

29	29	31	30
30	32	30	31
31	31	33	32
32	34	32	33
33	33	35	34
34	36	34	35
35	35	37	36
36	38	36	37
37	37	39	38
38	40	38	39
39	39	41	40
40	42	40	41
41	41	43	42
42	44	42	43
43	43	45	44
44	46	44	45
45	45	47	46
46	48	46	47
47	47	49	48
48	50	48	49
49	49	51	50
50	52	50	51

4. Condiciones de frontera.

Nodos restringidos: 2 Elementos cargados: 3

Nodo	Numeración global
1	1
2	2

Elem	Numeración global
1	46
2	48
3	50

5. Naturaleza de la carga din mica.

m = 100.00 (kg)
 F0 = 5000.00 (N)
 w = 157.08 (rad/seg)
 Rigidez equivalente = 192771.27 (N/m)
 Frecuencia natural = 43.91 (Hz)
 Respuesta fuera de fase: w/wn = 3.58
 Puntos de discretización = 9
 Período de vibración = 0.1431 (seg)
 Paso de discretización del tiempo = 0.0179 (seg)

paso	t (seg)	F(t) (N)
1	0.0000	0.0000
2	0.0179	1628.2875
3	0.0358	-3079.0518
4	0.0537	4194.1235
5	0.0716	-4851.9355
6	0.0894	4980.7671
7	0.1073	-4566.5728
8	0.1252	3654.5095
9	0.1431	-2344.0203

6. Generación de elementos por asociación de nodos.

ELE	N1	N2	N3
1	1	3	2
2	4	2	3
3	3	5	4
4	6	4	5
5	5	7	6
6	8	6	7
7	7	9	8
8	10	8	9

7. Construcción de la matriz K

Tamaño: 104 x 104
 Razón MAX/MIN = 315126054912.000000 / 55285272576.000000 = 5.700000
 MATRIZ SIMETRICA

8. Construcción de la matriz M

Tamaño: 104 x 104
 Razón MAX/MIN = 45280.199219 / 6.547377 = 6915.777381
 MATRIZ SIMETRICA

9. Construcción de la matriz equivalente

10. Construcción del vector f(t)

Tamaño: 104 x 1
 t = 0.000000
 Razón MAX/MIN = 0.000000 / -2169.976074 = -0.000000
 t = 0.017900
 Razón MAX/MIN = 0.000000 / -3212.082031 = -0.000000
 t = 0.035800
 Razón MAX/MIN = 175.981934 / -3212.082031 = -0.054787
 t = 0.053700
 Razón MAX/MIN = 175.981934 / -4854.220703 = -0.036253

t = 0.071600
 Razón MAX/MIN = 935.269470 / -4854.220703 = -0.192671
 t = 0.089500
 Razón MAX/MIN = 935.269470 / -5357.673828 = -0.174566
 t = 0.107400
 Razón MAX/MIN = 935.269470 / -5357.673828 = -0.174566
 t = 0.125300
 Razón MAX/MIN = 935.269470 / -5357.673828 = -0.174566
 t = 0.143200
 Razón MAX/MIN = 935.269470 / -5357.673828 = -0.174566

11. Reducción de la matriz K
 Tamaño: 100 x 100

12. Reducción de la matriz equivalente
 Tamaño: 100 x 100

12. Reducción del vector f(0)
 Tamaño: 100 x 1

13. Solución del caso est tico por el
 m, todo iterativo de Gauss-Seidel.

Matriz diagonalmente dominante...

**** PROCESO ITERATIVO EXITOSO ****

```

// Programa PRE-PROCESADOR para an lisis
// din mico de una viga en cantil:ver por
// el m, todo de elemento finito.
// Inicio: Junio 5 de 1998.
// A. DIRECTIVAS
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <dos.h>
#include <string.h>
#include <io.h>
#include <math.h>
#define PI 3.141592653
#define g 9.81
#define MAX_HOR 5 // Mximo número de elementos en dirección vertical
#define MAX_VER 30 // Mximo número de elementos en dirección horizontal
// B. PROTOTIPOS DE FUNCIONES
void presentat();
void geometria(float *l, float *a, float *b);
void propiedades(float *nu, float *E, float *densidad);
void coordenadas_globales(float *l, float *a, int *m,
                           int *n, int *tot_el, int *tot_nod);
void cond_de_frontera(int *tot_el, int *tot_nod, int *nod_res, int *el_carg);
void carga_dinamica(float *masa, float *base, float *F0, float *w);
void tipo_vibracion(float *l, float *a, float *b, float *base, float *E,
                    float *densidad, float *masa, float *w, float *wn);
int fuerza_discr(float *F0, float *w, float *wn, float *tau, float *E,
                 float *densidad, int *m, int *n, int *tot_el, int *tot_nod,
                 int *nod_res, int *el_carg,
                 int *puntos, float *tau, float *paso, float *masa);
char verificacion();
void lee_info_gen();
void lee_coord(int *tot_el);
void dibujo();
char correccion();
void salida();
// C. PROGRAMA PRINCIPAL
// Variables globales
FILE *reporte;
FILE *coord;
FILE *nodos;
FILE *elmcar;
FILE *fuerza;
FILE *clave;
char *ruta_reporte = "C:\\TESIS\\REPORTE.TXT";
char *ruta_coord = "C:\\TESIS\\COORDEN.TXT";
char *ruta_nodos = "C:\\TESIS\\NODRES.TXT";
char *ruta_elmcar = "C:\\TESIS\\ELMCAR.TXT";
char *ruta_fuerza = "C:\\TESIS\\FUERZA.TXT";
char *ruta_clave = "C:\\TESIS\\CLAVE.TXT";
void main()
{
// Variables locales a main()
char ver_cor;
int m, n, tot_el, tot_nod, nod_res, el_carg, puntos;
float l, a, b, nu, E, densidad, masa, base, F0, w, wn, tau, paso;
// Apertura de archivos de trabajo
reporte = fopen(ruta_reporte, "w"); // Reporte
coord = fopen(ruta_coord, "w"); // Coordenadas globales
nodos = fopen(ruta_nodos, "w"); // Nodos restringidos
elmcar = fopen(ruta_elmcar, "w"); // Elementos cargados
fuerza = fopen(ruta_fuerza, "w"); // Fuerza discretizada
clave = fopen(ruta_clave, "w"); // Datos clave del an lisis
// Llamadas a funciones
presentat();
do {
    geometria(&l, &a, &b);
    propiedades(&nu, &E, &densidad);
    coordenadas_globales(&l, &a, &m, &n, &tot_el, &tot_nod);
    cond_de_frontera(&tot_el, &tot_nod, &nod_res, &el_carg);
    fclose(coord); // Cierra archivo COORD.TXT
    fclose(nodos); // Cierra archivo NODRES.TXT
    fclose(elmcar); // Cierra archivo ELMCAR.TXT
    carga_dinamica(&masa, &base, &F0, &w);
    tipo_vibracion(&l, &a, &b, &base, &E, &densidad, &masa, &w, &wn);
    puntos = fuerza_discr(&F0, &w, &wn, &tau, &E, &paso);
    fclose(fuerza); // Cierra archivo FUERZA.TXT
    fclose(reporte); // Cierra archivo REPORTE.TXT
    claves(&l, &a, &b, &nu, &E, &densidad, &m, &n, &tot_el, &tot_nod,
           &nod_res, &el_carg, &puntos, &tau, &paso, &masa);
    fclose(clave); // Cierra archivo CLAVE.TXT
}
do {
    ver = verificacion();
    switch (ver) {
        case '1': lee_info_gen();
                break;
        case '2': lee_coord(&tot_el);
                break;
        case '3': dibujo();
                break;
        case '4': break;
    }
}
}

```



```

    fprintf(Reporte, "\n\n\n");
}
//*****
void cond_de_frontera(int *tot_el, int *tot_nod, int *nod_res, int *el_carg)
{
    int i;
    int etiqueta_nod[MAX_VERT], etiqueta_el[MAX_HORI];
    char op;

    // Escribe en pantalla
    do {
        clrscr();
        printf("4. Condiciones de frontera.\n\n");
        do {
            printf("a) Número de nodos restringidos: ");
            scanf("%i", &nod_res);
            } while ( (*nod_res < 2) || (*nod_res > MAX_VERT) );
            printf("\n\nNodo restringido Numeración global\n");
            for (i=0; i<*nod_res; i++) {
                do {
                    printf("%8i%20s", i+1, " ");
                    scanf("%i", &etiqueta_nod[i]);
                } while ( (etiqueta_nod[i] < 1) ||
                    (etiqueta_nod[i] > *tot_nod) );
                printf("\n\n");
            }
            do {
                printf("\n\nCorrección de datos (S/N): ");
                op = getch();
                } while ( (op != 's') && (op != 'S') &&
                    (op != 'n') && (op != 'N') );
            } while ( (op == 's') || (op == 'S') );
        }
    }
}

    printf("\n\n");
    do {
        printf("\n\nCorrección de datos (S/N): ");
        op = getch();
        } while ( (op != 's') && (op != 'S') &&
            (op != 'n') && (op != 'N') );
    } while ( (op == 's') || (op == 'S') );
}

// Escribe en archivo REPORTE.TXT
fprintf(Reporte, "4. Condiciones de frontera.\n");
fprintf(Reporte, " Nodos restringidos: %-2i Elementos cargados: %-2i\n",
    *nod_res, *el_carg);

fprintf(Reporte, "\n\nNodo Numeración global\n");
for (i=0; i<*nod_res; i++)
    fprintf(Reporte, "%3i%12i\n", i+1, etiqueta_nod[i]);

for (i=0; i<*el_carg; i++)
    fprintf(Reporte, "%3i%12i\n", i+1, etiqueta_el[i]);

// Escribe en archivos NODRES.TXT y ELMCAR.TXT
for (i=0; i<*nod_res; i++)
    fprintf(Reporte, "%5i", etiqueta_nod[i]);
}
fprintf(Reporte, "\n\n");
}
//*****
void carga_dinamica(float *masa, float *base, float *F0, float *w)
{
    char op;

    // Escribe en pantalla
    do {
        clrscr();
        printf("5. Naturaleza de la carga din mica.\n\n");
        printf("\n\nFunción de excitación F(t) = F0 sen wt\n\n");
        do {
            printf("Masa del motor (kg): ");
            scanf("%f", &masa);
            } while ( *masa <= 0.0);
            do {
                printf("Base del motor (m): ");
                scanf("%f", &base);
                } while ( *base <= 0.0);
    }
}

```

```

else printf("Resonancia:");
printf(" w/wn = %-1.2f\n", r);

// Escribe en archivo REPORTE.TXT
fprintf(reporte, " Rigidez equivalente = %-10.2f (N/m)\n", k);
fprintf(reporte, " Frecuencia natural = %-10.2f (Hz)\n", *wm);
if (r < 1) fprintf(reporte, " Respuesta en fase:");
else if (r > 1) fprintf(reporte, " Respuesta fuera de fase:");
else fprintf(reporte, " Resonancia:");
fprintf(reporte, " w/wn = %-1.2f\n", r);
}

//*****
int fuerza_discr(float *FO, float *w, float *wn, float *tau, float *paso)
{
    int i, n;
    float t = 0.0, Ft;
    printf("\n\n");
    do {
        printf("Puntos de discretización del tiempo\n");
        printf("m = {5, 9, 13, 17, 21, ..., 2k-1, 2k+3, ...}: ");
        scanf("%i", &n);
    } while ( ( n != 5) && ( n != 9) && ( n != 13) &&
              ( n != 17) && ( n != 21) );
    // C lculo del periodo de vibración y del paso
    *tau = 2*PI/( *wn);
    *paso = *tau/(n - 1);
    // Escribe en pantalla y archivos REPORTE.TXT y FUERZA.TXT
    printf("\n\nPeriodo de vibración = %-10.4f (seg)\n", *tau);
    printf("Paso de discretización del tiempo = %-10.4f (seg)\n", *paso);
    fprintf(reporte, " Puntos de discretización = %3i\n", n);
    fprintf(reporte, " Período de vibración = %-10.4f (seg)\n", *tau);
    fprintf(reporte, " Paso de discretización del tiempo = %-10.4f (seg)\n\n", *paso);
    fprintf(reporte, "paso t (seg) Ft (N)\n");
    for(i=0; i < n; i++){
        Ft = (*FO)*sin( (*w)*t);
        fprintf(reporte, "%3i%15.4f%15.4f\n", i+1, *paso*i, Ft);
        fprintf(fuerza, "%f\n", Ft);
        t += *paso;
    }
    fprintf(reporte, "\n\n");
    return (n);
}

//*****
void claves(float *t, float *a, float *b, float *nu, float *E,

```

```

do {
    printf("Amplitud (N): ");
    scanf("%f", FO);
    } while ( *FO <= 0.0);
do {
    do {
        printf("Velocidad del motor:\n");
        printf("w[1] RPM\n");
        printf("w[2] rad/seg\n");
        printf("\n\n"); Escoge su opción: ");
        op = getch();
    } while ( (op != '1') && (op != '2') );
    printf("\n\nVelocidad angular: ");
    scanf("%f", w);
    if (op == '1') *w == PI/30;
    } while ( *w <= 0.0);
do {
    printf("\n\nCorrección de datos (S/N): ");
    op = getch();
    } while ( (op != 's') && (op != 'S') &&
              (op != 'n') && (op != 'N') );
    } while ( (op == 's') || (op == 'S') );
}

// Escribe en archivo REPORTE.TXT
fprintf(reporte, "5. Naturaleza de la carga din mica.\n");
fprintf(reporte, " m = %-10.2f (kg)\n FO = %-10.2f (N)\n w = %-10.2f (rad/seg)\n",
        *masa, *FO, *w);
}

//*****
void tipo_vibracion(float *t, float *a, float *b, float *base, float *E,
                  float *densidad, float *masa, float *w, float *wn)
{
    float l, k, ll, r;
    l = (*b)*pow(*a, 3)/12;
    ll = *l - *base/2;
    k = 24*( *E)*l + (*densidad)*g*( *a)*(*b)*pow(ll, 2)*( pow(ll, 2) -
        4*ll*( *l) + 6*pow(*l, 2) );
    k = k/(8*pow(ll, 3) );
    *wn = sqrt(k/ *masa);
    r = *w / *wn;
    // Escribe en pantalla
    clrscr();
    printf("Rigidez equivalente = %-10.2f (N/m)\n", k);
    printf("Frecuencia natural = %-10.2f (Hz)\n", *wn);
    if (r < 1) printf("Respuesta en fase:");
    else if (r > 1) printf("Respuesta fuera de fase:");

```

```

float *densidad, int *n, int *n1, int *tot_el, int *tot_nod,
int *nod_res, int *el_carg,
int *puntos, float *tau, float *paso, float *masa)
{
    clrscr();
    printf("clave, %10.2f%10.2f%10.2f%10.2f%15.2f%10.2f\n",
        *l, *a, *b, *nu, *E, *densidad); // Geometria y propiedades
    printf("clave, %5f%5f%5f\n",
        *n, *n, *tot_el, *tot_nod); // Caracteristicas de la malla
    printf("clave, %5f%5f\n",
        *nod_res, *el_carg); // Condiciones de frontera
    printf("clave, %5f%10.4f%10.4f\n",
        *puntos, *tau, *paso, *masa); // Ciclo din mico
}

/*****
char verificacion()
{
    char op;
    do {
        clrscr();
        printf("4. Verificación de datos:\n\n\n\n\n");
        printf("\nOpciones para verificación: \n\n");
        printf("\n[1] Listado de información general \n");
        printf("\n[2] Listado de coordenadas globales \n");
        printf("\n[3] Graficación de la malla. \n");
        printf("\n[4] Salir de la verificación. \n\n");
        printf("\n\n\n\n\nElija su opción: ");
        scanf("%c", &op);
        } while ( (op != '1') && (op != '2') && (op != '3') && (op != '4') );
    return op;
}

/*****
void lee_info_gen()
{
    float l, a, b, nu, E, densidad, tau, paso;
    int n, n, tot_el, tot_nod, nod_res, el_carg, puntos;

    // Abre archivo CLAVE.TXT en modo lectura
    if ( (clave = fopen(ruta_clave, "r") ) == NULL)
    {
        clrscr();
        gotoxy(20,8);
        printf("Error al abrir el archivo CLAVE.TXT.");
        delay(8000);
        return;
    }

    fscanf(clave, "%f%f%f%f%f\n", &l, &a, &b, &nu, &E, &densidad);
    fscanf(clave, "%i%i%i%i\n", &n, &n, &tot_el, &tot_nod);
    fscanf(clave, "%i%i%i\n", &nod_res, &el_carg);
    fscanf(clave, "%i%i%f%f%f\n", &puntos, &tau, &paso);

    clrscr();
    printf("\n\n\n\n\nFORMACION GENERAL PARA EL ANALISIS.\n\n\n");
    printf("1. Dimensiones generales de la viga (metros) \n");
    printf(" L = %10.2f a = %10.2f b = %10.2f\n", l, a, b);
    printf("2. Propiedades del material (nu, E, densidad) \n");
    printf(" nu = %10.2f E = %15.2f (N/m2) densidad = %10.2f (kg/m3) \n", nu, E,
        densidad);
    printf("3. Generación de coordenadas nodales globales \n");
    printf(" Densidad de la malla: %i x %i \n", m, n);
    printf(" %-5i elementos, %5i nodos \n", tot_el, tot_nod);

    printf("4. Condiciones de frontera \n");
    printf(" Nodos restringidos: %-2i Elementos cargados: %-2i \n",
        nod_res, el_carg);

    printf("5. Ciclo din mico \n");
    printf(" Puntos de discretización = %-2i \n", puntos);
    printf(" Periodo de vibración = %10.4f (seg) paso = %10.4f (seg) \n", tau, paso);

    printf("\n\n\n\n\nEl listado ha terminado. Presiona ");
    printf("\n\n\n\n\nCualquier tecla para continuar...");
    getch();
}

/*****
void lee_coord(int *tot_el)
{
    int i,j;
    float x1,y1,x2,y2,x3,y3;

    clrscr();
    // Abre archivo COORD.TXT para verificar las coordenadas nodales globales
    if ( (coord = fopen(ruta_coord, "r") ) == NULL)
    {
        gotoxy(20,8);
        printf("Error al abrir el archivo COORD.TXT.");
        delay(8000);
        return;
    }

    // Encabezado de la salida
    printf("%-10s%-8s%-10s%-10s%-10s%-10s%-10s%-10s\n",

```



```

void asocia_nodos(int *n, int *tot_el);
void derivs(int *tot_el, float *b);
void matriz_K(float *nu, float *E, int *tot_el, int *tot_nod);
void matriz_M(float *densidad, int *tot_el, int *tot_nod);
void matriz_equiv(float *paso, int *tot_nod);
void vector_Fi(int *puntos, int *el_carg, int *tot_nod,
              int *tot_el, float *masa, float *densidad, float *b,
              float *paso);
void vector_v0(int *tot_nod);
void reduce_mat_K(int *tot_nod, int *nod_res);
void reduce_mat_eq(int *tot_nod, int *nod_res);
void reduce_vec_f0(int *tot_nod, int *nod_res);
void reduce_vec_v0(int *tot_nod, int *nod_res);
// Solución del caso est tico
// Solución del caso est tico
void Gauss_Seidel_est(int *tot_nod, int *nod_res);
// Solución del caso din mico
void vector_vf0();
void reduce_vec_fleg(int *tot_nod, int *nod_res);
void Gauss_Seidel_dyn(int *tot_nod, int *nod_res);

// C. PROGRAMA PRINCIPAL
void main()
{
// Variables locales a main()
int m, n, tot_el, tot_nod, nod_res, el_carg, puntos;
float l, a, b, nu, E, densidad, tau, paso, masa;

reporte = fopen(ruta_reporte, "a"); // Abre archivo de reporte para
PROCESADOR // agregar información del
// Lectura de datos
lee_claves(&l, &a, &b, &nu, &E, &densidad, &m, &n, &tot_el, &tot_nod,
          &nod_res, &el_carg, &puntos, &tau, &paso, &masa);

// C leulos iniciales
asocia_nodos(&n, &tot_el);
derivs(&tot_el, &b);
matriz_K(&nu, &E, &tot_el, &tot_nod);
matriz_M(&densidad, &tot_el, &tot_nod);
matriz_equiv(&paso, &tot_nod);
vector_Fi(&puntos, &el_carg, &tot_nod, &tot_el, &masa,
          &densidad, &b, &paso);
vector_v0(&tot_nod);
// Reducción de matrices por restricciones nodales
reduce_mat_K(&tot_nod, &nod_res);
reduce_mat_eq(&tot_nod, &nod_res);
reduce_vec_f0(&tot_nod, &nod_res);
reduce_vec_v0(&tot_nod, &nod_res);

reduce_vec_v0(&tot_nod, &nod_res);

// Solución del caso est tico
Gauss_Seidel_est(&tot_nod, &nod_res);
// Solución del caso din mico
vector_vf0();
reduce_vec_fleg(&tot_nod, &nod_res);
Gauss_Seidel_dyn(&tot_nod, &nod_res);
} // Termina programa principal

// D. DEFINICIAN DE FUNCIONES
/*****
void lee_claves(float *l, float *a, float *b, float *nu, float *E,
              float *densidad, int *m, int *n, int *tot_el, int *tot_nod,
              int *nod_res, int *el_carg,
              int *puntos, float *tau, float *paso, float *masa)
{
// Abre archivo de datos clave del an lisis
clave = fopen(ruta_clave, "r");
fscanf(clave, "%f%f%f%f%f%f\n", l, a, b, nu, E, densidad);
fscanf(clave, "%i%i%i%i\n", m, n, tot_el, tot_nod);
fscanf(clave, "%i%i%i\n", nod_res, el_carg);
fscanf(clave, "%i%i%f%i%f\n", puntos, tau, paso, masa);
// Cierra archivo
fclose(clave);
}
/*****
void asocia_nodos(int *n, int *tot_el)
{
int k, fila, col, nodo_1, nodo_2, nodo_3;

// Abre archivo para nodos globales
nodos = fopen(ruta_nodos, "w");

// C leulo de la fila, columna y nodos globales para el k-simo elemento
for( k = 1; k <= *tot_el; k++) {
// k no es múltiplo de 2n
if( div(k, 2*(*n))%2 == 0 ) {
// fila y columna
col = div(k, 2*(*n))%quot + 1;
fila = int(0.5 + 0.5*(k - (col - 1)*2*(*n)));
// nodos globales
}
}
}
}

```

```

if ( div(k, 2).rem != 0 ) { // k es impar
    nodo_1 = (col - 1)*( *n + 1) + fila;
    nodo_2 = nodo_1 + ( *n + 1);
    nodo_3 = nodo_1 + 1;
}
if ( div(k, 2).rem == 0 ) { // k es par
    nodo_1 = (col - 1)*( *n + 1) + fila;
    nodo_2 = nodo_1 + ( *n + 1);
    nodo_3 = nodo_1 - 1;
}
// k es múltiplo de 2n (siempre par)
if ( div(k, 2)*( *n).rem == 0 ) {
    // fila y columna
    col = div(k, 2)*( *n).quot;
    fila = *n;
    // nodos globales
    nodo_1 = (col + 1)*( *n + 1);
    nodo_2 = nodo_1 - ( *n + 1);
    nodo_3 = nodo_1 - 1;
}
// Guarda en archivo NODOS.TXT
fprintf(nodos, "%5d%5d%5d\n", nodo_1, nodo_2, nodo_3);
}
for( k = 1...

// Cierra archivo
fclose(nodos);
}
//*****
void derivs(int *tot_el, float *b)
{
    int k;
    float x1, y1, x2, y2, x3, y3;
    float A, V, Dphi1x, Dphi1y, Dphi2x, Dphi2y, Dphi3x, Dphi3y;
    // Abre archivos de coordenadas globales y derivadas parciales
    coord = fopen(ruta_coord, "r");
    deriv = fopen(ruta_deriv, "w");
    // Calcula derivadas parciales de funciones globales
    // de interpolación para el k-ésimo elemento
    for (k = 1; k <= *tot_el; k++) {
        // Lee coordenadas globales
        fscanf(coord, "%f%f%f%f%f\n", &x1, &y1, &x2, &y2, &x3, &y3);
        A = x2*y1 + x3*y2 + x1*y3 - x3*y1 - x1*y2 - x2*y3;
        V = fabs( A/2*( *b) ); // Volumen de elemento finito
        //Calcula derivadas parciales
        Dphi1x = (y3 - y2)/A;
        Dphi1y = (x2 - x3)/A;
        Dphi2x = (y1 - y3)/A;
        Dphi2y = (x3 - x1)/A;
        Dphi3x = (y2 - y1)/A;
        Dphi3y = (x1 - x2)/A;
        // Escribe en archivo DERIVS.TXT
        fprintf(deriv, "%f %f %f %f %f %f\n",
            Dphi1x, Dphi1y, Dphi2x, Dphi2y, Dphi3x, Dphi3y, V);
    }
    // for (k = 1...

// Cierra archivos
fclose(coord);
fclose(deriv);
}
//*****
void matriz_K(float *nu, float *E, int *tot_el, int *tot_nod)
{
    int i, j, k, nodo_1, nodo_2, nodo_3, sim = 0;
    float Dphi1x, Dphi1y, Dphi2x, Dphi2y, Dphi3x, Dphi3y, V, K[MAXXGL][MAXXGL];
    float Dphiix, Dphiiy, Dphijx, Dphijy, mu, lambda;
    float max = 1.0e-30, min = 1.0e30;
    // Calcula coeficientes de Lam,
    mu = 0.5*(E)/(1 + *nu);
    lambda = *nu*(E)/(1 + *nu)*(1 - 2*( *nu) );
    //Inicializa elementos de la matriz de rigidez
    for (i = 0; i < 2*( *tot_nod); i++)
        for (j = 0; j < 2*( *tot_nod); j++)
            K[i][j] = 0.0;
    // C leulo de la matriz de rigidez
    clrscr();
    printf("6. Construcción de la matriz K\n");
    printf(" Tamapo: %i x %i\n", 2*( *tot_nod), 2*( *tot_nod) );
    printf("\n\nConstruyendo...\n");
    fprintf(deriv, "6. Construcción de la matriz K\n");
    fprintf(deriv, " Tamapo: %i x %i\n", 2*( *tot_nod), 2*( *tot_nod) );
    // Abre archivos de nodos globales y derivadas parciales
    nodos = fopen(ruta_nodos, "r");
    deriv = fopen(ruta_deriv, "r");
    // An lisis de cada elemento finito
    for (k = 1; k <= *tot_el; k++) {
        // Lee nodos globales y derivadas parciales de las
        // funciones de interpolación para el k-ésimo elemento

```

```

// Calcula ji-simo elemento de la matriz K
for (i = 0; i < *tot_nod; i++) {
    for (j = 0; j < *tot_nod; j++) {

        // Relaciona nodos globales, funciones de interpolación
        // y elementos de la matriz K
        if ( (nodo_1 == (i+1)) && (nodo_2 == (j+1)) ) {
            Dphix = Dphi1x;
            Dphiy = Dphi1y;
            Dphix = Dphi2x;
            Dphiy = Dphi2y;
        }

        else if ( (nodo_1 == (i+1)) && (nodo_3 == (j+1)) ) {
            Dphix = Dphi1x;
            Dphiy = Dphi1y;
            Dphix = Dphi3x;
            Dphiy = Dphi3y;
        }

        else if ( (nodo_2 == (i+1)) && (nodo_3 == (j+1)) ) {
            Dphix = Dphi2x;
            Dphiy = Dphi2y;
            Dphix = Dphi3x;
            Dphiy = Dphi3y;
        }

        else if ( (nodo_2 == (i+1)) && (nodo_1 == (j+1)) ) {
            Dphix = Dphi2x;
            Dphiy = Dphi2y;
            Dphix = Dphi1x;
            Dphiy = Dphi1y;
        }

        else if ( (nodo_3 == (i+1)) && (nodo_1 == (j+1)) ) {
            Dphix = Dphi3x;
            Dphiy = Dphi3y;
            Dphix = Dphi1x;
            Dphiy = Dphi1y;
        }

        else if ( (nodo_3 == (i+1)) && (nodo_2 == (j+1)) ) {
            Dphix = Dphi3x;
            Dphiy = Dphi3y;
            Dphix = Dphi2x;
            Dphiy = Dphi2y;
        }

        else if ( (nodo_1 == (i+1)) && (nodo_1 == (j+1)) ) {
            Dphix = Dphi1x;
            Dphiy = Dphi1y;
            Dphix = Dphi1x;
            Dphiy = Dphi1y;
        }

        else if ( (nodo_2 == (i+1)) && (nodo_2 == (j+1)) ) {
            Dphix = Dphi2x;
            Dphiy = Dphi2y;
            Dphix = Dphi2x;
            Dphiy = Dphi2y;
        }

        else if ( (nodo_3 == (i+1)) && (nodo_3 == (j+1)) ) {
            Dphix = Dphi3x;
            Dphiy = Dphi3y;
            Dphix = Dphi3x;
            Dphiy = Dphi3y;
        }

        else {
            Dphix = 0.0;
            Dphiy = 0.0;
            Dphix = 0.0;
            Dphiy = 0.0;
        }

        K[2*i+1][2*j+1] += ((2*mu + lambda)*Dphix*Dphix + mu*Dphiy*Dphiy)*V;
        K[2*i+1][2*j+1] += ((2*mu + lambda)*Dphiy*Dphiy + mu*Dphix*Dphix
            )*V;
        }
    } // for (k = 1...
}

// Cierra archivos
fclose(nodos);
fclose(deriv);

// Guarda matriz en archivo matrizK
matrizK = fopen(ruta_matrizK, "w");
for (i = 0; i < 2*(*tot_nod); i++) {
    for (j = 0; j < 2*(*tot_nod); j++) {
        fprintf(matrizK, "%69.2f", K[i][j]/100000);
        fprintf(matrizK, "v\n");
    }
}
fclose(matrizK); // Cierra archivo

// Guarda matriz en archivo matK
matK = fopen(ruta_matK, "w"); // Abre archivo
for (i = 0; i < 2*(*tot_nod); i++)

```

```

for (j = 0; j < 2*(tot_nod); j++)
    fprintf(matK, "%f\n", K[i][j]);
fclose(matK); // Cierra archivo

// Busca m ximo y minimo elemento de la matriz K
for (i = 0; i < 2*(tot_nod); i++)
    for (j = 0; j < 2*(tot_nod); j++) {
        k = K[i][j];
        if ((K[i][j] < min) && (K[i][j] > 0.0)) min = K[i][j];
    }

// Verifica simetría de la matriz k
for (i = 0; i < 2*(tot_nod); i++)
    for (j = i+1; j < 2*(tot_nod); j++)
        if (K[i][j] != K[j][i]) sim = 1;

// Reporte de propiedades de la matriz K
printf(" Razón MAX/MIN = %f / %f = %f\n", max, min, max/min);
fprintf(reporte, " Razón MAX/MIN = %f / %f = %f\n", max, min, max/min);
if (sim == 0) {
    printf(" MATRIZ SIMETRICA");
    fprintf(reporte, " MATRIZ SIMETRICA");
}
else {
    printf(" MATRIZ NO SIMETRICA");
    fprintf(reporte, " MATRIZ NO SIMETRICA");
}
printf("\n\n");
fprintf(reporte, "\n\n");
}

// *****
void matrix_M(float *densidad, int *tot_el, int *tot_nod)
{
    int i, j, k, nodo_1, nodo_2, nodo_3, tipo, sim = 0;
    float x1, y1, x2, y2, x3, y3, Ar, a1, a2, a3, b1, b2, b3, c1, c2, c3, m;
    float A, B, C, D, E, F, M[MAXXGL][MAXXGL];
    float max = 1.0e-30, min = 1.0e30;

    // Inicializa elementos de la matriz M
    for (i = 0; i < 2*(tot_nod); i++)
        for (j = 0; j < 2*(tot_nod); j++)
            M[i][j] = 0.0;

    // C leulo de la matriz M
    printf("7. Construcción de la matriz M\n");
    printf(" Tamaño: %i x %i\n", 2*(tot_nod), 2*(tot_nod));
    printf("MatConstruyendo...\n");
    fprintf(reporte, "7. Construcción de la matriz M\n");

    fprintf(reporte, " Tamaño: %i x %i\n", 2*(tot_nod), 2*(tot_nod));
    coord = fopen(ruta_coord, "r");
    // Abre archivo de nodos y coordenadas globales
    nodos = fopen(ruta_nodos, "r");
    // An lisis de cada elemento finito
    for (k = 1; k <= *tot_el, k++) {
        // Lee nodos y coordenadas globales para el k-simo elemento
        fscanf(coord, "%i,%f,%f,%f\n", &x1, &y1, &x2, &y2, &x3, &y3);
        fscanf(nodos, "%i,%f,%f\n", &nodo_1, &nodo_2, &nodo_3);

        // Calcula ij-simo elemento de la matriz M
        for (i = 0; i < *tot_nod; i++) {
            for (j = 0; j < *tot_nod; j++) {
                // Calcula coeficientes de funciones de interpolación
                Ar = x2*y1 + x3*y2 + x1*y3 - x3*y1 - x1*y2 - x2*y3;
                a1 = (y3 - y2)/Ar;
                b1 = (x2 - x3)/Ar;
                c1 = (x2*y2 - x2*y3)/Ar;
                a2 = (y1 - y3)/Ar;
                b2 = (x3 - x1)/Ar;
                c2 = (x3*y3 - x3*y1)/Ar;
                a3 = (y2 - y3)/Ar;
                b3 = (x2 - x2)/Ar;
                c3 = (x2*y3 - x3*y2)/Ar;
                m = (y3 - y2)/(x3 - x2);

                // Identifica orientación de elemento finito
                if (y3 > y1) tipo = 1;
                else tipo = 2;

                // Relaciona nodos globales, funciones de interpolación
                // y elementos de la matriz M
                if (((nodo_1 == (i+1)) && (nodo_2 == (j+1))) ||
                    ((nodo_2 == (i+1)) && (nodo_1 == (j+1)))) {
                    A = a1*a2;
                    B = b1*b2;
                    C = a1*b2 + b1*a2;
                    D = a1*c2 + c1*a2;
                    E = b1*c2 + c1*b2;
                    F = c1*c2;
                }
                else if (((nodo_1 == (i+1)) && (nodo_3 == (j+1))) ||
                    ((nodo_3 == (i+1)) && (nodo_1 == (j+1)))) {
                    A = a1*a3;
                    B = b1*b3;
                    C = a1*b3 + b1*a3;
                }
            }
        }
    }
}

```

```

D = a1*c3 + c1*a3;
E = b1*c3 + c1*b3;
F = c1*c3;
}

else if( (( (modo_2 == (i+1)) && (modo_3 == (j+1))) ||
  ( (modo_3 == (i+1)) && (modo_2 == (j+1))) ) ) {
  A = a2*a3;
  B = b2*b3;
  C = a2*b3 + b2*a3;
  D = a2*c3 + c2*a3;
  E = b2*c3 + c2*b3;
  F = c2*c3;
}

else if( (modo_1 == (i+1)) && (modo_1 == (j+1)) ) {
  A = a1*a1;
  B = b1*b1;
  C = 2*a1*b1;
  D = 2*a1*c1;
  E = 2*b1*c1;
  F = c1*c1;
}

else if( (modo_2 == (i+1)) && (modo_2 == (j+1)) ) {
  A = a2*a2;
  B = b2*b2;
  C = 2*a2*b2;
  D = 2*a2*c2;
  E = 2*b2*c2;
  F = c2*c2;
}

else if( (modo_3 == (i+1)) && (modo_3 == (j+1)) ) {
  A = a3*a3;
  B = b3*b3;
  C = 2*a3*b3;
  D = 2*a3*c3;
  E = 2*b3*c3;
  F = c3*c3;
}

}

else {
  A = 0.0;
  B = 0.0;
  C = 0.0;
  D = 0.0;
  E = 0.0;
  F = 0.0;
}

}

if (tipo == 1) {
  M[2][2*j] +=
    (*densidad)*m*(x2 - x1)*(x2 - x1)/24*(
    2*A*(3*x1*x1 - 2*x1*x2 - x2*x2) -
    2*B*(m*m*x1*x1 - 2*m*m*x1*x2 +
    m*m*x2*x2 + 4*m*m*x1*y2 - 4*m*m*x2*y2 + 6*y2*y2) -
    8*x1*y2 + 4*x2*y2) -
    4*D*(2*x1 + x2) -
    4*E*(m*x1 - m*x2 + 3*y2) -
    12*F
  );
  /*
  (*densidad)*m*(x2 - x1)*(x2 - x1)/24*(
  2*A*(3*x1 + x2)*(x1 - x2) -
  2*B*(m*m*(x1 - x2)*(x1 - x2) + 4*m*y2*(x1
  C*(m*(3*x1 + x2)*(x1 - x2) + 4*y2*(2*x1 +
  -x2) + 6*y2*y2) -
  8*x2)*x2) -
  4*D*(2*x1 + x2) -
  4*E*(m*(x1 - x2) + 3*y2) -
  12*F);
  */
}

}

else {
  M[2][2*j] +=
    (*densidad)*(x2 - x3)/24*(
    A*(6*m*m*x2*x2*x2 - 2*m*x2*x2*x3 -
    2*m*x2*x3*x3 - 2*m*x3*x3*x3 - 8*x2*y1*y1 +
    8*x2*x3*y3 + 8*x3*x3*y3) +
    B*(2*m*m*m*x2*x2*x2 -
    2*m*m*m*x3*x3*x3 - 8*y1*y1*y1 +
    8*m*m*x2*x2*y3 - 16*m*m*m*x2*x3*y3 + 8*m*m*x3*x3*y3 +
    12*m*x3*y3*y3 + 8*y3*y3*y3) +
    C*(3*m*m*m*x2*x2*x2 - 5*m*m*m*x2*x2*x3 +
    m*m*x2*x3*x3 + m*m*x3*x3*x3 - 6*x2*y1*y1 - 6*x3*y3*y3 -
    4*m*x2*x3*y3 - 4*m*x3*x3*y3 + 6*x2*y3*y3 + 6*x3*y3*y3) +
    D*(8*m*x2*x2 - 4*m*x2*x3 - 4*m*x3*x3 -
    12*x2*y1 - 12*x3*y1 + 12*x2*y3 + 12*x3*y3) +
    E*(4*m*m*x2*x2 - 8*m*m*x2*x3 +
    4*m*m*x3*x3 - 12*y1*y1 + 12*m*x2*y3 - 12*m*x3*y3) +
    F*(12*m*x2 - 12*m*x3 - 24*y1 + 24*y3)
  );
  /*
  (*densidad)*(x2 - x3)/24*(
  2*A*(m*x2*(2*x2 + x3)*(x2 - x3) +
  4*(x3 - y1)*(x2*x2 + x2*x3 + x3*x3)
  ) +
  m*(x2*x2*x2 - x3*x3*x3) +
  ) +

```

```

3*m*(2*y3*y3 - m*m*x2*x3)*(x2 - x3) +
4*(y3*y3*y3 - y1*y1*y1) +
m*x3*(2*x2 + x3)*(x2 - x3) +
x2*x2*x2 +
x3 + 3*(y3*y3 - y1*y1) +
} // for (i = 0...
} // for (j = 0...
} // for (k = 1...
// Cierra archivos

// Guarda matriz en archivo matrizM
matrizM = fopen(ruta_matrizM,"w");
for (i = 0; i < 2*(tot_nod); i++) {
for (j = 0; j < 2*(tot_nod); j++)
fprintf(matrizM, "%9.2f", M[i][j]);
}
fclose(matrizM);

// Cierra archivo

// Guarda matriz en archivo matM
matM = fopen(ruta_matM,"w"); // Abre archivo
for (i = 0; i < 2*(tot_nod); i++)
for (j = 0; j < 2*(tot_nod); j++)
fprintf(matM, "%f\n", 2*Mij/(paso) + *paso*Kij/2);
}

// Busca m_ximo y m_inimo elemento de la matriz M
for (i = 0; i < 2*(tot_nod); i++)
for (j = 0; j < 2*(tot_nod); j++) {
if (M[i][j] > max) max = M[i][j];
if ((M[i][j] < min) && (M[i][j] > 0.0)) min = M[i][j];
}

// Verifica simetria de la matriz M
for (i = 0; i < 2*(tot_nod); i++)
for (j = i+1; j < 2*(tot_nod); j++)
if (M[i][j] != M[j][i]) sim = 1;

// Reporte de propiedades de la matriz M

2*B*(m*m*m*(x2*x2*x2 - x3*x3*x3) +
4*m*m*y3*(x2 - x3)*(x2 - x3) +
C*(3*m*m*x2*x2*(x2 - x3) + m*(4*y3 -
6*(y3*y3 - y1*y1)*(x2 + x3)) +
4*D*(m*(2*x2 + x3)*(x2 - x3) + 3*(y3 -
1 + m*m*(x2 - x3)*(x2 - x3) + 3*m*y3*(x2
12)* (m*(x2 - x3) + 2*(x2 - x3)));
*/

printff(" Razón MAX/MIN = %f / %f = %f\n", max, min, max/min);
fprintf(reporte, " Razón MAX/MIN = %f / %f = %f\n", max, min, max/min);
if (sim == 0) {
printff(" MATRIZ SIMETRICA");
fprintf(reporte, " MATRIZ SIMETRICA");
}
else {
printff(" MATRIZ NO SIMETRICA");
fprintf(reporte, " MATRIZ NO SIMETRICA");
}
printff("\n\n");
fprintf(reporte, "\n\n");
}

//*****
void matriz_equiv(float *paso, int *tot_nod)
{
float Kij, Mij;
int i, j;

// C leulo de la matriz equivalente
printff("9. Construcción de la matriz equivalente\n");
fprintf(reporte, "9. Construcción de la matriz equivalente\n");

// Abre archivo para leer matrices K y M y guardar matriz equivalente
matK = fopen(ruta_matK, "r");
matM = fopen(ruta_matM, "r");
mateq = fopen(ruta_mateq, "w");

// C leulo de matriz equivalente
for (i = 0; i < 2*(tot_nod); i++)
for (j = 0; j < 2*(tot_nod); j++) {
fscanf(matK, "%f", &Kij);
fscanf(matM, "%f", &Mij);
fprintf(mateq, "%f\n", 2*Mij/(paso) + *paso*Kij/2);
}

// Cierra archivos
fclose(mateq);
fclose(matK);
fclose(matM);
}

//*****
void vector_Ft(int *puntos, int *el_carg, int *tot_nod,
int *tot_el, float *masa, float *densidad, float *b,
float *paso)

```

```

    {
        int i, j, k, t, nodo_1, nodo_2, nodo_3, cargado, tipo, etiqueta, el[MAX_VERT];
        float x1, y1, x2, y2, x3, y3, A, ai, bi, ci, m, Ft, sj, bj, fl[MAXGL];
        float max = 1.0e-30, min = 1.0e30;

        // Abre archivo para guardar vector f(t)
        vector<float> v("vector.txt", "w");

        // Lee numeración global de elementos cargados
        elncar = fopen(ruta_elncar, "r");
        for (j = 0; j < *el_carg; j++)
            fscanf(elncar, "%i", &etiqueta_el[j]);
        fclose(elncar);

        // Reporte
        printf("8. Construcción del vector f(t)\n");
        printf("   Tamaño: %i x %i\n", 2*(tot_mod));
        printf("\n\nConstruyendo...\n");
        fprintf(Reporte, "8. Construcción del vector f(t)\n");
        fprintf(Reporte, "   Tamaño: %i x %i\n", 2*(tot_mod));

        // Abre archivo de fuerza discretizada
        fuerza = fopen(ruta_fuerza, "r");

        // Ciclo principal. Calcula vector f(t) para cada paso del tiempo
        for (t = 0; t < *puntos; t++) {

            // Lee fuerza discretizada
            fscanf(fuerza, "%f\n", &Ft);

            // Inicializa elementos del vector f(t)
            for (j = 0; j < 2*(tot_mod); j++)
                fl[j] = 0.0;

            // Ciclo del vector f(t)
            printf("   t = %f\n", *paso*t);
            fprintf(Reporte, "   t = %f\n", *paso*t);

            // Abre archivos de coordenadas globales y nodos globales
            nodos = fopen(ruta_nodos, "r");
            coord = fopen(ruta_coord, "r");

            // Análisis de cada elemento finito
            for (k = 1; k <= *tot_el; k++) {

                // Lee nodos y coordenadas globales para el k-ésimo elemento
                fscanf(coord, "%f%f%f%f%f%f\n", &x1, &y1, &x2, &y2, &x3, &y3);
                fscanf(nodos, "%f%f%f%f\n", &nodo_1, &nodo_2, &nodo_3);

                // Calcula i-ésimo elemento del vector f(t)
                // Identifica orientación de elemento finito
                if (y3 > y1) tipo = 1;
                else if (y3 > y2) tipo = 2;
                else tipo = 3;

                // Identifica elemento cargado
                for (j = 0; j < *el_carg; j++)
                    if (k == etiqueta_el[j]) cargado = 1;
                else cargado = 0;
                if (cargado == 1) break;
            }

            // Calcula coeficientes de funciones de interpolación
            // Relaciona nodos globales, funciones de interpolación
            // y elementos del vector f(t)
            A = x2*y1 + x3*y2 + x1*y3 - x3*y1 - x1*y2 - x2*y3;
            m = (y3 - y2)/(x3 - x2);
            if (nodo_1 == (t+1)) {
                ai = (y3 - y2)/A;
                bi = (x2 - x3)/A;
                ci = (x3*y2 - x2*y3)/A;
            }
            else if (nodo_2 == (t+1)) {
                ai = (y1 - y3)/A;
                bi = (x3 - x1)/A;
                ci = (x1*y3 - x3*y1)/A;
            }
            else if (nodo_3 == (t+1)) {
                ai = (y2 - y1)/A;
                bi = (x1 - x2)/A;
                ci = (x2*y1 - x1*y2)/A;
            }
            else {
                ai = 0.0;
                bi = 0.0;
                ci = 0.0;
            }

            if (tipo == 1) {
                bj = -m*(x2 - x1)*(x2 - x1)/6*(
                    ai*(2*x1 + x2) + bi*(m*x1 - m*x2 + 3*y2) + 3*ci
                )*(-densidad*g/*b);
                fl[2*t+1] += bj;
            }
            else {
                bj =
                    (x2 - x3)/6*(

```

```

3*x2*y3 + 3*x3*y3 ) +
ai*( 2*m*x2*x2 - m*x2*x3 - m*x2*x3 - 3*x2*y1 - 3*x3*y1 +
bi*( m*m*x2*x2 - 2*m*m*x2*x3 + m*m*x3*x3 - 3*y1*y1 +
3*m*x2*y3 - 3*m*x3*y3 + 3*y3*y3 ) +
ci*( 3*m*x2 - 3*m*x3 - 6*y1 + 6*y3 )
/*
(x2 - x3)/6*(
ai*( m*m*x2*x2 + x3)*(x2 - x3) + 3*(x2 + x3)*(y1 - y1) +
bi*( m*m*x2*x2 - x3 - x3 - 3*m*y3*(x2 - x3) + 3*(y3 - y3 -
y1*y1) +
ci*( 3*m*(x2 - x3) + 6*(y3 - y1) )
)*(- *densidad*g/*b);
*/
sj = ( ai*(x3*x3*x3 - x2*x2*x2)/3 + (bi*y2 + ci)*(x3*x3 - x2*x2)/2
)*(- *masa*g - Fy)/( *el_carg*( *b))*cargado;
f[2*i+1] += bj + sj;
}
} // for (i = 0...
} // for (k = 1...
// Cierra archivos

// Guarda vector en archivo vectorFt
fprintf(vectorFt, "%f\n", *paso*t);
for (i = 0; i < 2*( *tot_nod); i++)
    fprintf(vectorFt, "%15.4f\n", f[i]);

// Busca m_ximo y m_jnimo elemento del vector f(t)
for (i = 0; i < 2*( *tot_nod); i++) {
    if ( f[i] > max ) max = f[i];
    if ( f[i] < min ) min = f[i];
}

// Reporte de propiedades del vector f(t)
printf(" Razón MAX/MIN = %f / %f = %f\n", max, min, max/min);
fprintf(reporte, " Razón MAX/MIN = %f / %f = %f\n", max, min, max/min);
} // for (t = 0...

printf("\n\n");
fprintf(reporte, "\n\n");
fclose(fuerza);
fclose(vectorFt);
}

//*****
void vector_v0(int *tot_nod);
{
int j;
float t, v0;

// Genera vector v(0)
vectorv0 = fopen(ruta_vectorv0, "w");
t = 0.0;
fprintf(vectorv0, "%f\n", t);
for (j = 0; j < 2*( *tot_nod); j++) {
    v0 = 0.0;
    fprintf(vectorv0, "%f\n", v0);
}
fclose(vectorv0);
}

//*****
void reduce_mat_K(int *tot_nod, int *nod_res)
{
float K[MAXGLJ][MAXGLJ];
int i, j, k, nres, etiqueta_nod[MAX_HOR];

// Reducción de la matriz K
printf("10. Reducción de la matriz K\n");
printf("\nReduciendo...\n");
fprintf(reporte, "10. Reducción de la matriz K\n");

// Lee matriz K
matK = fopen(ruta_matK, "r"); // Abre archivo
for (i = 0; i < 2*( *tot_nod); i++)
    for (j = 0; j < 2*( *tot_nod); j++)
        fscanf(matK, "%f\n", &K[i][j]);
fclose(matK); // Cierra archivo

// Lee numeración global de nodos restringidos
nodres = fopen(ruta_nodres, "r");
for (k = 1; k <= *nod_res; k++)
    fscanf(nodres, "%i", &etiqueta_nod[k]);
fclose(nodres);

// Reducción de la matriz K
for (k = 1; k <= *nod_res; k++) {
    // Empareja subíndice de matriz con numeración de nodo restringido
    nres = etiqueta_nod[k] - k;

    // Reducción de columnas
    for (i = 0; i < *tot_nod - k + 1; i++)
        for (j = 0; j < *tot_nod - k; j++)
            if (j >= nres) {
                K[2*i][2*j] = K[2*i][2*j+2];
            }
        }
    }
}

```



```

printf("\n\nReduciendo...\n");
fprintf(Reporte, "11. Reducción del vector f(0)\n");

// Lee vector f(0)
vectorFt = fopen(ruta_vectorFt, "r");
fscanf(vectorFt, "%f\n", &t);
for (i = 0; i < 2*(tot_nod); i++)
    fscanf(vectorFt, "%f\n", &f[i]);
fclose(vectorFt);

// Lee numeración global de nodos restringidos
nodos = fopen(ruta_nodos, "r");
for (k = 1; k <= tot_nod; k++)
    fscanf(nodos, "%i", &etiqueta_nod[k]);
fclose(nodos);

// Reducción del vector f(0)
for (k = 1; k <= tot_nod; k++) {
    // Empareja subíndice de vector con numeración de nodo restringido
    nres = etiqueta_nod[k] - k;

    // Reducción de renglones
    for (i = 0; i < tot_nod - k; i++)
        if (i >= nres)
            v0[2*i] = v0[2*i+2];
    }

// Guarda vector f(0) reducido
vector0r = fopen(ruta_vector0r, "w");
for (i = 0; i < 2*(tot_nod - tot_nres); i++)
    fprintf(vector0r, "%f\n", v0[i]);
fclose(vector0r);

}

//*****

void Gauss_Seidel_est(int *tot_nod, int *nod_res)
{
    float A[MAXXGL][MAXXGL], b[MAXXGL], x[MAXXGL];
    float x_ant, temp, sum, error, i = 0.0;
    int i, j, k, l, tol, max_iter, num_iter = 0,
        tol_excedida = TRUE, diagonal = FALSE, n;

    // Tamaño de las matrices
    n = 2*(tot_nod - tot_nres);

    // Lee datos
    matKr = fopen(ruta_matKr, "r");
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            fscanf(matKr, "%f\n", &A[i][j]);
    fclose(matKr);

    vector0 = fopen(ruta_vector0, "r");
    for (i = 0; i < n; i++)
}

```

```

printf("\n\nReduciendo...\n");
fprintf(Reporte, "11. Reducción del vector f(0)\n");

// Lee vector f(0)
vectorFt = fopen(ruta_vectorFt, "r");
fscanf(vectorFt, "%f\n", &t);
for (i = 0; i < 2*(tot_nod); i++)
    fscanf(vectorFt, "%f\n", &f[i]);
fclose(vectorFt);

// Lee numeración global de nodos restringidos
nodos = fopen(ruta_nodos, "r");
for (k = 1; k <= tot_nod; k++)
    fscanf(nodos, "%i", &etiqueta_nod[k]);
fclose(nodos);

// Reducción del vector f(0)
for (k = 1; k <= tot_nod; k++) {
    // Empareja subíndice de vector con numeración de nodo restringido
    nres = etiqueta_nod[k] - k;

    // Reducción de renglones
    for (i = 0; i < tot_nod - k; i++)
        if (i >= nres)
            f[2*i] = f[2*i+2];
    }

// Guarda vector f(0) reducido
vector0 = fopen(ruta_vector0, "w");
for (i = 0; i < 2*(tot_nod - tot_nres); i++)
    fprintf(vector0, "%f\n", f[i]);
fclose(vector0);

}

//*****

void reduce_vec_v0(int *tot_nod, int *nod_res)
{
    float t, v0[MAXXGL];
    int i, k, nres, etiqueta_nod[MAX_HOR];

    // Reducción del vector v(0) equivalente
    printf("11. Reducción del vector v(0)\n");
    printf("\n\nReduciendo...\n");
    fprintf(Reporte, "11. Reducción del vector v(0)\n");

    // Lee vector v(0)
    vector0 = fopen(ruta_vector0, "r");
    fscanf(vector0, "%f\n", &t);
}

```

