



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

EL PROBLEMA DE VALORES Y VECTORES CARACTERISTICOS  
EN ANALISIS ESTRUCTURAL POR COMPUTADORA.  
UNA REVISION DE ENFOQUES

T E S I S  
QUE PARA OBTENER  
EL TITULO DE  
INGENIERO CIVIL  
PRESENTA  
HECTOR URQUIJO PARRA

MEXICO, D. F.

1982



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.



Universidad Nacional  
Autónoma

FACULTAD DE INGENIERIA  
DIRECCION  
60-1-378

Señor HECTOR URQUIJO PARRA,  
P r e s e n t e .

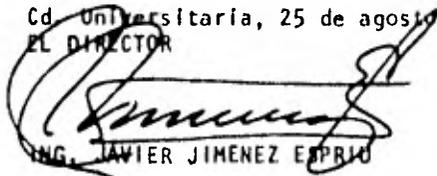
En atención a su solicitud, me es grato hacer de su conocimiento el tema que aprobado por esta Dirección propuso el Profesor Ing. Gustavo Rafael Aranda Hernández, para que lo desarrolle como tesis para su Examen Profesional de la carrera de Ingeniero CIVIL.

"EL PROBLEMA DE VALORES Y VECTORES CARACTERISTICOS EN ANALISIS ESTRUCTURAL POR COMPUTADORA, UNA REVISION DE ENFOQUES"

- I. Introducción.
- II. Métodos de solución de sistemas de ecuaciones.
- III. El problema de valores y vectores característicos.
- IV. Convergencia.
- V. Aplicaciones.
- VI. Conclusiones y recomendaciones.
- VII. Reconocimiento.
- VIII. Referencias.
- IX. Tablas, figuras y listados.

Ruego a usted se sirva tomar debida nota de que en cumplimiento con lo especificado por la Ley de Profesiones, deberá prestar Servicio Social durante un tiempo mínimo de seis meses como requisito indispensable para sustentar Examen Profesional; así como de la disposición de la Coordinación de la Administración Escolar en el sentido de que se imprima en lugar visible de los ejemplares de la tesis, el título del trabajo realizado.

Atentamente  
'POR MI RAZA HABLARA EL ESPIRITU'  
Cd. Universitaria, 25 de agosto de 1982  
EL DIRECTOR

  
ING. JAVIER JIMENEZ ESPINO

  
JJE/OBLH/ser

1.	INTRODUCCION	1
2.	MÉTODOS DE SOLUCION DE SISTEMAS DE ECUACIONES	3
2.1	Introducción	3
2.2	Métodos de solución directa	4
2.3	Comentario a los métodos de solución directa utilizando matrices ortogonales	12
2.4	Comentario a los métodos iterativos de solución	12
3.	EL PROBLEMA DE VALORES Y VECTORES CARACTERISTICOS	14
3.1	Introducción	14
3.2	Método de Jacobi	17
3.3	Método de bisección	23
3.4	Método de iteración inversa	26
3.5	Método QR	29
3.6	Método de Lanczos	34
3.7	Método de iteración inversa-Householder-QR	37
3.8	Comentario a la solución de sistemas grandes	43
4.	CONVERGENCIA	46

5.	APLICACIONES	48
6.	CONCLUSIONES Y RECOMENDACIONES	50
7.	RECONOCIMIENTO	52
8.	REFERENCIAS	53

**TABLAS**

**FIGURAS**

**LISTADOS**

## 1. INTRODUCCION

En Análisis Estructural una gran parte del tiempo de computadora es usado en la solución del problema de valores y vectores característicos y en la solución del sistema de ecuaciones lineales, por este motivo, los algoritmos de solución eficientes tienen buen reconocimiento.

El presente trabajo es una recopilación de información de lo existente en la literatura, se pretende hacer un análisis comparativo entre los diferentes métodos de solución y colaborar al desarrollo técnico, tanto formativo como de aplicación, en la solución del problema de valores y vectores característicos y la solución del sistema de ecuaciones lineales, en su aplicación a programas de computadora enfocados a problemas de análisis estructural. Sin embargo, no se presenta en forma exhaustiva el algoritmo de solución para cada método, solo se da una rápida descripción de cada uno de ellos.

En este trabajo se presentan listados de subrutinas programadas en lenguaje FORTRAN IV. Para la adaptación y ejecución de los programas se utilizó una terminal de la máquina Burroughs-6800 que se encuentra en el centro de cómputo de la UNAM.

En cuanto a la eficiencia de los procedimientos de solución se puede concluir que ninguno de los métodos es siempre absolutamente eficiente, ya que ello dependerá de varios factores, dentro de los que están el problema a ser resuelto.

## 2. METODOS DE SOLUCION DE SISTEMAS DE ECUACIONES

### 2.1 Introducción.

En este capítulo se considera la solución del sistema de ecuaciones de equilibrio en la aplicación a problemas de análisis estructural. Es muy importante en los problemas de análisis la forma de ensamblar las matrices para lograr una solución eficiente. Sin embargo, la efectividad total de un análisis depende en gran medida del procedimiento numérico usado en la solución del sistema de ecuaciones de equilibrio.

La solución del sistema de ecuaciones generalmente emplea gran parte del tiempo de procesado de una computadora, por este motivo, los algoritmos de solución eficientes tienen buen reconocimiento y aceptación en el campo profesional. Los algoritmos más comúnmente usados se basan en la eliminación de Gauss.

En este capítulo se discuten diferentes métodos de solución de sistemas de ecuaciones lineales. Se han tomado aquellos que en la literatura aparecen recomendados como los más adecuados para análisis estructural, (refs 1-3).

Para problemas de análisis estructural, el sistema de ecuaciones a ser resueltas está bien condicionado, ya que la matriz de coeficientes es generalmente simétrica, poco poblada y con la diagonal principal pesada.

Esencialmente hay dos diferentes clases de métodos para resolver un sistema de ecuaciones simultáneas: técnicas de solución directa y métodos iterativos de solución. En un método de solución directa el sistema se resuelve usando un número de pasos y operaciones que son predeterminadas de una manera exacta, mientras que en un método iterativo, como su nombre lo indica, se emplean iteraciones. Cualquiera de estos métodos de solución tienen ciertas ventajas; sin embargo, en casi todas las aplicaciones los métodos de solución directa son más eficientes, por esta razón este capítulo se destina a las técnicas de solución directa.

## 2.2 Métodos de solución directa.

**2.2.1 Método de Cholesky.** El método es adecuado únicamente para la solución de sistemas donde la matriz de coeficientes es definida positivamente, lo que implica que todos los elementos  $Q_{ij}$  sean positivos. El método de Cholesky se emplea de manera efectiva en la transformación del problema general de valores característicos a la forma estándar ( como se verá más adelante ).

El método se basa en que cualquier matriz cuadrada se puede expresar como el producto matricial de una triangular superior y una triangular inferior. Por lo tanto, el procedimiento se reduce a la inversión de dos matrices triangulares, lo cual es un procedimiento muy sencillo. En todos los problemas prácticos de análisis estructural las matrices a ser invertidas son simétricas, y únicamente una matriz triangular se necesita invertir, (refs 1 y 2).

Para aplicar el método se supone primero que la matriz  $A$ , que es no singular, se puede expresar como el producto de una matriz triangular inferior  $L$  y una matriz triangular superior  $U$ , tal que.

$$A = LU$$

Para calcular las matrices  $L$  y  $U$  se utiliza el siguiente algoritmo, (ref 1).

$$l_{ij} = a_{ij} - \sum_{r=j}^{i-1} l_{ir} U_{rj} \quad i \geq j \quad \begin{array}{l} \text{Matriz triangular} \\ \text{inferior.} \end{array}$$

$$U_{ij} = \frac{a_{ij} - \sum_{r=j}^{i-1} l_{ir} U_{rj}}{l_{ij}} \quad i < j \quad \begin{array}{l} \text{Matriz triangular} \\ \text{superior.} \end{array}$$

$$U_{ii} = 1$$

La inversión de la matriz  $A$  se hace de la siguiente manera.

$$A^{-1} = [LU]^{-1} = U^{-1}L^{-1}$$

para determinar la inversa de la matriz triangular inferior, se utilizarán las siguientes relaciones:

$$LL^{-1} = LM = I$$

$$M = L^{-1}$$

En donde  $M$  se puede obtener usando las siguientes relaciones

$$m_{ii} = 1/l_{ii}, \quad \text{para los elementos de la diagonal.}$$

$$m_{ij} = -\frac{1}{l_{jj}} \sum_{r=j}^{i-1} l_{ir} m_{rj} \quad i > j \quad \text{elementos abajo de la diagonal.}$$

$$m_{ij} = 0 \quad i < j \quad \text{elementos arriba de la diagonal.}$$

Se tiene que la inversa de una matriz triangular inferior,  $L$ , también es una matriz triangular inferior.

Similarmente, se puede mostrar que la inversa de la matriz triangular superior  $U$  también es una matriz triangular superior, tal que si

$$U^{-1} = N$$

y los elementos de  $N$  están dados por.

$$\begin{aligned} n_{ij} &= 1 && \text{para los elementos de la diagonal.} \\ n_{ij} &= -\sum_{r=j}^{i-1} n_{ir} U_{rj} && \text{elementos arriba de la diagonal.} \\ n_{ij} &= 0 && \text{elementos bajo la diagonal.} \end{aligned} \quad i < j \quad i > j$$

Habiendo calculado los elementos de las matrices triangulares  $M$  y  $N$ , se puede calcular la inversa de la matriz  $A$  como.

$$A^{-1} = U^{-1} L^{-1} = NM$$

Si en el cálculo de la matriz triangular inferior,  $L$ , se encuentra algún elemento de la diagonal que sea nulo,  $l_{jj} = 0$ , entonces el método falla, y será necesario intercambiar el  $i$ -ésimo renglón con cualquier otro renglón.

Cuando la matriz  $A$  es simétrica, el procedimiento para hallar la inversa se puede acortar. Esto es posible debido a que  $A$  se puede expresar en la forma:

$$A = \Lambda \Lambda^T$$

Donde  $\Lambda$  es una matriz triangular inferior.

Al aplicar el método usado para hallar los elementos de  $L$  y  $U$ , se puede mostrar que los elementos de  $\Lambda$  están dados por.

$$\lambda_{ii} = (a_{ii} - \sum_{r=1}^{i-1} \lambda_{ir}^2)^{1/2} \quad i = j$$

$$\lambda_{ij} = \frac{a_{ij} - \sum_{r=1}^{i-1} \lambda_{ir} \lambda_{jr}}{\lambda_{jj}} \quad i > j$$

$$\lambda_{ij} = 0 \quad i < j$$

Los elementos  $U_{ij}$  de la matriz inversa de  $\Lambda$  pueden determinarse de  $\Lambda \Lambda^{-1} = I$ , lo cual conduce a lo siguiente:

$$U_{ij} = 1 / \lambda_{ii} \quad i = j$$

$$U_{ij} = - \frac{\sum_{r=1}^{i-1} \lambda_{ir} U_{rj}}{\lambda_{ii}} \quad i > j$$

$$U_{ij} = 0 \quad i < j$$

La inversa de la matriz  $\Lambda$  es una matriz triangular inferior. Para determinar los elementos de  $\Lambda^{-1}$  se debe notar que para cualquier matriz no singular  $A$ .

$$(A^T)^{-1} = \frac{\hat{A}^T}{|A|} = (A^{-1})^T$$

y por tanto

$$A^{-1} = (\Lambda \Lambda^T)^{-1} = (\Lambda^{-1})^T \Lambda^{-1}$$

Que es lo que se buscaba.

2.2.2. Eliminación de Gauss. En el procedimiento de eliminación de Gauss, la primera ecuación se elimina por una combinación de ella con las siguientes ecuaciones; esto se hace para reducir a cero los elementos  $a_{ij}$   $j=2, \dots, N$ . El proceso se repite para cada grupo de ecuaciones a ser reducidas hasta que la última ecuación es de la forma  $K_{NN}U_N = r_N$ , (ref 3).

La solución se obtiene por un procedimiento de sustitución regresiva.

$$AX = b \quad (2.2.2.1)$$

Si las ecuaciones 1 a la K-1 han sido eliminadas, las operaciones para eliminar la ecuación k-ésima puede escribirse como se indica a continuación.

Reducción de la matriz de coeficientes

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)} a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}} \quad \begin{array}{l} i = k+1, \dots, N \\ j = k, \dots, N \end{array} \quad (2.2.2.2)$$

Reducción del vector de cargas

$$b_i^{(k)} = b_i^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} b_k^{(k-1)} ; \quad i = k+1, \dots, N \quad (2.2.2.3)$$

en donde el índice superior (k) indica el siguiente elemento a ser eliminado de la K-ésima ecuación.

Ahora se puede ver que la reducción del vector de cargas es equivalente a tratar este vector como una columna (N+1)-ésima de la matriz de coeficientes y aplicar la operación de la ec(2.2.2.2), se puede aplicar la ec(2.2.2.3) por separado, o la ec(2.2.2.2) a la matriz argumento  $[A, b]$ .

Después de haberse reducido todo a una sola ecuación, el procedimiento de sustitución regresiva es el siguiente.

$$X = \frac{b_N^{(N-1)}}{a_{NN}^{(N-1)}} ; \quad X_i = \frac{b_i^{(i-1)} - \sum_{j=i+1}^N d_{ij}^{(i-1)} X_j}{a_{ii}^{(i-1)}}, \quad i=N-1, \dots, 1. \quad (2.2.2.4)$$

Para fines de ahorrar memoria de computadora, la solución se almacena en el vector  $b$ . Ahora, para un número cualquiera de vectores de carga, se repetirán las ecs(2.2.2.3) y (2.2.2.4) sin efectuar operaciones adicionales en la matriz de coeficientes.

**2.2.3 Reducción de Crout.** La principal característica del procedimiento de reducción de Crout es la reordenación de la secuencia en la cual los términos de la matriz de coeficientes se modifican. En el procedimiento de eliminación de Gauss, la modificación de cada término, en la matriz de coeficientes reducida, se hace cada vez que se elimina una ecuación. Para calcular a mano este procedimiento es tedioso porque requiere que la matriz de coeficientes se escriba muchas veces. Ahora, en el procedimiento de reducción de Crout, solo es necesario volver a escribir la matriz de coeficientes una vez, porque cada término se cambia directamente de su valor inicial, en la matriz sin reducir, a su valor final en la matriz triangularizada llena (ref 3).

Esta secuencia de reducción tiene ventajas si se utiliza una computadora, porque es posible eliminar todas las operaciones innecesarias. Ahora, las operaciones de reducción para una matriz de coeficientes y un vector de cargas puede escribirse de la siguiente manera

$$a_{ij}^{(i-1)} = a_{ij} - \sum_{k=2}^{i-1} \frac{a_{ki}^{(k-1)} a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}} \quad j=2, \dots, N$$

$$b_i^{(i-1)} = b_i - \sum_{k=2}^{i-1} \frac{a_{ki}^{(k-1)}}{a_{kk}^{(k-1)}} b_k \quad \begin{array}{l} i=2, \dots, j \\ i=2, \dots, N \end{array}$$

Procedimiento de sustitución regresiva

$$X_N = \frac{b_N^{(N-1)}}{a_{NN}^{(N-1)}} ; \quad X_i = \frac{b_i^{(i-1)} - \sum_{j=i+1}^N a_{ij}^{(i-1)} X_j}{a_{ii}^{(i-1)}} \quad , \quad i=N-1, \dots, 1$$

Con el algoritmo de solución de ecuaciones descrito por Jensen y Parks (ref 3) se tiene la posibilidad de reconocer situaciones en las cuales no se crean nuevos enlaces, de manera que un número mayor de operaciones innecesarias puedan eliminarse. Sin embargo, este algoritmo requiere la creación de arreglos de índices los cuales consumen espacio y son complicados. La necesidad de almacenar arreglos de índices es desventajoso para una solución en una computadora porque consumen espacio, sobre todo si se trata de una computadora con poca capacidad.

Las ecuaciones modificadas de la reducción de Crout pueden resumirse como sigue.

Reducción de la matriz de coeficientes A

$$a_{ij}^{(i-1)} = a_{ij} - \sum_{k=2}^{i-1} \bar{a}_{ki}^{(k-1)} a_{kj}^{(k-1)}$$

en donde

$$\bar{a}_{ki}^{(k-1)} = a_{ki}^{(k-1)} / a_{kk}^{(k-1)}$$

$$j=2, \dots, N$$

$$i = L_j + 1, \dots, j - 1$$

$$K_0 = \max(L_i, L_j)$$

y

$$a_{jj}^{(j-1)} = a_{jj} - \sum_{k=L_j}^{j-1} \bar{a}_{kj}^{(k-1)} a_{kj}^{(k-1)}, \quad j = 2, \dots, N. \quad (2.2.3.1)$$

Reducción del vector de cargas

$$b_i^{(i-1)} = b_i - \sum_{k=L_{N+1}}^{i-1} \bar{a}_{ki}^{(k-1)} b_k^{(k-1)} \quad i = L_{N+1} + 1, \dots, N \quad (2.2.3.2)$$

$$b_i^{(i-1)} = b_i^{(i-1)} / a_{ii}^{(i-1)} \quad i = L_{N+1}, \dots, N \quad (2.2.3.3)$$

La ec(2.2.3.3) se realiza después que se ha completado la ec(2.2.3.2).

En estas ecuaciones  $L_{N+1}$  = número de renglón del primer elemento diferente de cero que entra en el vector  $b$ .

Sustitución regresiva.

$$x_k = b_k^{(k-1)} \quad k = 1, \dots, N$$

$$x_k = x_k - \bar{a}_{ki}^{(k-1)} x_i \quad i = N, N-1, \dots, 2$$

$$k = L_i, L_{i+1}, \dots, i-1$$

Nótese que  $a_{kj}^{(k-1)}$  es remplazada por  $\bar{a}_{kj}^{(k-1)} = a_{kj}^{(k-1)} / a_{kk}^{(k-1)}$  cuando se aplica la ec(2.2.3.1). Nótese también que las ecuaciones para el procedimiento de sustitución regresiva se escribe de manera que sea consistente con este algoritmo.

### 2.3 Comentario a los métodos de solución directa utilizando matrices ortogonales

En la práctica en análisis estructural siempre que se trate de un esquema de solución directa, es el procedimiento de solución de Gauss el que se ha empleado más frecuentemente en cualquiera de sus variantes. El procedimiento es efectivo en costo y numéricamente estable; sin embargo, debe hacerse notar que una solución directa puede también llevarse a cabo usando otras técnicas. En este punto se definen rotaciones como otro tipo de matrices ortogonales que se pueden usar para reducir  $K$  a una matriz triangular superior. Como se mencionó antes, las factorizaciones de Givens y Householder no son usadas frecuentemente en la práctica, primero porque el procedimiento de solución no es económico, si se compara con el esquema de reducción de Gauss. Durante la factorización de la matriz de rigidez, las propiedades de simetría y ancho de banda se pierden. Una ventaja de ambas factorizaciones, la de Givens y Householder, es que son bastante estables y pueden ser usadas efectivamente en la solución de sistemas de ecuaciones mal condicionadas. Sin embargo, el propósito principal de la presentación del procedimiento de la factorización de Givens y Householder es el introducir el uso de rotación de matrices.

### 2.4 Comentario a los métodos iterativos de solución

En la actualidad todos los programas grandes en análisis estructu-

ral utilizan alguna forma de eliminación de Gauss para resolver el sistema de ecuaciones. Sin embargo, es interesante notar que durante el desarrollo inicial de los métodos de elemento finito, los algoritmos iterativos de solución se emplearon extensamente y, en la actualidad, muchas investigaciones se dedican a mejorar varios esquemas iterativos de solución. Una desventaja básica de un método iterativo de solución es que el tiempo empleado en la solución puede estimarse muy aproximadamente, ya que el número de iteraciones requeridas en la convergencia depende de la condición de la matriz  $K$  y qué tan efectivos sean los factores de aceleración usados.

Las soluciones directas usando una de las técnicas vistas al principio del capítulo son siempre más efectivas. Sin embargo, al menos para un propósito ilustrativo es importante familiarizarse también con un método iterativo de solución que ha sido usado extensamente, llamado, el método de Gauss-Seidel. El objetivo en esta sección es comentar brevemente el método iterativo de solución de Gauss-Seidel. Aunque los métodos de solución directa son usados casi exclusivamente en programas de análisis estructural, es importante reconocer algunas ventajas del método de Gauss-Seidel. Los esquemas de solución son usados efectivamente en el problema de áreas de reanálisis y optimización; aunque una forma de eliminación de Gauss puede ser más eficiente. Si en reanálisis una estructura a sido cambiada levemente, la solución anterior es un buen comienzo para la solución iterativa de Gauss-Seidel de la nueva estructura.

### 3. EL PROBLEMA DE VALORES Y VECTORES CARACTERISTICOS

#### 3.1 Introducción

El problema generalizado de valores característicos es encontrado en muchos campos y una solución numérica eficiente ha sido objeto de muchas investigaciones recientes. En análisis estructural toma usualmente la forma

$$K\phi = \lambda M\phi$$

Donde  $K$  y  $M$  son las matrices de rigidez y de masas; poseen algunas propiedades distintivas las cuales a menudo son usadas para mejorar la eficiencia en la solución. Las propiedades más importantes son la simetría, lo poco poblado (porosas y bandeadas) y su condición de positiva definida.

Antes de discutir algunos algoritmos de solución, se presentan importantes consideraciones básicas para la solución del problema de valores característicos.

El problema más simple es el de la forma estándar de valores característicos

$$K\phi = \lambda\phi \quad (3.1.1)$$

Donde  $K$  es la matriz de rigidez, se recordará que  $K$  es simétrica y positiva definida. Existen  $n$  valores característicos y correspondientes vectores característicos que satisfacen la ec(3.1.1).

El  $i$ -ésimo par de valor y vector característico está dado por

$(\lambda_i, \phi_i)$ , donde los valores característicos son ordenados de acuerdo con sus magnitudes:

$$0 \leq \lambda_1 < \lambda_2 < \dots \leq \lambda_m < \lambda_n$$

La solución para  $P$  parejas de valor y vector característico se puede escribir como:

$$K\Phi = \Phi\Lambda \quad (3.1.2)$$

Donde  $\Phi$  es una matriz de orden  $n \times P$  en donde las columnas son iguales a los  $P$  vectores característicos y  $\Lambda$  es una matriz diagonal en donde los elementos de la diagonal corresponden a los valores característicos. En cualquier caso  $\Phi = [\phi_1, \dots, \phi_P]$  y  $\Lambda = \text{diag}(\lambda_i, i=1, \dots, P)$ . Se recuerda que si  $K$  es positiva definida,  $\lambda_i > 0$ ,  $i=1, \dots, n$  y si  $K$  es semidefinida positiva  $\lambda_i \geq 0$ ,  $i=1, \dots, n$ , donde el número de valores característicos con valor cero es igual a los modos en el sistema de cuerpo rígido.

Con mucha frecuencia se considera la superposición de modos en el problema de valores característicos para el análisis de vibraciones.

En tal caso, se considera el problema generalizado de valores característicos.

$$K\phi = \lambda M\phi \quad (3.1.3)$$

Donde  $K$  y  $M$  son respectivamente, la matriz de rigidez y la de masas. Los valores característicos  $\lambda_i$  y vectores característicos  $\phi_i$  son las frecuencias en vibraciones libres y correspondientes vectores de forma modal, respectivamente. Una matriz de masas en banda, obtenida en un análisis consistente siempre es positiva definida, mientras una matriz de masas concentradas es positiva definida únicamente si todos los elementos de la diagonal son mayores que cero. En general, una matriz de masas es semidefinida positiva.

En analogía con la ec(3.1.2), la solución para  $P$  valores característicos y correspondientes vectores característicos de la ec(3.1.3) puede escribirse:

$$K\Phi = M\Phi\Lambda$$

Donde las columnas en  $\Phi$  son los vectores característicos y  $\Lambda$  es una matriz diagonal cuyos elementos corresponden a los valores característicos.

Se debe notar que el problema generalizado en la ec(3.1.3) se puede reducir al problema estándar de la ec(3.1.1) si  $M$  es una matriz identidad.

Aquí y en los siguientes puntos se discutirá la solución de los problemas de valores característicos  $K\phi = \lambda\phi$  y  $K\phi = \lambda M\phi$  de las ecs(3.1.1) y (3.1.3). Se debe hacer notar que todos los algoritmos que se presentan son también aplicables a la solución de otros problemas de valores característicos, siempre que sean de la misma for-

ma y las matrices satisfagan las condiciones de positiva definida, y semidefinida.

Considerando la solución actual con computadora de los problemas de valores característicos, se recuerda la importancia de usar procedimientos de cálculo efectivos. Esto es más importante en el cálculo de valores y vectores característicos, porque en general requiere mucho más esfuerzo de computadora que la solución de las ecuaciones de equilibrio estático. Una consideración importante es que los algoritmos de solución deberán ser estables, lo cual es más difícil de llevar a cabo que en análisis estático.

Una variedad de métodos de solución de sistemas de valores característicos se han desarrollado, aunque la mayoría de los métodos han sido dados para matrices bastante generales. Ahora, en un análisis de elemento finito o de análisis estructural, la solución requerida es de un problema específico, en donde cada matriz tiene propiedades específicas. Los algoritmos de solución deberán tomar ventaja de esas — propiedades para hacer la solución lo más económica posible.

### 3.2 Método de Jacobi

3.2.1 Solución del problema estándar. El método de Jacobi es un método iterativo que transforma una matriz real simétrica en una matriz diagonal, en donde los elementos de la diagonal son los valores característicos.

Este método consiste en aplicar a una matriz una sucesión de rotaciones en el plano, destinadas a reducir a cero todos los elementos fuera de la diagonal.

El método básico de solución de Jacobi ha sido desarrollado para resolver el problema de valores característicos en su forma estándar, este método ha sido extensamente usado y unas de sus mayores ventajas son la simplicidad y estabilidad. Para su desarrollo, se pueden escribir las siguientes ecuaciones

$$\Phi K \Phi = \Lambda \quad (3.2.1.1)$$

$$\Phi M \Phi = I \quad (3.2.1.2)$$

Ahora, aplicando las propiedades de los vectores característicos en las ecs(3.2.1.1) y (3.2.1.2) con  $M=I$  son aplicables a todas las matrices simétricas  $K$  con ninguna restricción en los valores característicos, el método de Jacobi puede ser usado para calcular valores característicos, con valor positivo, negativo o cero.

En el método de solución de Jacobi se define la matriz  $P_k$  como una matriz de rotación la cual, es seleccionada de manera que reduzca a cero un elemento fuera de la diagonal en la matriz  $K_k$ . Si el elemento  $(i,j)$  es el que se quiere reducir a cero, la correspondiente matriz ortogonal  $P_k$  es.

$$P_k = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & \cos \Theta & & & \\ & & & -\sin \Theta & & \\ & & & & \ddots & \\ & & \sin \Theta & & & \cos \Theta \\ & & & & & & 1 \end{bmatrix}$$

i-ésima
j-ésima columna

i-ésimo
j-ésimo renglón

Donde  $\Theta$  es seleccionada de la condición que el elemento  $(i,j)$  en  $K_{k+1}$  será cero.

Llamando al elemento  $(i, j)$  en  $K_k$  por  $K_{ij}^{(k)}$  se usa.

$$\text{tang} 2\theta = \frac{2K_{ij}^{(k)}}{K_{ii}^{(k)} - K_{jj}^{(k)}} \quad K_{ii}^{(k)} \neq K_{jj}^{(k)}$$

y

$$\theta = \pi / 4 \quad \text{para} \quad K_{ii}^{(k)} = K_{jj}^{(k)}$$

Se debe notar que la evaluación numérica de  $K_{k+1}$  requiere únicamente la combinación lineal de dos renglones y dos columnas. También se puede obtener ventaja del hecho que la matriz  $K_k$  es simétrica para toda  $K$ ; se trabaja únicamente con la parte triangular superior(o inferior), incluyendo los elementos de la diagonal.

Es importante notar que aunque la transformación reduce un elemento fuera de la diagonal a cero, este elemento puede cambiar de valor durante las transformaciones siguientes. Por esta razón, para el diseño de un algoritmo se tiene que decidir cuál elemento se ha de reducir a cero. Una elección consiste en hacer siempre cero el elemento mayor fuera de la diagonal. Ahora, la búsqueda del elemento mayor consume tiempo y puede ser preferible, por simplicidad, llevar a cabo una transformación sistemática, renglón por renglón o columna por columna, el cual es conocido como el procedimiento cíclico de Jacobi.

Un procedimiento que ha sido usado muy efectivamente es el método del umbral de Jacobi, en donde los elementos fuera de la diagonal son probados secuencialmente, es decir, renglón por renglón (o columna por columna), y la rotación se aplica solamente si el elemento es más grande que el umbral dado para que sea barrido.



para  $\alpha$  y  $\gamma$ :

$$\alpha K_{ii}^{(k)} + (1 + \alpha \gamma) K_{ij}^{(k)} + \gamma K_{jj}^{(k)} = 0 \quad (3.2.2.1)$$

y

$$\alpha m_{ii}^{(k)} + (1 + \alpha \gamma) m_{ij}^{(k)} + \gamma m_{jj}^{(k)} = 0 \quad (3.2.2.2)$$

si

$$\frac{K_{ii}^{(k)}}{m_{ii}^{(k)}} = \frac{K_{jj}^{(k)}}{m_{jj}^{(k)}} = \frac{K_{ij}^{(k)}}{m_{ij}^{(k)}}$$

(las submatrices consideradas son múltiplos escalares, los cuales

pueden ser considerados como un caso trivial), se usa  $\alpha = 0$  y  $\gamma = -K_{ij}^{(k)} / K_{jj}^{(k)}$ .

En general, se resuelve para  $\alpha$  y  $\gamma$  de las ecs (3.2.2.1) y (3.2.2.2)

se define.

$$\begin{aligned} \bar{K}_{ii}^{(k)} &= K_{ii}^{(k)} m_{ij}^{(k)} - m_{ii}^{(k)} K_{ij}^{(k)} \\ \bar{K}_{jj}^{(k)} &= K_{jj}^{(k)} m_{ij}^{(k)} - m_{jj}^{(k)} K_{ij}^{(k)} \\ \bar{K}^{(k)} &= K_{ii}^{(k)} m_{ij}^{(k)} - K_{jj}^{(k)} m_{ii}^{(k)} \end{aligned}$$

y

$$\gamma = -K_{ij}^{(k)} / X; \quad \alpha = \bar{K}_{jj}^{(k)} / X$$

El valor de  $X$  para obtener  $\alpha$  y  $\gamma$  se determina usando.

$$X = \frac{\bar{K}^{(k)}}{2} + \text{sign}(\bar{K}^{(k)}) \sqrt{\left(\frac{\bar{K}^{(k)}}{2}\right)^2 + \bar{K}_{ii}^{(k)} \bar{K}_{jj}^{(k)}}$$

Las relaciones de arriba para  $\alpha$  y  $\gamma$  son usadas y tienen un buen desarrollo para el caso en que  $M$  la matriz de masas es positiva definida, llena o en banda.

En tal caso (y también bajo menos condiciones restrictivas) se tiene que

$$\left(\frac{\bar{K}^{(k)}}{2}\right)^2 + \bar{K}_{ii}^{(k)} \bar{K}_{jj}^{(k)} > 0.$$

y por lo tanto  $X$  siempre será diferente de cero. En suma, también,

el  $\det R \neq 0$ , lo cual es una condición necesaria para que el algoritmo trabaje.

En la discusión anterior del método generalizado de Jacobi se indicaron algunas ventajas de la técnica de solución. Primero, la transformación del problema generalizado a la forma estándar se puede evitar. Esto es particularmente ventajoso cuando las matrices están mal condicionadas, y los elementos fuera de la diagonal en  $K$  y  $M$  son pequeños o, equivalentemente, cuando haya únicamente pocos valores diferentes de cero fuera de la diagonal. En el primer caso la solución directa de  $K\phi = \lambda M\phi$  evita la solución de un problema estándar de una matriz con elementos muy grandes y pequeños. En el segundo caso el problema de valores característicos es necesariamente resuelto ya que los elementos son cero, pequeños o únicamente unos pocos elementos fuera de la diagonal en  $K$  y  $M$  no resultan un cambio grande en los elementos de la diagonal de las matrices, los cocientes de los cuales son los valores característicos. En suma, la rápida convergencia puede tardar cuando los elementos fuera de la diagonal son pequeños.

Se debe hacer notar que el método de solución de Jacobi resuelve el problema simultáneamente para todos los valores y vectores característicos correspondientes. Sin embargo, en análisis estructural se requiere únicamente algunos valores característicos y el uso del método de Jacobi puede ser muy ineficiente, en particular cuando el orden de  $K$  y  $M$  son grandes. En tal caso existen métodos de solución que únicamente resuelven el problema para un número determinado de valores y vectores característicos requeridos.

Sin embargo, el método de Jacobi puede ser muy eficiente, cuando el orden de las matrices  $K$  y  $M$  es relativamente pequeño, la solución del sistema no es muy cara y puede también ser atractivo por su simplicidad y elegancia en la solución.

### 3.3 Método de Bisección

Los valores característicos  $\lambda$  son las raíces del determinante ahora, el número de raíces de la ec(3.3.1) se obtiene por la sucesiva aplicación del método de Bisección, que se explica a continuación. En este método se supone una raíz aproximada  $\lambda_1$  y un incremento  $\Delta\lambda_1$ . Se calcula  $\Delta_1$  y  $\Delta_2$  donde.

$$|K - \lambda M| = 0 \quad (3.3.1)$$

y

$$\begin{aligned} \Delta_1 &= |K - \lambda_1 M| \\ \Delta_2 &= |K - (\lambda_1 + \Delta\lambda) M| \end{aligned}$$

Si estos determinantes tienen signos opuestos la raíz se encontrará entre  $\lambda_1$  y  $\lambda_1 + \Delta\lambda_1$ . Ahora, el intervalo  $\Delta\lambda_1$  se reduce a la mitad y en el nuevo punto  $\lambda_1 + \Delta\lambda_1/2$ , se calcula el determinante  $\Delta_3$ .

$$\Delta_3 = |K - (\lambda_1 + \Delta\lambda_1/2) M|$$

Se revisa el signo de los determinantes  $\Delta_1$ ,  $\Delta_3$  y  $\Delta_2$ . Si la primera de las dos parejas tiene diferente signo, entonces la raíz se encontrará entre  $\lambda_1$  y  $\lambda_1 + \Delta\lambda_1/2$ . El intervalo  $\Delta\lambda_1/2$  se reduce a la mitad y en el nuevo punto  $\lambda_1 + \Delta\lambda_1/4$  se calcula el determinante  $\Delta_4$ .

$$\Delta_4 = |K - (\lambda_1 + \Delta\lambda_1/4) M|$$

El procedimiento continúa hasta conseguir la exactitud requerida para las raíces.

Si al comenzar, los determinantes  $\Delta_1$  y  $\Delta_2$  tienen el mismo signo, entonces la raíz no se encuentra entre  $\lambda_1$  y  $\lambda_1 + \Delta\lambda_1$ , y se tendrá que dar otro incremento y en el nuevo punto  $\lambda_1 + 2\Delta\lambda_1$  se calcula el determinante  $\Delta'_3$ .

$$\Delta'_3 = |K - (\lambda_1 + 2\Delta\lambda_1)M|$$

Si los determinantes  $\Delta_2$  y  $\Delta'_3$  tienen diferente signo, el procedimiento anterior se emplea para evaluar la raíz. Por otro lado, si el determinante tiene el mismo signo entonces la raíz se incrementa adicionalmente hasta conseguir un cambio de signo.

### 3.3.1 Cálculo de los valores característicos de una matriz tridiagonal simétrica por el método de bisección.

Este método se basa en el teorema de Gershgorin y las propiedades de la sucesión de Sturm.

Ahora, si  $A$  es una matriz tridiagonal simétrica, (ref. 2).

$$A = \begin{bmatrix} d_1 & b_1 & 0 & & 0 \\ b_1 & d_2 & b_2 & & 0 \\ 0 & b_2 & d_3 & & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & b_{n-1} & d_n \\ & & & b_n & d_n \end{bmatrix}$$

Entonces la sucesión principal menor de la matriz  $A - \lambda I$  es.

$$P_1(\lambda) = d_1 - \lambda, \quad P_2(\lambda) = (d_2 - \lambda)(d_1 - \lambda) - b_1^2, \text{ etc.}$$

Para introducir  $P(\lambda) = 1$  se puede calcular todos los menores de la relación de recurrencia.

$$P_i(\lambda) = (d_i - \lambda)P_{i-1}(\lambda) - b_{i-1}^2 P_{i-2}(\lambda) \quad (i=2, \dots, n)$$

Ahora, si se define  $\alpha(\lambda)$  como el número de cambios de signo en la secuencia  $P_0(\lambda), P_1(\lambda), \dots, P_n(\lambda)$ , esto es, como el número de

veces en el cual los miembros consecutivos en la secuencia difieren en signo.

Barth y asociados (ref 8) mencionan que el cálculo de  $Q(\lambda)$  por este método es muy estable para matrices de orden modesto. La dificultad es evitada reemplazando la sucesión  $P_i(\lambda)$  por una sucesión  $q_i(\lambda)$  definida por.

$$q_i(\lambda) = P(\lambda) / P(\lambda) \quad (i=1(1)n)$$

El número  $Q(\lambda)$  es ahora dado por el número negativo  $q_i(\lambda)$ . El  $q_i(\lambda)$  satisface la relación.

$$q_i(\lambda) = d_i - \lambda$$

$$q_i(\lambda) = d_i - \lambda - b_i / q_{i-1}(\lambda)$$

A primera vista esta relación parece muy peligrosa, porque  $q_{i-1}(\lambda)$  puede ser cero para alguna  $i$ , en tal caso es necesario reemplazar el valor cero  $q_{i-1}(\lambda)$  por una apropiada cantidad pequeña.

Comparando el cálculo de  $q_i(\lambda)$  con el de  $P_i(\lambda)$  se observa que dos multiplicaciones se han reemplazado por una división; adicionalmente, solo se tiene que detectar el signo de cada  $q_i(\lambda)$  en vez de comparar el signo de  $P_{i-1}(\lambda)$  y  $P_i(\lambda)$ .

Es fácil ver que cuando trabajemos con el  $q_i(\lambda)$  la condición  $b_i \neq 0$  se puede omitir.

El teorema de Genshgorin muestra que los límites superior e inferior en los valores característicos están dados por.

$$\begin{aligned} \chi_{\max} &= \max \{ d_i + |b_i| + |b_{i+1}| \} \\ \chi_{\min} &= \min \{ d_i - |b_i| - |b_{i+1}| \} \end{aligned}$$

los valores característicos se encuentran en orden descendente.

### 3.4 Método de Iteración Inversa.

El método de Iteración Inversa es efectivo para calcular un vector característico y el valor característico correspondiente.

Se supone que  $K$  es positiva definida, mientras  $M$  es la matriz de masas que puede ser diagonal, con o sin ceros en la diagonal, o bien puede ser una matriz de masas en banda. Si  $K$  es semidefinida positivamente, se deberá efectuar un cambio antes de la iteración, (ref 2).

En la solución se considera un vector inicial para la iteración y luego se evalúa en cada paso iterativo  $K=1,2,\dots$  :

$$K\bar{x}_{K+1} = Mx_K \quad (3.4.1)$$

y

$$x_{K+1} = \frac{\bar{x}_{K+1}}{(\bar{x}_{K+1}^T M \bar{x}_{K+1})^{1/2}} \quad (3.4.2)$$

con la condición que  $x_i$  no sea ortogonal a  $M$  para  $\phi_i$  significa que  $x_i^T M \phi_i \neq 0$ , se tiene que

$$x_{K+1} \longrightarrow \phi_i \quad \text{cuando} \quad K \longrightarrow \infty$$

El paso básico en la iteración es la solución de la ec(3.4.1) en la cual se evalúa un vector  $\bar{x}_{K+1}$  con una dirección cercana a un vector característico previo  $x_K$ . El cálculo en la ec(3.4.2) solo asegura que el tamaño de la  $M$ -pesada del nuevo vector iterativo  $x_{K+1}$  sea la unidad;

se necesita que  $x_{K+1}$  satisfaga la relación de ortogonalidad de la masa.

$$x_{K+1}^T M x_{K+1} = 1 \quad (3.4.3)$$

Sustituyendo en  $x_{K+1}$  de la ec(3.4.2) a la (3.4.3) se encuentra que

la ec(3.4.3) en verdad se satisface.

Si el escalar en la ec(3.4.2) no se incluye en la iteración, los elementos del vector iterativo crecen (o decrecen) en cada paso y el vector iterativo no converge a  $\phi_j$  solo a un múltiplo de este.

Las relaciones en las ecs(3.4.1) y (3.4.2) son el algoritmo básico de iteración inversa. Ahora en la implementación actual a una computadora es más efectivo iterar de la siguiente forma. Asumiendo que  $y_k = Mx_k$ , se evalúa para  $K=1, 2, \dots$ ,

$$K \bar{x}_{K+1} = y_K \quad (3.4.4)$$

$$\bar{y}_{K+1} = M \bar{x}_{K+1} \quad (3.4.5)$$

$$\rho(\bar{x}_{K+1}) = \frac{\bar{x}_{K+1}^T y_K}{\bar{x}_{K+1}^T \bar{y}_{K+1}} \quad (3.4.6)$$

$$y_{K+1} = \frac{\bar{y}_{K+1}}{(\bar{x}_{K+1}^T \bar{y}_{K+1})^{1/2}} \quad (3.4.7)$$

donde, con la condición que  $y_1^T \phi_j \neq 0$ ,

$$y_{K+1} \longrightarrow M \phi_j \quad (\bar{x}_{K+1}) \longrightarrow \lambda_j \text{ cuando } K \longrightarrow \infty$$

Se debe notar que se pasa de la ec(3.4.4) a(3.4.7) con el cálculo del producto matricial  $Mx$  en la ec(3.4.1) para iterar en  $y_K$  un poco más que en  $x_K$ . Solo el valor de  $\bar{y}_{K+1}$  es evaluado en uno u otro procedimiento;  $\bar{y}_{K+1}$  es calculado en la ec(3.4.2) y será evaluado en la ec(3.4.5). Usando el segundo procedimiento de iteración se obtiene en la ec(3.4.6) una aproximación al valor característico  $\lambda_j$  dado por el cociente de Rayleigh  $\rho(\bar{x}_{K+1})$ . Esto es una aproximación a  $\lambda_j$  la cual es convenientemente usada para determinar la convergencia en la iteración.

3.4.1 Iteración Frontal. Este método es complementario al método de iteración inversa, (ref 2).

Mientras que en iteración inversa se supone que  $K$  es positiva definida, aquí se considera que  $M$  es definida positiva; de otra manera se hará un cambio. Se tiene que elegir un vector inicial de iteración  $X_1$ , en el método de iteración frontal se evalúa, para  $K=1, 2, \dots$

$$y \quad M\bar{x}_{k+1} = Kx_k \quad (3.4.8)$$

$$x_{k+1} = \frac{\bar{x}_{k+1}}{(\bar{x}_{k+1}^T M \bar{x}_{k+1})^{1/2}} \quad (3.4.9)$$

con la condición que  $X_1$  no es ortogonal a  $M$  para  $\phi_n$ , se tiene que

$$x_{k+1} \rightarrow \phi_n \quad \text{cuando} \quad K \rightarrow \infty$$

la analogía con el de iteración inversa es que se resuelve la ec (3.4.8) antes que (3.4.1) para obtener un mejor vector característico. Esto significa que en la iteración inversa se necesita triangularizar la matriz  $K$  y en la iteración frontal se descompone  $M$ . Un procedimiento de iteración frontal más efectivo que el de las ecs(3.4.8) y (3.4.9) se puede obtener usando ecuaciones que son análogas a aquellas vistas en las ecs(3.4.4) a (3.4.10). Suponiendo que  $y_1 = Kx_1$ , se evalúa para  $K=1, 2, \dots, \dots$ ,

$$M\bar{x}_{k+1} = y_k \quad (3.4.10)$$

$$\bar{y}_{k+1} = K\bar{x}_{k+1} \quad (3.4.11)$$

$$\rho(\bar{x}_{k+1}) = \frac{\bar{x}_{k+1}^T \bar{y}_{k+1}}{\bar{x}_{k+1}^T y_k} \quad (3.4.12)$$

$$y_{k+1} = \frac{\bar{y}_{k+1}}{(\bar{x}_{k+1}^T y_k)^{1/2}} \quad (3.4.13)$$

con la condición que  $\phi_n^T y_j \neq 0$

$$y_{k+1} \xrightarrow{K} K \phi_n \quad \rho(\bar{x}_{k+1}) \xrightarrow{K} \lambda_n \quad \text{cuando } K \rightarrow \infty$$

### 3.5 Método QR

En el presente este es el más eficiente y amplio método usado en el cálculo de todos los valores característicos de una matriz. El método fue publicado en 1961 por J.G.F. Francis y después ha sido sujeto a intensa investigación. El método es bastante complejo en su teoría y aplicación. A continuación se da una introducción al método (refs 5 y 6).

Dada una matriz  $A$ , hay una factorización tal que

$$A = QR$$

En donde  $R$  es una matriz triangular y  $Q$  es una matriz ortogonal.

Se permite que  $A_1 = A$ , y se define una secuencia de matrices  $A_m, Q_m$  y  $R_m$  por

$$A_m = Q_m R_m \quad A_{m+1} = R_m Q_m \quad m=1, 2, \dots, \quad (3.5.1)$$

Si  $A$  satisface cualquier número de suposiciones diferentes en esta forma, entonces la secuencia  $\{A_m\}$  converge a una matriz triangular con los valores característicos de  $A$  en su diagonal o a una cercana matriz triangular de la cual los valores característicos pueden ser calculados fácilmente. En esta forma la convergencia es lenta; y una técnica conocida como cambio de origen es usada para acelerar la convergencia.

De la ec(3.5.1) serán derivadas las siguientes propiedades.

De la ec(3.5.1)  $R_m = Q_m^T A_m$ , y hasta aquí.

$$A_{m+1} = Q_m^T A_m Q_m \quad (3.5.2)$$

$A_{m+1}$  es ortogonal y similar a  $A_m$ , y así por inducción  $A = A$ . De la ec(3.5.2)

$$A_{m+1} = Q_m^T \dots Q_1^T A_1 Q_1 \dots Q_m \quad (3.5.3)$$

introduciendo las matrices  $P_m$  y  $U_m$  en

$$P_m = Q_1 \dots Q_m \quad U_m = R_m \dots R_1$$

y de la ec(3.5.3)

$$A_{m+1} = P_m^T A_1 P_m \quad m \geq 1$$

La matriz  $P_m$  es ortogonal y  $U_m$  es una matriz triangular superior:

De la ec(3.5.2) con  $m$  reemplazando a  $m+1$

$$Q_{m-1} \dots Q_m^T A_m = A_{m-1} Q_{m-1} \dots Q_m$$

usando esto,

$$P_m U_m = A_{m-1} Q_{m-1} \dots Q_m R_{m-1} \dots R_1 = A_{m-1} P_{m-1} U_{m-1}$$

Después  $P_m U_m = A_{m-1} P_{m-1} U_{m-1}$ , por inducción en la última ecuación

$$P_m U_m = A_1^m \quad m \geq 1$$

El método QR puede ser relativamente caro porque la factorización QR consume tiempo cuando se repite muchas veces. Para disminuir el costo, se prepara la matriz en una forma simple, en donde la factorización QR es mucho menos cara.

Si  $A$  es una matriz simétrica, esta se puede reducir a una matriz simétrica tridiagonal. Si  $A$  es una matriz no simétrica, esta se puede reducir a una matriz similar de Hessenberg.

Si esta es una matriz triangular superior excepto para un elemento de la subdiagonal diferente de cero. La matriz  $A$  es reducida a la forma Hessenberg usando el mismo algoritmo como fue usado para reducir una matriz simétrica a una forma tridiagonal.

Ahora, cuando la matriz  $A$  es tridiagonal o Hessenberg, el método matricial de Householder toma una forma simple cuando calculamos la factorización  $QR$ . Solo las rotaciones en el plano son usadas en lugar del método matricial de Householder porque estas son un poco más eficientes para calcular y aplicar en esta situación. Habiendo efectuado  $A_1 = QR_1$  y  $A_2 = R_1 Q_1$ , se tiene conocida la forma de  $A_2$  que es la misma de  $A_1$  en orden a continuar usando la forma menos cara de la factorización  $QR$ .

Supóngase que  $A_1$  tiene la forma Hessenberg, la factorización  $A_1 = QR_1$  con el siguiente valor para  $Q_1$ :

$$Q_1 = H_1 \dots H_{n-1}$$

en donde cada  $H_k$  es una matriz de Householder.

$$H_k = I - 2W^{(k)}W^{(k)T} \quad 1 \leq k \leq n-1$$

Porque la matriz  $A_1$  tiene la forma de Hessenberg, los vectores  $W^{(k)}$  puede demostrarse que tienen la forma especial.

$$W_i^{(k)} = 0 \quad \text{para} \quad i < k \text{ e } i > k+1 \quad (3.5.4)$$

De la ec(3.5.4), la matriz  $H_k$  difiere de la identidad solo en que los cuatro elementos en posición  $(k,k)$ ,  $(k,k+1)$ ,  $(k+1,k)$  y  $(k+1,k+1)$ . Otra cosa importante es que el producto de una matriz triangular y una matriz Hessenberg es una matriz Hessenberg. Justo multiplicando las dos formas de matrices, observando los respectivos modelos de ceros.

Combinando estos resultados se observa que  $R_1$  es una matriz triangular superior, se tiene que  $A_2 = R_1 Q_1$  esta deberá tener la forma de Hessenberg. Del resultado anterior,  $A_2$  también deberá ser Hessenberg. Solo entonces  $A_2$  es simétrica.

$$A_2^T = (Q_1^T A_1 Q_1)^T = Q_1^T A_1^T Q_1 = Q_1^T A_1 Q_1^T = A_2$$

Ahora si cualquier matriz simétrica de Hessenberg es tridiagonal, se demuestra que  $A_2$  es tridiagonal.

**3.5.1 Método QR con cambios.** El algoritmo se aplica generalmente con un cambio de origen para los valores característicos para incrementar la velocidad de convergencia, (refs 5 y 6).

Por una secuencia de constantes  $\{C_m\}$ , se define  $A_1 = A$  y

$$\begin{aligned} A_m - C_m I &= Q_m R_m \\ A_{m+1} &= C_m I + R_m Q_m \quad m=1, 2, \dots \end{aligned}$$

Las matrices  $A_m$  son similares a  $A_1$ , después

$$\begin{aligned} R_m &= Q_m^T (A_m - C_m I) \\ A_{m+1} &= C_m I + Q_m^T (A_m - C_m I) Q_m \\ &= C_m I + Q_m^T A_m Q_m - C_m I \\ A_{m+1} &= Q_m^T A_m Q_m \quad m \geq 1 \end{aligned}$$

Los valores característicos de  $A_{m+1}$  son los mismos que los de  $A_m$ , y entonces los mismos que los de  $A$ .

Para ser más específico en la elección del cambio  $\{C_m\}$ , se considera únicamente una matriz tridiagonal simétrica  $A$ . Para  $A_m$  se permite

$$A_m = \begin{bmatrix} \alpha_n^{(m)} & B_{n-1}^{(m)} & 0 & \dots & 0 \\ B_{n-1}^{(m)} & \alpha_{n-1}^{(m)} & B_{n-2}^{(m)} & & \\ 0 & \dots & \dots & \dots & \\ \vdots & & & & \\ 0 & \dots & \dots & B_{n,j}^{(m)} & \dots \end{bmatrix}$$

Hay dos métodos para elegir  $\{C_m\}$ , uno consiste en hacer  $C_m = \alpha_n^{(m)}$  y que  $C_m$  sea el valor característico de.

$$\begin{bmatrix} \alpha_{n-1}^{(m)} & B_{n-1}^{(m)} \\ B_{n-1}^{(m)} & \alpha_n^{(m)} \end{bmatrix} \quad (3.5.1.1)$$

El cual es cercano a  $\alpha_n^{(m)}$ . La segunda estrategia es preferible; solo en algún caso las matrices  $A_m$  convergen a una matriz diagonal en bloque en donde los bloques tienen orden 1 o 2. En uno u otro caso la elección de  $\{C_m\}$  garantiza que

$$B_{n-1}^{(m)} B_{n-2}^{(m)} \rightarrow 0 \quad \text{cuando} \quad m \rightarrow \infty \quad (3.5.1.2)$$

generalmente con mucho más velocidad que con el método original QR.

$$\|A_{m+1}\|_2 = \|Q_n^T A_m Q_n\|_2 = \|A_m\|_2$$

De la ec(3.5.1.2) de  $\{B_{n-1}^{(m)}\}$  y  $\{B_{n-2}^{(m)}\}$ , se tiene que cualquiera  $B_{n-1}^{(m)} \rightarrow 0$  o  $B_{n-2}^{(m)} \rightarrow 0$  cuando  $m \rightarrow \infty$ . En la mayoría de los casos,  $\alpha_n^{(m)}$  converge a un valor característico de  $A$ , y en el otro caso similar, dos valores característicos pueden obtenerse fácilmente de la submatriz en la ec(3.5.1.1).

El método QR con cambios puede aplicarse a la matriz reducida. La elección del cambio es ayudar a hacer la convergencia más rápida a cero de manera que sea más rápida para  $B_{n-1}^{(m)} B_{n-2}^{(m)}$  que para

los restantes elementos fuera de la diagonal de la matriz. De esta manera el método QR se convierte en un método general rápido, más rápido que cualquier otro método en el presente tiempo. Para la prueba de convergencia del método QR con cambios, ver Wilkinson, (refs 5 y 6).

### 3.6 Método de Lanczos

El método de Lanczos se utiliza para transformar el problema general de valores característicos como.

$$AX = \lambda BX \quad (3.6.1)$$

En donde  $A$  y  $B$  son matrices reales y simétricas,  $B$  es definida positiva, el vector columna  $X$  representa un vector característico y  $\lambda$  es su correspondiente valor característico, (ref 7).

El método se basa en suponer que una matriz de transformación  $R$  que no tiene significado único, puede aplicarse a las matrices  $A$  y  $B$  teniendo la siguiente transformación.

$$R^T A R = T \quad (3.6.2)$$

y

$$R^T B R = I \quad (3.6.3)$$

En donde  $T$  es una matriz tridiagonal simétrica,  $I$  es la matriz identidad. En la ec(3.6.3) tenemos que la matriz  $R$  es ortogonal con respecto a  $B$ . Las relaciones en las ecs(3.6.2) y (3.6.3) se usan para transformar la ec(3.6.1) en la siguiente igualdad.

$$Ty = \lambda y \quad (3.6.4)$$



Las expresiones para  $\beta_i$  y el vector  $r_{i+1}$  son necesarias para completar el algoritmo. Arreglando los términos en la ec(3.6.9) dan

$$\beta_i B r_{i+1} = v_i \quad (3.6.12)$$

donde

$$v_i = A r_i - \beta_{i-1} B r_{i-1} - \alpha_i B r_i, \quad (3.6.13)$$

Si  $\beta_0$  toma el valor cero, entonces  $v_i$  se convierte en un vector conocido para todos los pasos, porque todos los términos en la ec (3.6.13) están definidos. Entonces  $B$  es definida-positiva, y puede factorizarse como producto de una matriz triangular inferior y una superior de la siguiente forma.

$$B = U^T U \quad (3.6.14)$$

Sustituyendo la ec(3.6.14) en la ec(3.6.12) se tiene.

$$\beta_i U^T U r_{i+1} = v_i \quad (3.6.15)$$

si se define  $z_i$  como

$$z_i = \beta_i U r_{i+1} \quad (3.6.16)$$

entonces las ecs(3.6.15) y (3.6.16) dan una solución para  $z_i$  a través de un proceso de eliminación frontal, representado por:

$$U^T z_i = v_i \quad (3.6.17)$$

Para calcular  $\beta_i$ , se considera el resultado del producto interno de  $z_i$  con su transpuesta.

$$z_i^T z_i = \beta_i^2 r_{i+1}^T U^T U r_{i+1} = \beta_i^2 \quad (3.6.18)$$

Así,  $\beta_i$  se calcula como:

$$\beta_i = (z_i^T z_i)^{1/2} \quad (3.6.19)$$

Dividiendo la ec(3.6.16) entre  $\beta_i$  se obtiene una expresión de

donde  $r_{i+1}$  puede ser obtenida por un proceso de sustitución regresiva, dado por.

$$U r_{i+1} = \frac{1}{\beta_i} z_i \quad (3.6.20)$$

El proceso se repite hasta que todos los elementos de  $T$  y los vectores de  $R$  son calculados. Entonces los vectores y valores característicos del problema se evalúan con el algoritmo QR.

Se debe hacer notar que el algoritmo descrito anteriormente falla siempre que  $\beta_i = 0$  (como se ve en la ec(3.6.20). En tal caso, el vector  $r_{i+1}$  se toma arbitrariamente con:

$$r_{i+1} = \{0, 0, \dots, 1/\sqrt{\beta_{i,i}}, \dots, 0\}$$

Además, la ortogonalidad de los vectores en  $R$  tiende a deteriorarse con cada paso; de manera que estos vectores serán ortogonalizados de nuevo con respecto a  $B$  después de cada nueva evaluación.

### 3.7 El Método de Iteración Inversa de Householder - QR

Una importante técnica de solución de transformación es el método de Iteración Inversa de Householder - QR, aunque este procedimiento está restringido a la solución del problema estándar de valores característicos. Ahora si se considera el problema de valores característicos en la forma  $K\phi = \lambda M\phi$ , se deberá transformar primero el problema a la forma estándar antes de aplicar el método de solución. Cabe aclarar que esta transformación es efectiva en algunos casos, (ref 2).

En esta parte, se considera el problema  $K\phi = \lambda\phi$ , en donde  $K$  puede tener valores característicos cero (y también los puede tener ne-

gativos). Este método de solución está compuesto de los siguientes tres pasos de solución.

1. Transformación de Householder que se emplea para reducir la matriz  $K$  a una forma tridiagonal.
2. Iteración QR para calcular todos los valores característicos de la matriz tridiagonal.
3. Usando Iteración Inversa, se calculan los vectores característicos de la matriz tridiagonal y se transforman para obtener los valores característicos de  $K$ .

Una diferencia básica con el método de solución de Jacobi es que la matriz  $K$  se transforma primero a una forma tridiagonal sin efectuar iteraciones. Entonces esta matriz puede ser usada en la solución iterativa QR, en la cual se calculan todos los valores característicos, la transformación de  $K$  a una forma tridiagonal requiere más operaciones numéricas.

3.7.1 La Reducción de Householder. La reducción de Householder a una forma tridiagonal involucra  $(n-2)$  transformaciones de la forma:

$$\begin{aligned} K_2 &= P_1^T K_1 P_1 \\ K_3 &= P_2^T K_2 P_2 \\ &\vdots \\ K_{n-1} &= P_{n-2}^T K_{n-2} P_{n-2} \end{aligned} \quad (3.7.1.1)$$

usando  $K_j = K$ , se calcula

$$K_{k+1} = P_k^T K_k P_k; \quad k=1, \dots, n-2 \quad (3.7.1.2)$$

donde las  $P_k$  son matrices de transformación de Householder, que están dadas por

$$P_k = I - \theta W_k W_k^T \quad (3.7.1.3)$$

con 
$$\theta = \frac{2}{W_K^T W_K} \quad (3.7.1.4)$$

Para mostrar cómo se calcula el vector  $W_K$  el cual define la matriz  $P_K$ , se considera el caso  $K=1$ , que es típico. Primero se hace una partición de  $K_1, P_1$ , y  $W_1$  en submatrices como se muestra:

$$P = \left[ \begin{array}{c|c} 1 & 0 \\ \hline 0 & \bar{P} \end{array} \right]; \quad W_1 = \left[ \begin{array}{c} 0 \\ \hline \bar{W}_1 \end{array} \right] \quad (3.7.1.5)$$

Donde  $K_{11}, \bar{P}_1$  y  $\bar{W}_1$  son de orden  $(n-1)$ . En el caso general para  $K$  se tienen las matrices correspondientes de orden  $(n-K)$ . Ejecutando la multiplicación de la ec(3.7.1.2) se obtiene

$$K_2 = \left[ \begin{array}{c|c} K_{11} & -k_1^T \bar{P} \\ \hline \bar{P}^T k_1 & \bar{P}^T K_{11} \bar{P} \end{array} \right] \quad (3.7.1.6)$$

La condición ahora es que la primera columna y renglón de  $K_2$  esté en forma tridiagonal, es necesario que  $K_2$  sea de la forma.

$$K_2 = \left[ \begin{array}{c|cccc} K_{11} \times & 0 & \dots & 0 \\ \hline x & & & \\ 0 & & & \\ \vdots & & & \\ 0 & & & \end{array} \right] \quad (3.7.1.6a)$$

donde  $X$  indica un valor diferente de cero, y

$$\bar{K}_2 = \bar{P}^T K_{11} \bar{P} \quad (3.7.1.7)$$

La forma de  $K_2$  en la ec(3.7.1.6) se lleva con  $\bar{P}_1$  que es una matriz de reflexión. Sin embargo, se puede usar  $\bar{P}_1$  para reflejar el vector  $k_1$  de  $K_1$  en (3.7.1.5) en un vector que tiene únicamente la

primera componente diferente de cero. El nuevo vector tendrá el tamaño de  $k_1$ ,  $\bar{W}_1$  se determina de la condición.

$$(I - \theta \bar{W}_1 \bar{W}_1^T) k_1 = \pm \|k\|_2 e_1 \quad (3.7.1.8)$$

donde  $e_1$  es un vector unitario de dimensión  $(n-1)$ ;  $e_1^T = [1, 0, \dots, 0]$ , y el signo  $+$  ó  $-$  puede seleccionarse para obtenerse la mejor estabilidad numérica. Se debe notar que únicamente se necesita resolver para un múltiplo de  $\bar{W}_1$ . Se obtiene de la ec(3.7.1.8) un valor apropiado para  $\bar{W}_1$ ,

$$\bar{W}_1 = k_1 + \text{sign}(\kappa_{21}) \|k_1\|_2 e_1 \quad (3.7.1.9)$$

donde  $\kappa_{21}$  es el elemento  $(2,1)$  de  $K_1$ .

Con  $\bar{W}_1$  definida en la ec(3.7.1.9), la primera transformación de Householder,  $K=1$  en la ec(3.7.1.2) puede llevarse fuera. En el siguiente paso,  $K=2$ , se considera la matriz  $K_2$  en la ec(3.7.1.6) en la misma forma como se considera  $K_1$  en las ecs(3.7.1.5) a (3.7.1.9), porque la reducción de la primera columna y renglón de  $\bar{K}_2$  no afecta al primer renglón y columna en  $K_2$ .

**3.7.2 Iteración QR.** En el procedimiento de solución IHQR, la iteración QR se aplica a la matriz tridiagonal obtenida por la transformación de Householder de  $K$ . Sin embargo, se debe notar que la iteración QR puede ser aplicada a la matriz  $K$  con buenos resultados y que la transformación de  $K$  a una forma tridiagonal se hace para mejorar la eficiencia en la solución.

En lo siguiente se considera la iteración aplicada a una matriz simétrica  $K$ .

El paso básico en la iteración es descomponer  $K$  en la forma

$$K = QR \quad (3.7.2.1)$$

Donde  $Q$  es una matriz ortogonal y  $R$  es una matriz triangular superior. Se puede tener la siguiente forma.

$$RQ = Q^T K Q \quad (3.7.2.2)$$

Para calcular  $RQ$  se hace una transformación de la forma mostrada en la ec(3.7.1.1)

La factorización en la ec(3.7.2.1) se puede obtener aplicando el proceso de Gram-Schmidt a las columnas de  $K$ . En la práctica es más efectivo reducir  $K$  a una matriz superior usando la rotación de matrices de Jacobi, que se evalúa como

$$P_{n,n-1}^T \dots P_{3,1}^T P_{2,1}^T K = R \quad (3.7.2.3)$$

usando la ec(3.7.2.3), se tiene correspondiente a la ec(3.7.2.1)

$$Q = P_{2,1} P_{3,1} \dots P_{n,n-1} \quad (3.7.2.4)$$

El algoritmo de iteración  $QR$  se obtiene repitiendo el proceso dado en las ecs(3.7.2.1) y (3.7.2.2) usando la notación  $K_j = K$ , se forma.

$$K_k = Q_k R_k \quad (3.7.2.5)$$

y entonces

$$K_{k+1} = R_k Q_k \quad (3.7.2.6)$$

Donde pasando por alto, que los valores y vectores característicos no pueden estar en el orden usual.

$$K_{k+1} \xrightarrow{\infty} \Lambda \quad Q_1 \dots Q_k \xrightarrow{\infty} \Phi \quad \text{cuando } K \xrightarrow{\infty}$$

La relación de forma entre el método de solución  $QR$  y simple iteración inversa sugiere que es posible una aceleración de convergencia en la iteración  $QR$  descrita en las ecs(3.7.2.5) y (3.7.2.6). Esto es verdad y en la práctica la iteración  $QR$  es usada con cambios.

En la práctica la iteración QR se aplica después de reducir la matriz  $K$  a una forma tridiagonal usando las matrices de transformación de Householder, la solución QR será aplicada a la matriz  $K_{n-1}$  en la ec(3.7.1.2), la cual se llama ahora  $T_1$ .

Cuando la matriz es tridiagonal, el procedimiento QR es muy efectivo, por experiencia, se requiere de  $9n^2$  operaciones para encontrar todos los valores característicos.

3.7.3 Cálculo de los vectores característicos. Una vez que se han calculado los valores característicos, se calculan únicamente los vectores característicos requeridos de la matriz tridiagonal  $T_1$  por simple iteración inversa con cambios iguales a los correspondientes valores característicos. Usualmente son suficientes dos pasos de iteración inversa comenzando con un vector unitario lleno. Los vectores característicos de  $T_1$  entonces necesitan ser transformados con las transformaciones de Householder usadas para obtener los vectores característicos de  $K$ ; denotando el  $k$ -ésimo vector característico de  $T_1$  para  $\psi$ , se tiene, usando las matrices de transformación  $P_k$  en la ec(3.7.1.2)

$$\phi_k = P_1 P_2 \dots P_{n-1} \psi_k \quad (3.7.3.1)$$

Se debe notar que gran parte del número total de operaciones son usadas para la transformación en la ec(3.7.1.2) y, en muchos casos no necesitan calcular vectores característicos, por las transformaciones de vectores característicos en la ec(3.7.3.1). No obstante se nota que el cálculo de los valores característicos de  $T_1$  no es muy costoso, ya que lo requiere la reducción de  $K$  en una forma

que pueda ser usada efectivamente por el proceso iterativo requiere más esfuerzo numérico.

### 3.8 Comentario a la solución de sistemas grandes.

Los métodos de solución de sistemas grandes se basan en algunos factores fundamentales vistos antes. Considerando la efectividad en el procedimiento de solución, ninguno de los métodos es siempre más eficiente, la técnica de solución a ser usada se escogerá de acuerdo al problema específico a ser resuelto.

Ahora una solución eficiente de los valores y vectores característicos es de mucha importancia en análisis de superposición modal de sistemas grandes de elemento finito. Los procedimientos para la solución exacta de problemas grandes de valores característicos han sido costosos por mucho tiempo, por lo que se han tenido que emplear técnicas aproximadas de solución y estas son usadas tranquilamente. Ahora las técnicas aproximadas de solución deben ser usadas con cuidado y no pueden recomendarse para uso general.

Es importante emplear un algoritmo óptimo para el cálculo de los valores y vectores característicos requeridos para el problema

$K\phi = \lambda M\phi$  cuando el orden de  $K$  y  $M$  es grande, es natural intentar combinar los métodos de solución básicos vistos antes en un esfuerzo para diseñar técnicas más efectivas. Los métodos a ser diseñados tomarán ventajas específicas de la solución requerida y explotan las ventajas de cada procedimiento de solución básico que toma en cuenta las características específicas de  $K$  y  $M$ .

Existen dos métodos los cuales son efectivos en análisis de elemento finito, ellos son: el método de búsqueda del determinante y el método de iteración de subespacios. Las técnicas han sido extensamente usadas en programas de análisis de elemento finito.

Como se define un problema grande de valores característicos, en problemas de análisis de elemento finito solo interesan los valores característicos más pequeños. Correspondientemente, se define un problema de valores característicos como grande si este se abarata mucho si únicamente se resuelve para los valores y vectores característicos requeridos en lugar de calcular todos los valores y vectores característicos. En general, el sistema será grande si la memoria de una computadora de tamaño razonable es pequeña para usar el método de iteración inversa-Householder-QR o la técnica de solución generalizada de Jacobi.

En tal caso es probablemente más efectivo emplear, el método de búsqueda del determinante o el método de iteración de subespacios, porque la solución fuera de la memoria de alta velocidad usando métodos de transformación que calculan todos los valores característicos requiere mucho almacenamiento en memoria y son caros. Sin embargo, para una solución en memoria de alta velocidad, el método de búsqueda del determinante y el método de iteración de subespacios son más efectivos que el IHQR o el método de solución de Jacobi, porque aquellos calculan directamente (sin una transformación al problema estándar de valores característicos) solo los valores y correspondientes vectores característicos que se requieren. En suma, durante el proceso de solución se toma ventaja de las características especiales de las matrices  $K$  y  $M$ .

El método de búsqueda del determinante y el método de iteración de subespacios han sido desarrollados para la solución de los  $P$  mejores valores característicos y correspondientes vectores característicos, solo usando cambios en las estrategias, las técnicas pueden ser modificadas para calcular el mayor valor y correspondiente vector característico como se requiere en análisis de pandeo. En resumen, el método de búsqueda del determinante se puede emplear para una solución en memoria de alta velocidad de pequeños sistemas en banda, mientras que el método de iteración de subespacios es más eficiente en análisis de sistemas grandes en banda.

#### 4. CONVERGENCIA

El método más común para la convergencia de vectores característicos es el comparar la diferencia de los elementos correspondientes de sucesivos vectores arbitrarios, cada uno de los cuales ha sido normalizado de tal manera que el mayor de los elementos es la unidad, para un número pequeño arbitrario se tiene

$$|U_{n,j} - U_{n-1,j}| < \epsilon \text{ para toda } j$$

El método no puede ser satisfactorio si varios de los elementos son de diferente orden de magnitud debido a los efectos de escala. Por ejemplo, el ángulo de rotación puede ser mucho más pequeño que el de traslación. Una manera sencilla de evitar esta dificultad es modificar el criterio de convergencia como sigue

$$\left| \frac{U_{n,j} - U_{n-1,j}}{U_{n-1,j}} \right| < \epsilon$$

Sin embargo, este método puede causar problemas cuando uno de los elementos es cero o cercano a cero. En tal caso la diferencia entre sucesivos valores arbitrarios de los elementos con valor cero puede dar la impresión de que se trata de un elemento pequeño en magnitud. No obstante, será un valor grande cuando sea dividido entre  $U_{n-1,j}$ , creando la impresión de error, parecerá que la iteración no converge, sin embargo el criterio de convergencia puede ser satisfecho por otros elementos del vector. Este problema puede fácilmente ser superado saltando espacios en la prueba de convergencia, esto es cuando se tiene un elemento muy pequeño se compara con los otros elementos del vector.

Por ejemplo, si en una estructura de razonable tamaño se asume que cuando el elemento más grande del vector es normalizado a la unidad, un valor de  $10^{-6}$  en hecho representa un elemento con valor cero que no puede ser incluido en la prueba de convergencia.

## 5. APLICACIONES

La finalidad del presente trabajo es como ya se mencionó hacer una comparación entre los diferentes métodos de solución y al mismo tiempo hacer una recopilación de programas de solución tanto para el problema de valores característicos como para el problema de solución de sistemas de ecuaciones lineales, enfocados a la solución de problemas de análisis estructural, de fácil acceso en su aplicación. Para ilustrar la aplicación de las subrutinas aquí presentadas, se utilizaron cinco estructuras diferentes, cuatro marcos y una chimenea. Estas estructuras se muestran en las figs 6, 8, 10, 12a y 12b, de las cuales las estructuras de las figs 6 y 8 se usaron para el problema de solución de sistemas de ecuaciones y las estructuras de las figs 10, 12a y 12b para el problema de valores característicos.

Para mostrar las características de la matriz de rigidez y como de-

be ser almacenada antes de entrar a ser resuelta por alguna de las subrutinas, se muestra en las figs 7, 9 y 11 la matriz de rigidez de las estructuras de las figs 6, 8 y 10, ahora en las figs 3, 4 y 5 la forma en que deben ser ordenados los elementos de la matriz, ya sea en forma normal, en banda o en columna activa.

También se presentan tablas y figuras de tiempos de procesado de computadora(Bourrougs 6800), de variación del tiempo de procesado contra el número de iteraciones y tablas de frecuencias obtenidas de probar las diferentes subrutinas con los ejemplos señalados arriba.

## 6. CONCLUSIONES Y RECOMENDACIONES

En este trabajo se presentan varios algoritmos de solución para el problema de valores característicos y para el problema de solución de ecuaciones lineales. Se presentan tablas de tiempos de procesamiento en la memoria de alta velocidad de la computadora, estos tiempos se obtuvieron de probar los programas con los ejemplos 3,4 y 5 que corresponden a las figs 10, 12a y 12b para los de valores característicos y con los ejemplos 1 y 2, figs 6 y 8 para los de solución de ecuaciones lineales.

De los programas presentados de valores característicos para los ejemplos propuestos resultaron más eficientes en cuanto a tiempo de procesamiento, el basado en el método QR y el de Bisección.

Referente al problema de solución de sistemas de ecuaciones lineales se presentan varios programas basados en el método de elimina-

ción de Gauss y otros en el procedimiento de reducción de Crout, que también es bastante eficiente para el caso de matrices porosas como las provenientes de analizar sistemas estructurales.

Con respecto a los tiempos de procesado, se obtuvo que los programas basados en el método de reducción de Crout son más eficientes; sin embargo, debe tomarse en cuenta que todos los programas se usaron solo, con un programa principal que leía los datos y escribía los resultados, de los programas presentados unos resuelven el sistema en banda y otros en columna, estos últimos ocupan menos espacio para almacenar la matriz de rigidez, pero necesitan adicionalmente un vector que contiene la dirección de los elementos de la diagonal, ahora todo esto requiere un trabajo adicional que puede quitar eficiencia al programa, por la tanto dependerá del programa en el que se vaya a utilizar alguna subrutina el escoger una u otra.

Por la tanto considerando la eficiencia de los procedimientos de solución, ninguno de los métodos es siempre más eficiente, y la técnica de solución a ser usada dependerá del problema a ser resuelto, tanto para el problema de valores característicos como para el problema de solución de ecuaciones.

## 7. RECONOCIMIENTO

Se agradece al Instituto de Ingenieria de la ULAH por la facilidad de desarrollar parcialmente esta tesis durante el tiempo que trabajé en esa institución.

Al Ing. Juan Dyer por la ayuda proporcionada durante el desarrollo de la tesis.

Al Ing. Enrique Mena S. por las facilidades que me prestó durante el tiempo que trabajé en el departamento de Ingeniería Sismica.

A mis compañeros y amigos .

## 8. REFERENCIAS

1. J.S. Przemieniecki  
Theory of Matrix Structural Analysis,  
1968, McGraw-Hill, Inc.
2. K.-J. Bathe y E.L. Wilson  
Numerical Methods in Finite Element Analysis,  
Englewood Cliffs, New Jersey  
1976, Prentice-Hall, Inc.
3. D.P. Mondkar y G.H. Powell  
Towards Optimal In-Core Equation Solving,  
Computers & Structures, Vol. 4, PP. 531-548,  
Pergamon Press, 1974, Gran Bretaña.
4. J.L. Humar  
Eigenvalue Programs For Building Structures.  
Computers & Structures, Vol. 8, PP. 75-91  
Pergamon Press, 1978, Gran Bretaña.
5. G.M.L. Gladwell y U.C. Tahbildar  
The Algebraic Eigenvalue Problem  
- A Set of Fortran Subroutines,  
Solid Mechanics Division University of Waterloo  
N-. 18, Marzo 1972'
6. G.M.L. Gladwell  
The Algebraic Eigenvalue Problem:  
A Commentary on Some Computers Programs,  
Solid Mechanics Division University of Waterloo  
N-. 25, Mayo 1973.
7. W. Weaver, Jr. y D.M. Yoshida  
The Eigenvalue Problem For Banded Matrices,

Computers & Structures, Vol. 1, PP. 651-664  
Pergamon Press, 1971, Gran Bretaña

8. W. Barth, R.S. Martin y J.H. Wilkinson  
Calculation of the Eigenvalues of a Symmetric  
Tridiagonal Matrix by the Method of Bisection,  
Num. Math. 9, PP. 366-393, 1967
9. J. Dyer  
Comunicación Personal,  
Investigador de Carrera,  
Instituto de Ingeniería, UNAM, 1982
10. J. Dammy  
Apuntes de la materia Tópicos Estructurales,  
División de Estudios Superiores de la UNAM

Jacobi	Iteración Inversa	Lanczos	Bisección	RATQR
3026487	30,26487	30,26487	30,26446	30,26487
624,99581	624,99581	624,99581	624,99074	624,99580
346435094	3464,35094	3464,35094	3464,30159	3464,35093
9967,72401	9967,72401	9967,72401	9967,77602	9967,71531
19662,47368	19662,47368	19662,47368	19662,33022	19662,42106
31533,73533	31533,73416	31533,73533	31533,46037	31533,69471
42389,71849	42389,69398	42389,71849	42389,91513	42389,56298
53324,21419	53324,21416	53324,21419	53324,61543	53324,21419

Tabla 1. Frecuencias  $(\frac{E}{E_0})$  del problema de la fig 10

Jacobi	Lanczos	Bisección	RATQR
25,51773	25,51773	25,53550	25,53568
96,53106	96,53106	96,56470	96,56532
302,63041	302,63041	302,83170	302,83049
508,73441	508,73441	508,89551	508,90181
778,53070	778,53070	779,05096	779,05182
1125,30428	1125,30428	1125,65315	1125,66497
1333,82309	1333,82309	1333,65278	1334,63310
1829,65020	1829,65020	1829,71830	1829,72948
2184,34296	2184,34296	2186,10097	2186,10298
2908,92073	2908,92073	2908,67998	2908,92082

Tabla 2. Frecuencias  $(\frac{E}{E_0})$  del problema de la fig 12a

Jacobi	Lanczos	Bisección	RATQR
4.43307	4.43307	4.43311	4.43307
20.02890	20.02655	20.02677	20.02647
45.96047	45.96278	45.96339	45.96278
90.70947	90.70947	90.71026	90.70870
142.55100	142.55103	142.55229	142.55021
200.49618	200.49618	200.49635	200.49600
283.78950	283.78950	283.79064	283.78949
374.67031	374.67031	374.67403	374.67029
462.96685	462.96685	462.96683	462.96571
583.68560	583.68560	583.68603	583.68554
:	:	:	:
:	:	:	:
:	:	:	:
:	:	:	:
:	:	:	:
:	:	:	:
:	:	:	:
5600.39053	5600.39053	5600.32098	5600.36359
7327.84429	7327.84429	7327.84501	7327.84317
10833.07204	10833.07204	10833.06082	10832.96940
22311.48748	22311.48748	22311.24716	22311.48748

Tabla 3. Frecuencias  del problema de la fig 12b

Ejemplo	1	2
Método	Matriz de 14x14	27x27
Cholesky G. (ref 1)	0.3333	1.8666
Cholesky S. (ref 1)	0.3000	1.5166
GAUBAN (ref 9)	0.0333	0.1500
SISTRI (ref 10)	0.0500	0.2667
NUMSYM (ref 3)	0.0333	0.1500
COLSOL (ref 2)	0.1333	0.2500
OPTSOL (ref 3)	0.0500	0.1333

Tabla 4. Comparación de tiempo de procesamiento de computadora (Bourroughs 6800) entre los diferentes programas de solución de sistemas de ecuaciones para los problemas de las figs 6 y 8

Ejemplo	3	3	4	4	5	5
Método	$N=14$ $\sqrt{0.14}$ 3	8	3	10	5	30
Jacobi $N=5$		0.6333		0.9666		24.6166
Iteración Inv. 50 iteraciones	0.7833	2.2166				
Lanczos	0.3833	0.6666	0.5333	1.1500	4.5166	10.7333
Bisección	0.1833	0.3000	0.2333	0.4833	2.8333	5.9333
RATQR	0.1500	0.2166	0.2166	0.3500	2.7666	5.9500

Tabla 5. Comparación de tiempo de procesamiento de computadora (Bourroughs 6800) entre los diferentes métodos de solución del problema de valores y vectores característicos figs 10 y 12

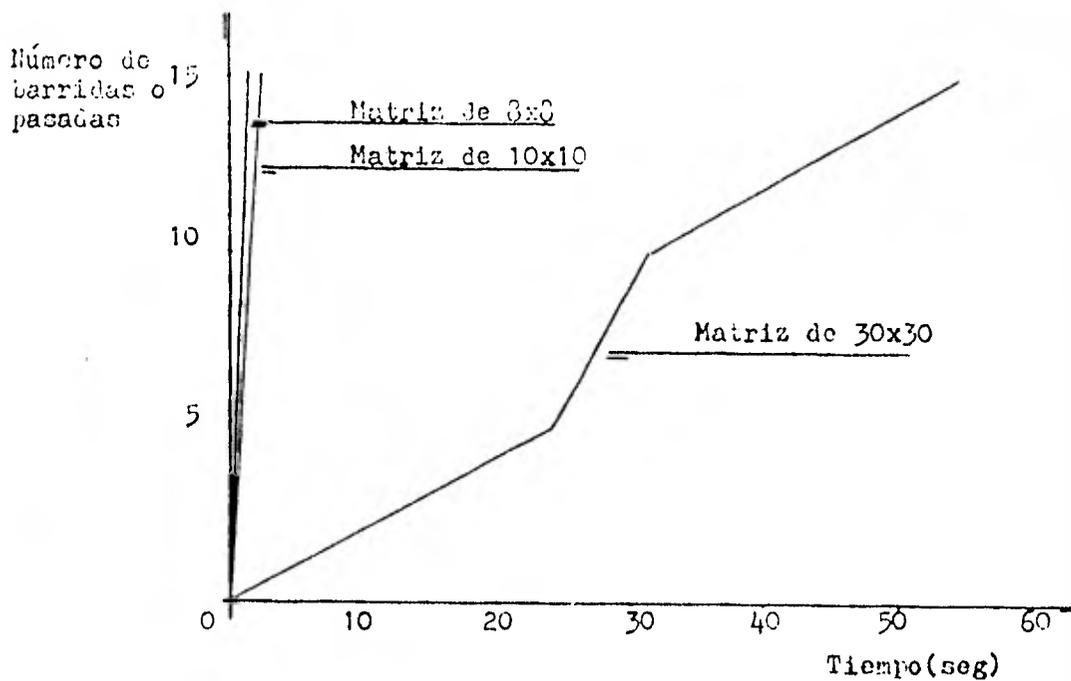


Fig. 1 Se muestra cómo varía el tiempo con el número de pasadas (barridas) a la matriz de rigidez, por el método de Jacobi, ejemplos 3, 4, 5.

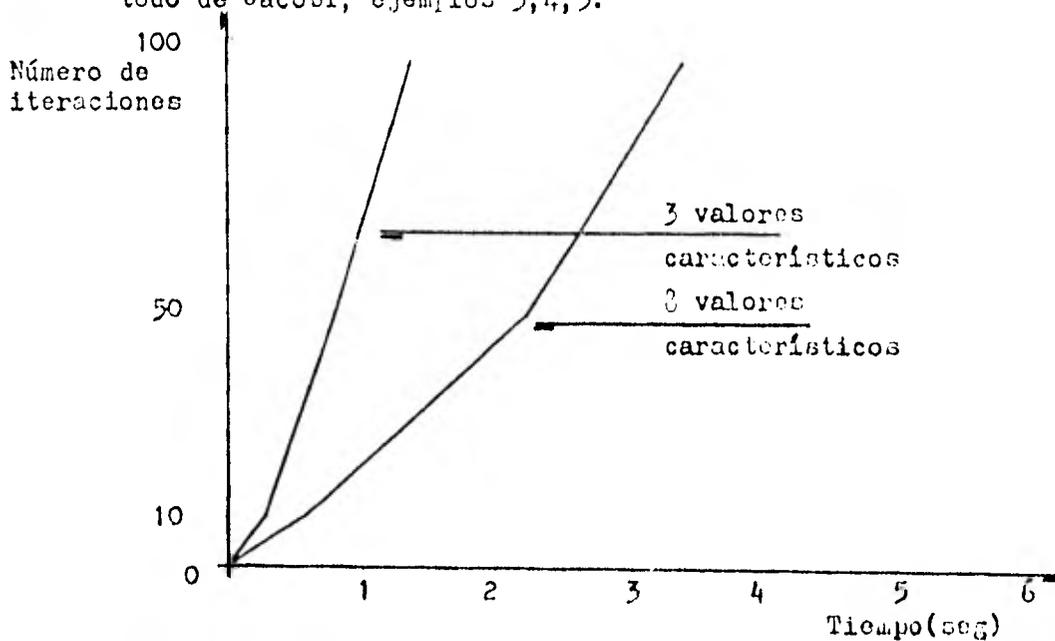


Fig. 2 Se muestra cómo aumenta el tiempo con el número de iteraciones para el programa de Iteración Inversa, para el ejemplo 3

Esquemas de Almacenaje.

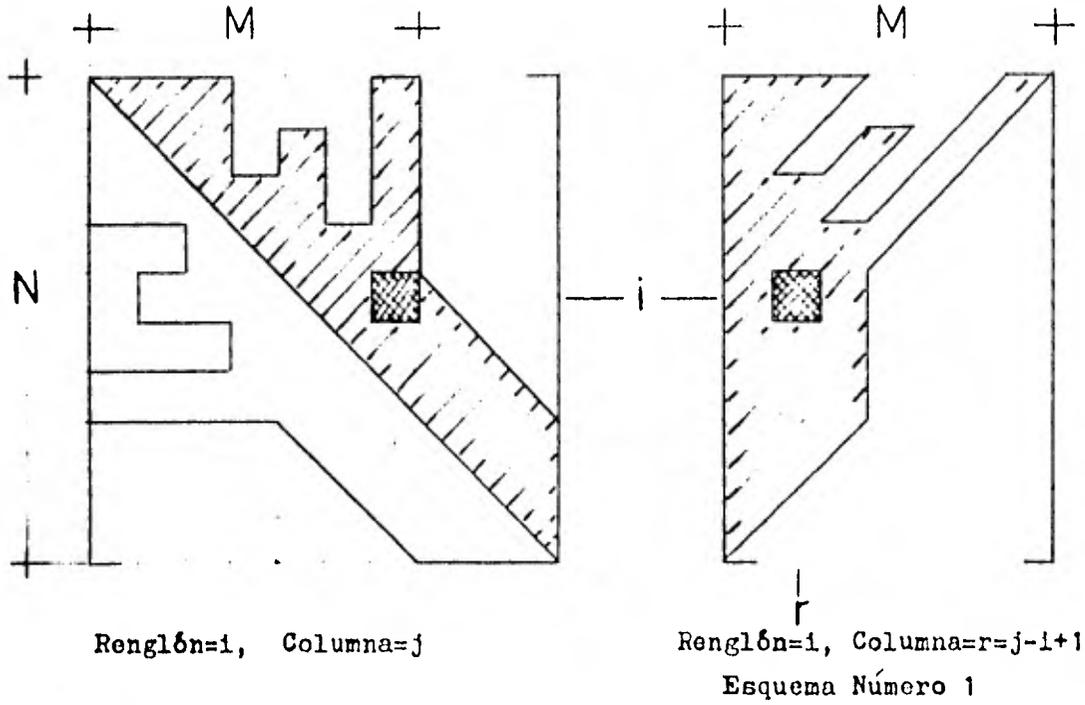


Fig. 3 Esquema de almacenaje normal y en banda.

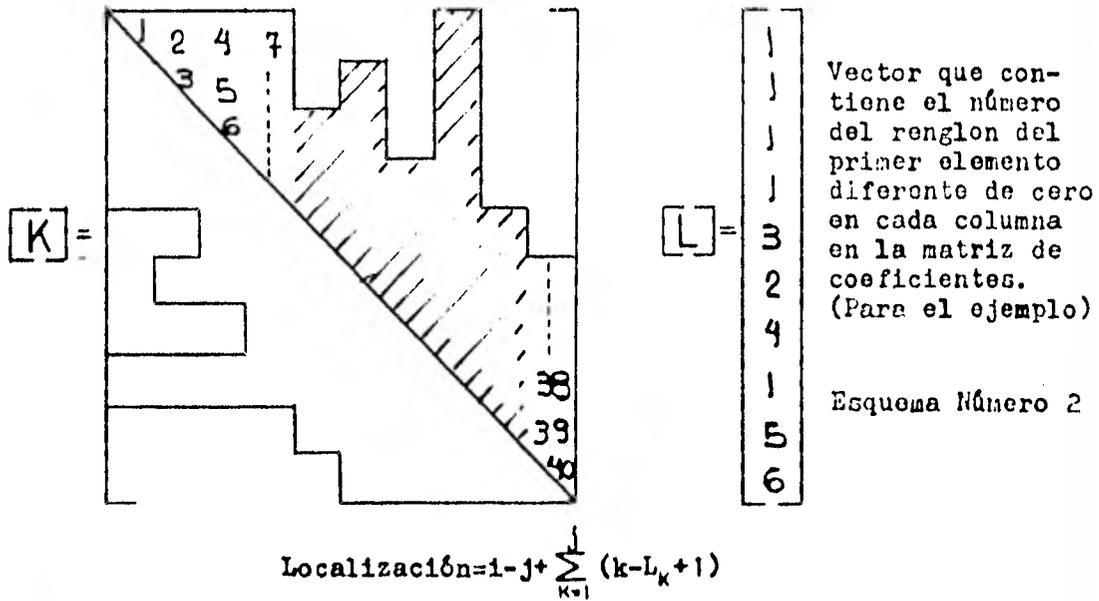


Fig. 4 Con éste esquema la matriz de coeficientes se almacena en columna activa.



$[K] =$

1.967	0.4		0.333			0.073	0.167	0.24	-0.24											
0.4	2.167	0.4		0.333		0.073	0.167	0.24		-0.24										
	0.4	1.967				0.333	0.073	0.167		0.24	-0.24									
0.333			1.467	0.4		0.167	0.167					0.24	-0.24							
	0.333		0.4	2.267	0.4	0.167	0.167					0.24		-0.24						
		0.333		0.4	1.467	0.167	0.167						0.24	0.24						
0.073	0.073	0.073	0.167	0.167	0.167	0.237	0.167													
0.167	0.167	0.167	0.167	0.167	0.167	0.167	0.167													
0.24	0.24							14.096	0.096											-8.0
0.24		0.24						0.096	14.192	0.096										-8.0
	0.24	0.24							0.096	14.096										-8.0
			0.24	0.24							-8.0									8.096
			0.24		0.24							-8.0								8.196
				0.24	0.24								-8.0							8.096

$[ton/m]$

Matriz de Rigidez

$[F] =$	0.0	Vector de cargas $[ton]$	$[D] =$	-20.686	Vector de desplazamientos $[m]$
	0.0			-33.195	
	0.0			-20.686	
	0.0			-29.955	
	0.0			-17.619	
	0.0			-29.955	
	20.0			954.075	
	30.0			1384.82	
	0.0			6.0	
	0.0			0.0	
	0.0			-6.0	
	0.0			7.344	
	0.0			0.0	
0.0	-7.344				

Fig. 7 Matriz de rigidez resultante de considerar tres grados de libertad por nudo, las trales no se deformen bajo carga axial.

### Ejemplo 2

Todas las barras son de sección de 50x60 cm.

$$\text{Area} = 0.30 \text{ m}^2$$

$$\text{Inercia} = 9 \times 10^3 \text{ m}^4$$

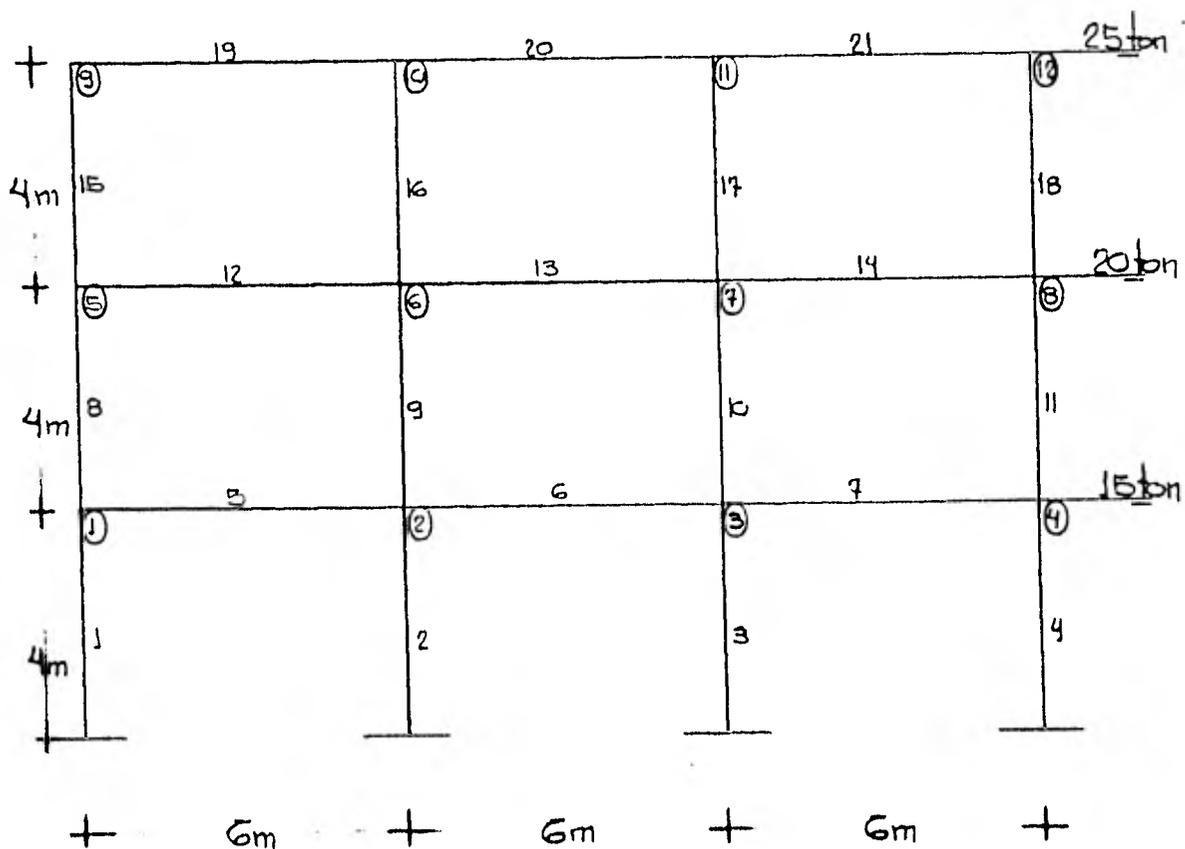


Fig. 8 Marco a ser analizado, se consideran tres grados de libertad por nudo, las trabes no se deforman bajo carga axial.

2.667	0.334			0.50						0.375	0.167	0.167								
0.334	3.334	0.334		0.50						0.375	0.167	0.167								
	0.334	3.467	0.40		0.50					0.375		0.167	0.073	0.24						
		0.40	2.80			0.50				0.375			0.24	0.24						
0.50				2.667	0.334		0.50			0.375	0.375						0.167	0.167		
	0.50			0.334	3.334	0.334		0.50		0.375	0.375						0.167	0.167		
		0.50		0.334	3.467	0.40		0.50		0.375	0.375						0.167	0.073	0.24	
			0.50		0.40	2.80			0.50	0.375	0.375							0.24	0.24	
				0.50			1.667	0.334			0.375	0.375								
					0.50		0.334	2.334	0.334			0.375	0.375							
						0.50		0.334	2.467	0.40			0.375	0.375						
							0.50		0.40	1.80				0.375	0.375					
								0.375	0.375	0.375	0.375		1.50	0.75						
0.375	0.375	0.375	0.375					0.375	0.375	0.375	0.375		0.75	1.50	0.75					
									0.375	0.375	0.375	0.375			0.75	0.75				
0.167	0.167												16.716	0.056					-8.33	
0.167		0.167											0.056	16.716	0.056				-8.33	
	0.167	0.073	0.24											0.056	16.81	0.056			-8.33	
		0.24	0.24												0.056	16.756			-8.33	
				0.167	0.167														16.716	0.056
				0.167		0.167													0.056	16.716
					0.167	0.073	0.24												0.056	16.81
						0.24	0.24													0.056
								0.167	0.167											-8.33
								0.167		0.167										-8.33
									0.167	0.073	0.24									-8.33
										0.24	0.24									-8.33

Fig. 9. Matriz de rigidez resultante de considerar tres grados de libertad por nudo. en el marco de la fig. 8 las traveses no se deforma bajo carga axial.



Ejemplo 3

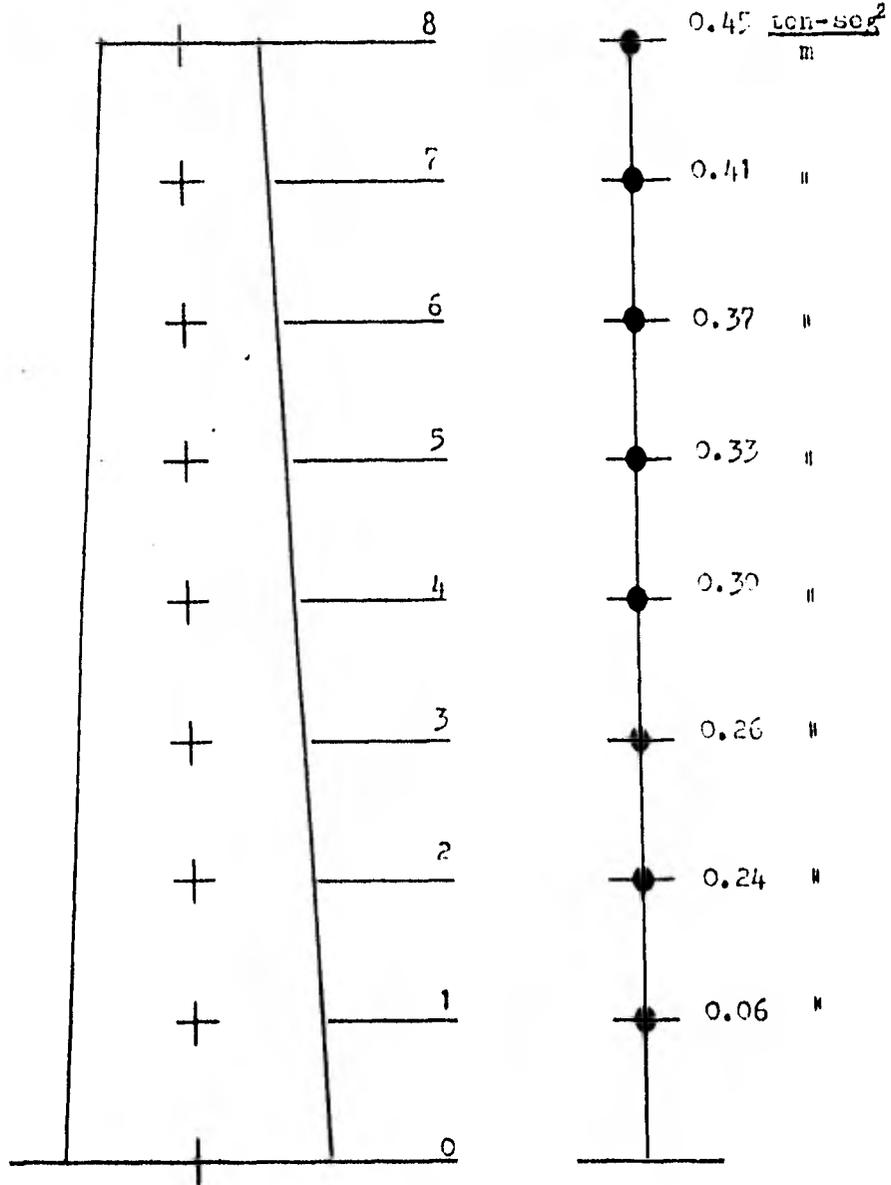


Fig. 10 Chimenea a ser analizada y concentración final de las masas.

12791.0	-6695.5	1307.8	119.41	-12.38	-7.77	20.23	-8.11	
-6695.5	9743.3	-5890.8	1122.0	83.51	-14.24	15.66	-4.43	
1307.8	-5890.8	8448.4	-5086.3	993.40	70.85	-21.02	8.45	
[K] = 119.41	1122.0	-5086.3	7159.4	-4249.9	850.29	37.60	0.94	[Ton/m]
-12.38	83.51	993.4	-4249.9	5950.0	-3519.9	724.3	20.60	
-7.77	-14.24	70.85	850.29	-3519.9	4856.8	-2865.1	615.82	
20.23	15.66	-21.02	37.60	724.30	-2865.1	3409.7	-1303.8	
-8.11	-4.43	8.45	0.94	20.60	615.82	-1303.8	664.32	

Matriz de Rigidez

[M] =	0.45	[Ton-seg <sup>2</sup> / m]
	0.41	
	0.37	
	0.33	
	0.30	
	0.26	
	0.24	
	0.06	

Vector de Masas

Fig. 11 Matriz de rigidez de la chimenea de la Fig. 10 y vector que contiene la diagonal de la matriz de masas.

Ejemplo 4 y 5

			10
			9
			8
			7
			6
			5
			4
			3
			2
			1
			0

			30
			29
			28
			27
			26
			25
			24
			23
			22
			21
			20
			19
			18
			17
			16
			15
			14
			13
			12
			11
			10
			9
			8
			7
			6
			5
			4
			3
			2
			1
			0

Figs. 12a y 12b Marcos a ser analizados, estos dos marcos serán utilizados para probar los programas que tratan el problema de valores y vectores característicos.

PROGRAMA PARA SOLUCION DE SISTEMAS DE ECUACIONES  
LINEALES POR EL METODO GENERAL DE CHOLESKY

```

2900          SUBROUTINE CHOSIM(A,X,N1,R1,U,E1,Z1,U1,U2)          00002900
3000          C C C C C C C C C C C C C C C C C C C C C C C C 00003000
3100          PROGRAMA                                          00003100
3200          RESUELVE SISTEMAS DE ECUACIONES LINEALES POR EL METODO 00003200
3300          GENERAL DE CHOLESKY.                               00003300
3400          C C C C C C C C C C C C C C C C C C C C C C C C 00003400
3500          REFERENCIA                                         00003500
3600          J.S. PRZEMJENIECKI                                 00003600
3700          "THEORY OF MATRIX STRUCTURAL ANALYSIS".           00003700
3800          DEAN OF ENGINEERING AIR FORCE INSTITUTE OF TECHNOLOGY. 00003800
3900          1968 BY MCGRAW-HILL, INC.                          00003900
4000          C C C C C C C C C C C C C C C C C C C C C C C C 00004000
4100          - - ENTRADA DE VARIABLES - -                       00004100
4200          A(N1,N1) = MATRIZ DE COEFICIENTES                 00004200
4300          X(N1)   = VECTOR DE TERMINOS INDEPENDIENTES      00004300
4400          N1      = NUMERO DE ECUACIONES                    00004400
4500          C C C C C C C C C C C C C C C C C C C C C C C C 00004500
4600          - - SALIDA - -                                       00004600
4700          R1(N1) = CONTIENE EL VECTOR SOLUCION DEL PROBLEMA 00004700
4800          C C C C C C C C C C C C C C C C C C C C C C C C 00004800
4900          DIMENSION A(N1,N1),U(N1,N1),E1(N1,N1)            00004900
5000          DIMENSION X(N1),R1(N1)                            00005000
5100          DIMENSION U1(N1,N1),U2(N1,N1),Z1(N1,N1)           00005100
5200          C C C C C C C C C C C C C C C C C C C C C C C C 00005200
5300          OBTIENE LA MATRIZ TRIANGULAR INFERIOR E1 Y LA MATRIZ TRIANGULAR 00005300
5400          SUPERIOR U                                         00005400
5500          C C C C C C C C C C C C C C C C C C C C C C C C 00005500
5600          DO 10 I=1,N1                                        00005600
5700          DO 10 J=1,N1                                        00005700
5800          IF (J.NE.I) GO TO 1550                            00005800
5900          U(I,J)=1.0                                         00005900
6000          U(I,J)=1.0                                         00006000
6100          1550 IF(J.GT.I) GO TO 1620                          00006100
6200          S8=0.                                               00006200
6300          DO 20 K=1,J-1                                        00006300
6400          S8=S8+E1(I,K)*U(K,J)                               00006400
6500          E1(I,J)=A(I,J)-S8                                  00006500
6600          GO TO 10                                           00006600
6700          1620 S9=0.                                          00006700
6800          DO 30 K1=1,J-1                                       00006800
6900          S9=S9+E1(I,K1)*U(K1,J)                             00006900
7000          U(I,J)=(A(I,J)-S9)/E1(I,I)                        00007000
7100          10 CONTINUE                                        00007100
7200          C C C C C C C C C C C C C C C C C C C C C C C C 00007200
7300          AQUI SE OBTIENE LA INVERSA DE LA MATRIZ E1        00007300
7400          C C C C C C C C C C C C C C C C C C C C C C C C 00007400
7500          DO 40 I=1,N1                                        00007500
7600          DO 40 J=1,N1                                        00007600
7700          IF (J.NE.I) GO TO 1780                            00007700
7800          Z1(I,J)=1.0/E1(I,I)                               00007800
7900          GO TO 40                                           00007900
8000          1780 IF(J.GT.I) GO TO 1850                          00008000
8100          S3=0.                                               00008100
8200          DO 70 L4=J,I-1                                       00008200
8300          S3=S3+E1(I,L4)*Z1(I,L4)                            00008300
8400          Z1(I,J)=-S3/E1(I,I)                                00008400
8500          GO TO 40                                           00008500

```

```

8600      1850 Z1(I,J)=0.
8700      40 CONTINUE
8800      C
8900      C      AQUÍ SE OBTIENE LA INVERSA DE LA MATRIZ U
9000      C
9100      DO 71 I=1,N1
9200      DO 71 J=1,N1
9300      IF(J.NE.I) GO TO 2000
9400      U1(I,J)=1.0
9500      GO TO 71
9600      2000 IF(I.GT.J) GO TO 2070
9700      S4=0.
9800      DO 72 K=I,J-1
9900      72 S4=S4+U1(I,K)*U(K,J)
10000     U1(I,J)=-S4
10100     GO TO 71
10200     2070 U1(I,J)=0.
10300     71 CONTINUE
10400     C
10500     C      AQUÍ SE MULTIPLICA LAS MATRICES INVERSAS DE E1 Y U PARA OBTENER
10600     C      LA INVERSA DE LA MATRIZ A
10700     C
10800     DO 99 I=1,N1
10900     DO 99 J=1,N1
11000     S5=0.
11100     DO 73 K=1,N1
11200     73 S5=S5+U1(I,K)*Z1(K,J)
11300     99 U2(I,J)=S5
11400     C
11500     C      AQUÍ SE MULTIPLICA LA INVERSA DE LA MATRIZ A POR EL VECTOR DE
11600     C      TERMINOS INDEPENDIENTES PARA OBTENER LA SOLUCION
11700     C
11800     DO 74 I=1,N1
11900     S7=0.
12000     DO 75 K=1,N1
12100     75 S7=S7+U2(I,K)*X(K)
12200     74 R(I)=S7
12300     RETURN
12400     END

```

```

00008600
00008700
00008800
00008900
00009000
00009100
00009200
00009300
00009400
00009500
00009600
00009700
00009800
00009900
00010000
00010100
00010200
00010300
00010400
00010500
00010600
00010700
00010800
00010900
00011000
00011100
00011200
00011300
00011400
00011500
00011600
00011700
00011800
00011900
00012000
00012100
00012200
00012300
00012400

```

```

0000000000 00 00 00
0000000000 00 000 00
00 00 0000 00
00 00 0000 00
000000 00 00 000 00
000000 00 00 000 00
00 00 00 0000
00 00 00 0000
00 00 00 0000
00 00 00 0000

```

PROGRAMA PARA SOLUCION DE SISTEMAS DE ECUACIONES  
LINEALES PARA MATRICES SIMETRICAS POR EL METODO  
DE CHOLESKY

```

2900      SUBROUTINE CHILSKY(A,X,N1,R1,E,B,C,D)
3000      C
3100      C      PROGRAMA
3200      C      RESUELVE SISTEMAS DE ECUACIONES LINEALES POR EL METODO DE
3300      C      CHOLESKY PARA MATRICES SIMETRICAS
3400      C
3500      C      REFERENCIA
3600      C      J.K.S. PRZEMIENTECKI
3700      C      "THEORY OF MATRIX STRUCTURAL ANALYSIS",
3800      C      DEAN OF ENGINEERING AIR FORCE INSTITUTE OF TECHNOLOGY,
3900      C      1968 BY MCGRAW-HILL, INC.
4000      C
4100      C      -- ENTRADA DE VARIABLES --
4200      C      A(N1,N1) = MATRIZ DE RIGIDEZ
4300      C      X(N1)   = VECTOR DE CARGAS
4400      C      N1     = NUMERO DE ECUACIONES
4500      C
4600      C      -- SALIDA --
4700      C      R1(N1) = CONTIENE EL VECTOR DE DESPLAZAMIENTOS
4800      C
4900      C
5000      C      DIMENSION A(N1,N1),E(N1,N1),B(N1,N1),C(N1,N1),X(N1),
5100      C      *R1(N1),D(N1)
5200      C
5300      C      OBTIENE LA MATRIZ TRIANGULAR SUPERIOR
5400      C
5500      DO 150 J=1,N1
5600      DO 150 K=1,N1
5700      IF (J.NE.I) GO TO 25
5800      S=0.
5900      DO 20 K=1,I-1
6000      20 S=S+A(I,K)*E(I,K)
6100      IF(A(I,J).GE.S) GO TO 85
6200      WRITE(6,1000)
6300      STOP
6400      85 E(I,I)=SQRT(A(I,I)-S)
6500      GO TO 150
6600      25 IF(J.GT.I) GO TO 140
6700      S1=0.
6800      DO 40 K1=1,J-1
6900      40 S1=S1+E(I,K1)*E(I,K1)
7000      E(I,I)=(A(I,I)-S1)/E(I,I)
7100      GO TO 150
7200      140 E(I,I)=0.
7300      150 CONTINUE
7400      C
7500      C      OBTIENE LA INVERSA DE LA MATRIZ E Y C C R
7600      C
7700      DO 50 I=1,N1
7800      DO 50 J=1,N1
7900      IF (J.NE.I) GO TO 250
8000      B(I,I)=1.0/E(I,I)
8100      GO TO 50
8200      250 IF(J.GT.I) GO TO 320
8300      S3=0.
8400      DO 60 L4=J,I-1
8500      60 S3=S3+E(I,L4)*R(L4,I)

```

```

00002200
00003000
00003100
00003200
00003300
00003400
00003500
00003600
00003700
00003800
00003900
00004000
00004100
00004200
00004300
00004400
00004500
00004600
00004700
00004800
00004900
00005000
00005100
00005200
00005300
00005400
00005500
00005600
00005700
00005800
00005900
00006000
00006100
00006200
00006300
00006400
00006500
00006600
00006700
00006800
00006900
00007000
00007100
00007200
00007300
00007400
00007500
00007600
00007700
00007800
00007900
00008000
00008100
00008200
00008300
00008400
00008500

```

```

8600      B(I,J)=-S3/E(I-1)
8700      GO TO 50
8800      B(I,J)=0.
8900      50 CONTINUE
9000
9100      C
9200      C
9300      SF MULTIPLICA B POR SU TRANSPUESTA PARA OBTENER A-1
9400      DO 70 J=1,N1
9500      DO 70 I=1,N1
9600      S0=0.
9700      DO 80 K=1,N1
9800      S0=S0+R(K,I)*R(K,J)
9900      C(I,J)=S0
10000     C
10100     C
10200     C
10300     DO 90 I=1,N1
10400     S7=0.
10500     DO 100 K=1,N1
10600     100 S7=S7+C(I,K)*X(K)
10700     90 R1(I)=S7
10800     RETURN
10900     1000 FORMAT(" SISTEMA MAL CONDICIONADO",/)
11000     END

```

```

00008700
00008800
00008900
00009000
00009100
00009200
00009300
00009400
00009500
00009600
00009700
00009800
00009900
00010000
00010100
00010200
00010300
00010400
00010500
00010600
00010700
00010800
00010900
00011000

```

```

0000000000 00 00 00
0000000000 00 00 00
00 00 00 00
00 00 00 00
000000 00 00 00
000000 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00

```

PROGRAMA PARA SOLUCION DE SISTEMAS DE ECUACIONES  
LINEALES PARA MATRICES SIMETRICAS, USANDO UN ES-  
QUEMA DE ALMACENAJE EN BANDA

```

5300      SUBROUTINE GAUBAN (A,B,NR1,NC2)
5400      C
5500      C
5600      C
5700      C
5800      C
5900      C
6000      C
6100      C
6200      C
6300      C
6400      C
6500      C
6600      C
6700      C
6800      C
6900      C
7000      C
7100      C
7200      C
7300      C
7400      C
7500      C
7600      C
7700      C
7800      C
7900      C
8000      C
8100      C
8200      C
8300      C
8400      C
8500      C
8600      C
8700      C
8800      C
8900      C
9000      C
9100      C
9200      C
9300      C
9400      C
9500      C
9600      C
9700      C
9800      C
9900      C
10000     C
10100     C
10200     C
10300     C
10400     C
10500     C
10600     C
10700     C
10800     C
10900     C

```

RESUELVE SISTEMAS DE ECUACIONES LINEALES PARA MATRICES  
 SIMETRICAS Y EN BANDA. USANDO UN ESQUEMA DE ALMACENAMIENTO  
 EN BANDA. LA MATRIZ DE RIGIDEZ ENTRA A LO SUBPROGRAMA DE  
 ACUERDO AL ESQUEMA DE ALMACENAMIENTO NUMERO 1.  
 SE UTILIZA EL METODO DE GAUSS.

REFERENCIA  
 M. EN T. HYER DE LEON JIJAN  
 INSTITUTO DE INGENIERIA UNAM.

LA LLAMADA A LA SUBROUTINA ES: PCHIA FOR  
 CALL GAUBAN(A,B,NR1,NC2)

-- ENTRADA DE VARIABLES --  
 ES(NR1,MB) = MATRIZ DE RIGIDEZ EN BANDA  
 R(NR1,NC2) = MATRIZ DE CARGAS  
 NR1 = NUMERO DE ECUACIONES  
 MB = ANCHO DE BANDA DE "A"  
 NC2 = NUMERO DE CONDICIONES DE CARGA

-- SALIDA --  
 R(NR1,NC2) = MATRIZ DE DESPLAZAMIENTOS

DIMENSION A(50,MD),B(50,NC2)  
 N1=NR1-1  
 M1=NR1-MB+1  
 JS=MD  
 DO 300 K=1,N1  
 KK=K  
 T1=A(K,J)  
 T1=1./T1  
 K1=K-1  
 IF(K.GT.M1) JS=NR1-K1  
 DO 150 J1=2,JS  
 JS=J1  
 T2=A(K,J1)\*T1  
 IF(T2.EQ.0.) GO TO 150  
 DO 100 I1=2,JS  
 JS=I1  
 I=I1+K1  
 100 A(I,J1)=A(I,J1)-A(I,I1)\*T2  
 150 CONTINUE  
 DO 250 J=1,NC2  
 T2=R(K,J)\*T1  
 IF(T2.EQ.0.) GO TO 250  
 DO 200 I1=2,JS  
 I=I1+K1  
 200 B(I,J)=B(I,J)-A(I,I1)\*T2  
 250 CONTINUE  
 300 CONTINUE  
 T1=1./A(NR1,1)  
 N1=NR1-1  
 DO 710 K=1,NC2

00005300  
 00005310  
 00005320  
 00005330  
 00005340  
 00005350  
 00005360  
 00005370  
 00005380  
 00005390  
 00005400  
 00005410  
 00005420  
 00005430  
 00005440  
 00005450  
 00005460  
 00005470  
 00005480  
 00005490  
 00005500  
 00005510  
 00005520  
 00005530  
 00005540  
 00005550  
 00005560  
 00005570  
 00005580  
 00005590  
 00005600  
 00005610  
 00005620  
 00005630  
 00005640  
 00005650  
 00005660  
 00005670  
 00005680  
 00005690  
 00005700  
 00005710  
 00005720  
 00005730  
 00005740  
 00005750  
 00005760  
 00005770  
 00005780  
 00005790  
 00005800  
 00005810  
 00005820  
 00005830  
 00005840  
 00005850  
 00005860  
 00005870  
 00005880  
 00005890  
 00005900  
 00005910  
 00005920  
 00005930  
 00005940  
 00005950  
 00005960  
 00005970  
 00005980  
 00005990  
 00010000  
 00010010  
 00010020  
 00010030  
 00010040  
 00010050  
 00010060  
 00010070  
 00010080  
 00010090

```

11000      .IS=1
11100      R(NR1,K)=R(NR1,I)*T1
11200      DO 155 II=1,N1
11300      .IS=.IS+1
11400      I=NR1-II
11500      IF (II.GE.NR) .IS=.NR
11600      SUM=B(I,K)
11700      II=I
11800      DO 110 .I=2,.IS
11900      II=II+1
12000      110 SUM=SUM+A(I,II)*B(II,K)
12100      155 R(I,K)=SUM/A(I,II)
12200      210 CONTINUE
12300      RETURN
12400      END

```

```

00011000
00011100
00011200
00011300
00011400
00011500
00011600
00011700
00011800
00011900
00012000
00012100
00012200
00012300
00012400

```

```

0000000000 00 00 00
0000000000 00 00 00
00 00 0000 00
00 00 0000 00
000000 00 00 00 00
000000 00 00 00 00
00 00 00 00 00
00 00 00 00 00
00 00 00 00

```

PROGRAMA PARA SOLUCION DE SISTEMAS DE ECUACIONES  
LINEALES PARA MATRICES SIMETRICAS, USANDO UN ES-  
QUEMA DE ALMACENAJE EN BANDA

```

5300 SUBROUTINE SISTRI (FS, KC, NR, NC, IND)
5400 C
5500 C
5600 C
5700 C
5800 C
5900 C
6000 C
6100 C
6200 C
6300 C
6400 C
6500 C
6600 C
6700 C
6800 C
6900 C
7000 C
7100 C
7200 C
7300 C
7400 C
7500 C
7600 C
7700 C
7800 C
7900 C
8000 C
8100 C
8200 C
8300 C
8400 C
8500 C
8600 C
8700 C
8800 C
8900 C
9000 C
9100 C
9200 C
9300 C
9400 C
9500 C
9600 C
9700 C
9800 C
9900 C
10000 C
10100 C
10200 C
10300 C
10400 C
10500 C
10600 C
10700 C
10800 C
10900 C

```

PROGRAMA  
RESUELVE SISTEMAS DE ECUACIONES LINEALES PARA MATRICES  
SIMETRICAS Y EN BANDA, USANDO UN ESQUEMA DE ALMACENAJE EN  
BANDA. LA MATRIZ DE RIGIDEZ ENTRA A LA SUBROUTINA DE CUERPO  
AL ESQUEMA DE ALMACENAJE NUMERO 11.

REFERENCIA  
ING. JULIO RAMY GIOS  
DIVISION DE ESTUDIOS DE POSGRADO UNAM.

LA LLAMADA ES HECHA POR  
CALL SISTRI (FS, KC, NR, NC, IND)

- ENTRADA DE VARIABLES -  
FS (NR, NC) = MATRIZ DE RIGIDEZ EN BANDA  
KC (NR, NC) = MATRIZ DE CARGAS  
NR = NUMERO DE ECUACIONES  
NC = ANCHO DE BANDA DE "RES"  
NC = NUMERO DE CONDICIONES DE CARGA  
IND = PARAMETRO DE EFICIENCIA  
(1) IND=0 TRIANGULARIZA LA MATRIZ DE RIGIDEZ EN BANDA  
Y RESUELVE EL SISTEMA DE ECUACIONES  
(2) IND=1 TRIANGULARIZA LA MATRIZ DE RIGIDEZ EN BANDA

- - SALIDA -  
KC (NR, NC) = MATRIZ QUE CONTIENE LA SOLUCION DE SISTEMA

DIMENSION ES (50, NC), KC (50, NC)  
CALL TRIAN (FS, NR, NC, 50, IND)  
IF (IND, EQ, 1) CALL EXIT  
CALL RESUEL (FS, NR, NC, 50, KC, NC, 50)  
RETURN  
END

SUBROUTINE TRIAN (A, NN, NAN, NDI, IND)

SUBROUTINA PARA TRIANGULARIZAR MATRICES EN BANDA

DIMENSION A (NDI, 1)  
IND=0  
DO 20 J=1, NN  
N1=NAN  
IF (1, GT, NN-NAN) CONTINUE  
DO 20 J=1, N1  
SUMA=0.  
K1=J-1  
K2=J  
K3=J+1  
IF (K3, EQ, 0, OR, K1, EQ, NAN) GO TO 11  
SUMA=SUMA+A (K3, K1)+A (K3, K2)  
K1=K1+1  
K2=K2+1

```

11000      K3=K3-1
11100      GO TO 10
11200      11  SUMA=A(I,J)-SUMA
11300          IF(J.NE.1)GO TO 12
11400          IF(SUMA.LE.0)GO TO 13
11500          A(I,J)=SORT(SUMA)
11600          GO TO 90
11700      12  A(I,J)=SUMA/A(I,1)
11800      90  CONTINUE
11900      RETURN
12000      13  WRITE(6,1000)I,SUMA
12100          IND=1
12200      RETURN
12300      1000 FORMAT(5X,"ERROR EN EL RENDON",I5,"  SUMA=","F15.5")
12400      END
12500
12600
12700
12800
12900      SUBROUTINE RESUEL(A,NN,NAN,ND1,YY,ND2)
13000      SUBROUTINA PARA RESOLVER UN SISTEMA DE ECUACIONES DADO EN FORMA
13100
13200      DIMENSION A(ND1,1),YY(ND2,1)
13300      DO 200 J=1,ND1
13400      DO 50 I=1,NN
13500      SUMA=0.
13600      K1=2
13700      K3=I-1
13800      10  IF(K1.EQ.NAN+1.OR.K3.EQ.0)GO TO 50
13900      SUMA=SUMA+A(K3,I)*YY(K3,J)
14000      K1=K1+1
14100      K3=K3-1
14200      GO TO 10
14300      50  YY(I,J)=(YY(I,J)-SUMA)/A(I,1)
14400      DO 100 IP=1,NN
14500      I=NN-IP+1
14600      SUMA=0.
14700      K1=2
14800      K3=I+1
14900      51  IF(K3.EQ.NN+1.OR.K1.EQ.NAN+1)GO TO 100
15000      SUMA=SUMA+A(I,1)*YY(K3,J)
15100      K1=K1+1
15200      K3=K3-1
15300      GO TO 51
15400      100  YY(I,J)=(YY(I,J)-SUMA)/A(I,1)
15500      200  CONTINUE
15600      RETURN
15700      END

```

```

0000000000 00 00 00
0000000000 00 000 00
00 00 0000 00
00 00 0000 00
000000 00 00 000 00
000000 00 000 00
00 00 00 0000
00 00 00 0000
00 00 00 000
00 00 00 00

```

PROGRAMA PARA SOLUCION DE SISTEMAS DE ECUACIONES  
LINEALES PARA MATRICES SIMETRICAS, USANDO ALMACENAJE COMPACTO Y ESQUEMA DE REDUCCION EN COLUMNA



```

0400      60 KK=MIN0(10,ND)
0500      C=0.0
0600      DO 70 L=1,KK
0700      C=C+A(KL+L)*A(KL+L)
0800      A(KL+L)=A(KL+L)-C
0900      80 K=K+1
1000      90 K=N
1100      B=0.0
1200      DO 100 KK=KL,KU
1300      K=K-1
1400      KI=MAXA(K)
1500      C=A(KK)/A(KI)
1600      B=B+C*A(KK)
1700      100 A(KK)=C
1800      A(KN)=A(KN)-B
1900      110 IF (A(KN)) 120,120,140
2000      120 WRITE(10UT,2000) N,A(KN)
2100      STOP
2200      140 CONTINUE
2300      C.C.C.
2400      REDUCE EL VECTOR DE CARGAS.
2500      C.C.C.
2600      150 DO 180 N=1,NN
2700      KI=MAXA(N)+1
2800      KU=MAXA(N+1)-1
2900      IF (KU-KI) 180,160,170
3000      160 K=N
3100      C=0.0
3200      DO 170 KK=KL,KU
3300      K=K-1
3400      170 C=C+A(KK)*V(K)
3500      V(N)=V(N)-C
3600      180 CONTINUE
3700      C.C.C.
3800      SUSTITUCION REGRESIVA
3900      C.C.C.
4000      DO 200 N=1,NN
4100      K=MAXA(N)
4200      200 V(N)=V(N)/A(K)
4300      IF (NN.EQ.1) RETURN
4400      N=NN
4500      DO 230 L=2,NN
4600      KI=MAXA(N)+1
4700      KU=MAXA(N+1)-1
4800      IF (KU-KI) 230,210,210
4900      210 K=N
5000      DO 220 KK=KL,KU
5100      K=K-1
5200      220 V(K)=V(K)-A(KK)*V(N)
5300      230 N=N-1
5400      RETURN
5500      2000 FORMAT(// "ALTO-LA MATRIZ DE RIGIDEZ NO ES DEFINIDA POSITIVA" //
5600      * "PIVOTE NEGATIVO EN LA ECUACION",I4 //
5700      * "PIVOTE=",F20.12)
5800      END

```

```

00000400
00000500
00000600
00000700
00000800
00000900
00001000
00001100
00001200
00001300
00001400
00001500
00001600
00001700
00001800
00001900
00002000
00002100
00002200
00002300
00002400
00002500
00002600
00002700
00002800
00002900
00003000
00003100
00003200
00003300
00003400
00003500
00003600
00003700
00003800
00003900
00004000
00004100
00004200
00004300
00004400
00004500
00004600
00004700
00004800
00004900
00005000
00005100
00005200
00005300
00005400
00005500
00005600
00005700
00005800
00005900
00006000
00006100
00006200
00006300
00006400
00006500
00006600
00006700
00006800
00006900
00007000
00007100
00007200
00007300
00007400
00007500
00007600
00007700
00007800
00007900
00008000
00008100
00008200
00008300
00008400
00008500
00008600
00008700
00008800
00008900
00009000
00009100
00009200
00009300
00009400
00009500
00009600
00009700
00009800
00009900
00010000
00010100
00010200
00010300
00010400
00010500
00010600
00010700
00010800
00010900
00011000
00011100
00011200
00011300
00011400
00011500
00011600
00011700
00011800
00011900
00012000
00012100
00012200
00012300
00012400
00012500
00012600
00012700
00012800
00012900
00013000
00013100
00013200
00013300
00013400
00013500
00013600
00013700
00013800
00013900
00014000

```

```

0000000000 00 00 00 00
0000000000 00 000 00
00 00 0000 00
00 00 0000 00
00000000 00 00 0000 00
00000000 00 00 0000 00
00 00 00 0000 00
00 00 00 0000 00
00 00 00 0000 00

```

PROGRAMA PARA SOLUCION DE SISTEMAS DE ECUACIONES  
LINEALES PARA MATRICES SIMETRICAS, USANDO UN ES-  
QUEMA DE ALMACENAJE EN BANDA

```

5300 SUBROUTINE NUMSYM (A,B,NA,NEQ,MBAND,KEY) 00005300
5400 00005400
5500 PROGRAM 00005500
5600 RESUELVE SISTEMAS DE ECUACIONES LINEALES PARA MATRICES SIME 00005600
5700 TRICAS USANDO ALMACENAJE EN BANDA PARA LA MATRIZ DE RIGIDEZ. 00005700
5800 LA MATRIZ DE RIGIDEZ ENTRA A ESTA SUBROUTINA DE ACUERDO AL 00005800
5900 ESQUEMA DE ALMACENAJE NUMERO 1. 00005900
6000 SE UTILIZA EL METODO DE ELIMINACION DE GAUSS 00006000
6100 00006100
6200 REFERENCIA 00006200
6300 DIAMBAR P. MONDAR AND GRAHAM H. POWELL 00006300
6400 "TOWARDS OPTIMAL IN CORE EQUATION SOLVING". 00006400
6500 COMPUTERS & STRUCTURES, VOL. 4, PP. 531-548. 00006500
6600 PERGAMON PRESS 1974. PRINTED IN GREAT BRITAIN 00006600
6700 00006700
6800 LA LLAMADA A LA SUBROUTINA ES HECHA POR 00006800
6900 CALL NUMSYM(A,B,NA,NEQ,MBAND,KEY) 00006900
7000 00007000
7100 - - ENTRADA DE VARIABLES - - 00007100
7200 NEQ = NUMERO DE ECUACIONES. 00007200
7300 MBAND = MAXIMO ANCHO DE LA BANDA DE LA MATRIZ. 00007300
7400 A = ES UN ARREGLO DE DIMENSION NEQ X MBAND, LA CUAL CON- 00007400
7500 TIENE LA MATRIZ DE COEFICIENTES ALMACENADA EN LA FORMA 00007500
7600 NUMERO 1. 00007600
7700 B = VECTOR DE DIMENSION NEQ, EL CUAL CONTIENE EL VECTOR 00007700
7800 DE CARGA. 00007800
7900 NA = VECTOR DE DIMENSION NEQ, EL CUAL ENTRA QUE CONTIENE VA- 00007900
8000 LORES ARBITRARIOS. 00008000
8100 KEY = PARAMETROS DE EJECUCION. 00008100
8200 (1) KEY=1 REDUCCION DE LA MATRIZ DE COEFICIENTES. 00008200
8300 (2) KEY=2 REDUCCION DEL VECTOR DE CARGA Y SUSTITUCION 00008300
8400 REGRESIVA. 00008400
8500 (3) KEY=3 REDUCCION DE LA MATRIZ DE COEFICIENTES Y DEL 00008500
8600 VECTOR DE CARGAS Y SUSTITUCION REGRESIVA. 00008600
8700 - - SALIDA - - 00008700
8800 VECTOR B QUE CONTIENE EL VECTOR SOLUCION. 00008800
8900 VECTOR NA QUE CONTIENE LOS VALORES DEL NUMERO DE RENGLONES DEL 00008900
9000 PRIMER ELEMENTO DIFERENTE DE CERO ENTRANDO EN CADA COLUMNA 00009000
9100 DE LA MATRIZ DE COEFICIENTES. 00009100
9200 ARREGLO A QUE CONTIENE LOS VALORES MODIFICADOS DE LA MATRIZ 00009200
9300 00009300
9400 00009400
9500 DIMENSION A(200),B(NEQ),NA(NEQ) 00009500
9600 NEQ=NEQ-1 00009600
9700 NDAA=MBAND 1 00009700
9800 GO TO (10,110,19) KEY 00009800
9900 00009900
10000 REDUCE LA MATRIZ DE COEFICIENTES A 00010000
10100 00010100
10200 10 NADR=1 00010200
10300 DO 30 N=2,MBAND 00010300
10400 NADR=NADR+NEQ 00010400
10500 JADR=NADR 00010500
10600 N1=N-1 00010600
10700 DO 20 J=1,N1 00010700
10800 IF (A(JADR),NE,0) GO TO 30 00010800
10900 20 JADR=JADR-NEQ 00010900

```

11000		I=N1	00011000
11100	30	NA(N)=I	00011100
11200		M=0	00011200
11300		NADR=NEG*MBAA	00011300
11400		DO 50 N=MBAND,NEQ	00011400
11500		M=M+1	00011500
11600		NADR=NADR+1	00011600
11700		JADR=NADR	00011700
11800		N1=N-1	00011800
11900		DO 40 I=M,N1	00011900
12000		IF (A(JADR),NE,0.) GO TO 50	00012000
12100	40	JADR=JADR-NEQ	00012100
12200		I=N1	00012200
12300	50	NA(N)=I	00012300
12400		NA(I)=1	00012400
12500		J1=0	00012500
12600		DO 100 J=2,NEQ	00012600
12700		IF=NA(I)	00012700
12800		IF1=IF+1	00012800
12900		J1=J-1	00012900
13000		J1=J1+NEQ	00013000
13100		JK=J1+NEQ	00013100
13200		IFN=IF*NEQ	00013200
13300		IF (IF1,GT,IL) GO TO 60	00013300
13400		JIA=J1-IFN+1	00013400
13500		IKA=JK-IFN	00013500
13600		NN=IF1*NEQ-JK	00013600
13700		DO 70 I=IF1,IL	00013700
13800		KF=MAX0(NA(I),IF)	00013800
13900		JKA=JK-KF*NEQ	00013900
14000		AA=0.0	00014000
14100		DO 60 K=IKA,JKA,NEQ	00014100
14200	60	AA=AA+A(K)*A(NN+K)	00014200
14300		A(JIA)=A(JIA)-AA	00014300
14400		NN=NN+NEQ	00014400
14500		IKA=IKA-NEQ	00014500
14600	70	JIA=JIA-NEQ	00014600
14700	80	JKA=JK-IFN	00014700
14800		AA=0.0	00014800
14900		DO 90 K=IF,IL	00014900
15000		CC=A(JKA)/A(K)	00015000
15100		AA=AA+A(JKA)*CC	00015100
15200		A(JKA)=CC	00015200
15300	90	JKA=JKA-NEQ	00015300
15400		A(J)=A(J)-AA	00015400
15500	100	CONTINUE	00015500
15600		GO TO (190,190,110) KEX	00015600
15700			00015700
15800			00015800
15900			00015900
16000			00016000
16100	110	DO 120 N=1,NEQ	00016100
16200		IF (D(N),NE,0.) GO TO 130	00016200
16300	120	CONTINUE	00016300
16400		N=NEQ	00016400
16500	130	N1=N+1	00016500
16600		NADR=N1*NEQ	00016600
16700		DO 150 I=N1,NEQ	00016700
16800		KF=MAX0(NA(I),N)	00016800
16900		KL=I-1	00016900
17000		JADR=NADR-KF*NEQ	00017000

C  
E  
D

REDUCCION DEL VECTOR DE CARGA, Y SUSTITUCION REPROSIVA

```

17000      RR=0.0
17100      DO 140 K=KF,KL
17200      RB=RB+A(.IADR)*B(K)
17300 140 .IADR=.IADR-NEQ
17400      B(K)=B(K)-RB
17500 150 NADR=NADR+NEQ
17600      DO 160 I=N,NEQ
17700 160 B(I)=B(I)/A(I)
17800      NN=NEQ
17900      NADR=NN*NADR
18000      DO 180 N=1,NEQ
18100      KF=NA(NN)
18200      KL=NN-1
18300      .IADR=NADR-KF*NEQ
18400      DR=D(NN)
18500      DO 170 K=KF,KL
18600      B(K)=B(K)-A(.IADR)*RR
18700 170 .IADR=.IADR-NEQ
18800      NADR=NADR-NEQ
18900 180 NN=NN-1
19000 190 RETURN
19100      END

```

```

00017000
00017100
00017200
00017300
00017400
00017500
00017600
00017700
00017800
00017900
00018000
00018100
00018200
00018300
00018400
00018500
00018600
00018700
00018800
00018900
00019000
00019100

```

```

0000000000 00 00 00 00
0000000000 00 000 00
00 00 0000 00
00 00 0000 00
000000 00 0000 00
000000 00 0000 00
00 00 00 0000
00 00 00 0000
00 00 00 0000
00 00 00 0000

```

PROGRAMA PARA SOLUCION DE SISTEMAS DE ECUACIONES  
LINEALES PARA MATRICES SIMETRICAS, USANDO ALMACENAJE  
COMPACTO Y ESQUERA DE REDUCCION EN COLUMNA

```

100      SUBROUTINE OPTSOL (A,B,NA,NEQ,KEY)
110      PROGRAM
120      RESUELVE SISTEMAS DE ECUACIONES LINEALES PARA MATRICES
130      SIMETRICAS USANDO ALMACENAJE COMPACTO Y ESQUEMA DE REDUC-
140      CION EN COLIMNA. LA MATRIZ DE RIGIDEZ ENTRA A ESTA SUBRUTI-
150      NA DE ACUERDO AL ESQUEMA DE ALMACENAJE NUMERO ?.
160      SE UTILIZA EL METODO DE CLIMINACION DE GAUSS.
170
180      REFERENCIA
190      DICAMBAR P. MONDKAR AND GRAHAM H. POWELL,
200      "TOWARDS OPTIMAL IN-CORE EQUATION SOLVING",
210      COMPUTERS & STRUCTURES, VOL. 4 PP. 531-540,
220      PERAMON PRESS 1974. PRINTED IN GREAT BRITAIN
230
240      LA LLAMADA A LA SUBROUTINA ES HECHA POR
250      CALL OPTSOL(A,B,NA,NEQ,KEY)
260
270      -- ENTRADA DE VARIABLES --
280      NEQ = NUMERO DE ECUACIONES
290      A   = VECTOR QUE CONTIENE LA MATRIZ DE COEFICIENTES USANDO EL
300      ESQUEMA DE ALMACENAJE NUMERO ?
310      B   = VECTOR DE DIMENSION NEQ, EL CUAL CONTIENE EL VECTOR
320      DE CARGA
330      NA  = VECTOR DE DIMENSION NEQ, CONTIENE LOS VALORES DE NUMERO
340      DEL RENGLON DEL PRIMER ELEMENTO DIFERENTE DE CERO DE CADA
350      COLUMNA EN LA MATRIZ DE COEFICIENTES
360      KEY = PARAMETROS DE EJECUCION
370      (1) KEY=1 REDUCCION DE LA MATRIZ DE COEFICIENTES
380      (2) KEY=2 REDUCCION DEL VECTOR DE CARGA Y SUSTITUCION
390      REGRESIVA
400      (3) KEY=3 REDUCCION DE LA MATRIZ DE COEFICIENTES Y EL VEC-
410      TOR DE CARGA Y SUSTITUCION REGRESIVA
420
430      -- SALIDA --
440      VECTOR B QUE CONTIENE EL VECTOR SOLUCION
450      VECTOR NA QUE CONTIENE LAS DIRECCIONES DE LOS ELEMENTOS DE LA
460      DIAGONAL DE LA MATRIZ DE COEFICIENTES EN EL VECTOR A
470      VECTOR A QUE CONTIENE LA MATRIZ DE COEFICIENTES MODIFICADA
480
490      DIMENSION A(1),B(NEQ),NA(NEQ)
500      NEQ=NEQ-1
510      GO TO (10,100,10) KEY
520
530      REDUC LA MATRIZ DE COEFICIENTES.
540
550      10 NST0=1
560      DO 20 J=2,NEQ
570      NST0=NST0+J-NA(J)+1
580      20 NA(J)=NST0
590      NA(1)=1
600      IL=1
610      NA,IP=NA(1)
620      DO 90 J=2,NEQ
630      NA,J=NA(1)
640      .IK=NA,1-1
650      IF=1-.IK+NA,IP
660
670      00000100
680      00000200
690      00000300
700      00000400
710      00000500
720      00000600
730      00000700
740      00000800
750      00000900
760      00001000
770      00001100
780      00001200
790      00001300
800      00001400
810      00001500
820      00001600
830      00001700
840      00001800
850      00001900
860      00002000
870      00002100
880      00002200
890      00002300
900      00002400
910      00002500
920      00002600
930      00002700
940      00002800
950      00002900
960      00003000
970      00003100
980      00003200
990      00003300
1000     00003400
1010     00003500
1020     00003600
1030     00003700
1040     00003800
1050     00003900
1060     00004000
1070     00004100
1080     00004200
1090     00004300
1100     00004400
1110     00004500
1120     00004600
1130     00004700
1140     00004800
1150     00004900
1160     00005000
1170     00005100
1180     00005200
1190     00005300
1200     00005400
1210     00005500
1220     00005600
1230     00005700

```

5800	IF (IF,GE,J) GO TO 60	00005800
5900	IF1=IF+1	00005900
6000	IF (IF1,GT,IL) GO TO 60	00006000
6100	JIA=2+NA*IP	00006100
6200	NAIP=NA(IF)	00006200
6300	KL=IF+.JK	00006300
6400	DO 50 I=IF1,IL	00006400
6500	NAI=NA(I)	00006500
6600	IK=NAI-I	00006600
6700	II=I+I-NAI+NAIP	00006700
6800	IF (II,GE,I) GO TO 40	00006800
6900	KF=MAX0(II,IF)+.JK	00006900
7000	JJ=IK+.JK	00007000
7100	AA=0.0	00007100
7200	DO 30 K=KF,KI	00007200
7300	AA=AA+A(K)*A(LIK)	00007300
7400	A(JIA)=A(JIA)-AA	00007400
7500	40 JIA=JIA+1	00007500
7600	KI=KL+I	00007600
7700	50 NAIP=NAI	00007700
7800	60 KF=IK+IF	00007800
7900	KL=NAJ-I	00007900
8000	AA=0.0	00008000
8100	DO 70 K=KF,KL	00008100
8200	NAI=NA(IF)	00008200
8300	CC=A(K)/A(NAI)	00008300
8400	AA=AA+A(K)*CC	00008400
8500	A(K)=CC	00008500
8600	70 IF=IF+1	00008600
8700	A(NAJ)=A(NAJ)-AA	00008700
8800	80 IL=IL+1	00008800
8900	90 NAJ=NAJ	00008900
9000	GO TO (200,200,100),KEY	00009000
9100	C	00009100
9200	C	00009200
9300	C	00009300
9400	100 DO 110 N=J,NEQ0	00009400
9500	IF (R(N),NE,0.) GO TO 120	00009500
9600	110 CONTINUE	00009600
9700	N=NEQ0	00009700
9800	120 NI=N+1	00009800
9900	II=NI+1	00009900
10000	KL=N	00010000
10100	NAIP=NA(N)	00010100
10200	DO 150 I=NI,NE0	00010200
10300	NAI=NA(I)	00010300
10400	II=II-NAI+NAIP	00010400
10500	IF (II,GE,I) GO TO 140	00010500
10600	KF=MAX0(II,N)	00010600
10700	IK=NAI-I	00010700
10800	IKA=IK+KF	00010800
10900	BB=0.0	00010900
11000	DO 130 K=KF,KI	00011000
11100	BR=BR+A(IKA)*B(K)	00011100
11200	130 IKA=IKA+1	00011200
11300	R(I)=R(I)-BR	00011300
11400	140 II=II+1	00011400
11500	KL=KL+1	00011500
11600	150 NAIP=NAI	00011600
11700	DO 160 J=N,NE0	00011700

```

11800      NAT=NA(I)
11900      160 R(I)=R(I)/A(NAT)
12000      J=NEQ
12100      NAJ=NA(NEQ)
12200      DO 170 I=1,NEQD
12300      NAJIP=NA(J,I)
12400      JKA=NAJIP+1
12500      II=J-NAJ+JKA
12600      IF (II.GE.J) GO TO 180
12700      KI=J-1
12800      BB=B(J)
12900      DO 170 K=II,KI
13000      R(K)=B(K)-A(JKA)*BB
13100      170 JKA=JKA+1
13200      180 J=J-1
13300      190 NAJ=NAJIP
13400      200 RETURN
13500      END

```

```

00011800
00011900
00012000
00012100
00012200
00012300
00012400
00012500
00012600
00012700
00012800
00012900
00013000
00013100
00013200
00013300
00013400
00013500

```

```

0000000000 00 00 00
0000000000 00 000 00
00 00 000 00
00 00 000 00
000000 00 00 000 00
000000 00 00 000 00
00 00 00 0000
00 00 00 0000
00 00 00 0000
00 00 00 0000

```

PROGRAMA PARA CALCULAR LAS FORMAS MODALES Y FRECUEN-  
CIAS DE UN SISTEMA ESTRUCTURAL CON UNA MATRIZ DE MA-  
SAS DIAGONAL

3800		SUBROUTINE EVAL/EVAL (SM,VAL,VEC,NEG,NM,LM,MT,D,CO,NA,	00003800
3900		*ALLOW,ND,IN,C)	00003900
4000	C		00004000
4100	C	PROGRAMA	00004100
4200	C	CALCULA LAS FORMAS MODALES Y FRECUENCIAS DE UN SISTEMA ESTRU-	00004200
4300	C	CTURAL CON UNA MATRIZ DE MASAS DIAGONAL.	00004300
4400	C	ESTA SUBROUTINA UTILIZA EL METODO DE PODER INVERSO CON CAMBIOS DE	00004400
4500	C	POWER METHOD WITH SHIFTS), PARA CALCULAR LOS VALORES Y VECTORES	00004500
4600	C	CHARACTERISTICOS.	00004600
4700	C		00004700
4800	C	REFERENCIA	00004800
4900	C	J.L.L. HUMAR	00004900
5000	C	"EIGENVALUE PROGRAMS FOR BUILDING STRUCTURES",	00005000
5100	C	COMPUTER & STRUCTURES, VOL. 8, PP. 25-31,	00005100
5200	C	PERGAMON PRESS, 1968, PRINTED IN GREAT BRITAIN.	00005200
5300	C		00005300
5400	C	LA LLAMADA A LA SUBROUTINA ES HECHA POR	00005400
5500	C	EVAL/EVAL (SM,VAL,VEC,NEG,NM,LM,MT,D,CO,NA,ALLOW,ND,IN,C)	00005500
5600	C		00005600
5700	C	- - ENTRADA DE VARIABLES - -	00005700
5800	C	ND =DIMENSION ASIGNADA A LA MATRIZ DE RIGIDEZ EN LA LLAMADA	00005800
5900	C	AL PROGRAMA.	00005900
6000	C	NDO =ORDEN DE LA MATRIZ DE RIGIDEZ.	00006000
6100	C	NM =NO. DE MODOS A SER CALCULADOS.	00006100
6200	C	LM =NO. DE MODOS A SER ENCONTRADOS DE UN ORIGEN.	00006200
6300	C	MT =MAXIMO NUMERO DE ITERACIONES A SER USADAS EN EL CALCULO	00006300
6400	C	DE UNA FORMA MODAL.	00006400
6500	C	IW =NUMERO CORRESPONDIENTE A LA UNIDAD DE SALIDA.	00006500
6600	C	A =MATRIZ DE RIGIDEZ.	00006600
6700	C	SM =VECTOR QUE CONTIENE LA DIAGONAL DE LA MATRIZ DE MASAS.	00006700
6800	C	VAL =ARREGLO QUE CONTIENE LAS FRECUENCIAS.	00006800
6900	C	VEF =MATRIZ QUE CONTIENE LAS FORMAS MODALES; CADA FORMA MODAL	00006900
7000	C	FORMA UNA COLUMNA DE LA MATRIZ.	00007000
7100	C	D =MATRIZ DINAMICA.	00007100
7200	C	CO =ARREGLO QUE CONTIENE UN VECTOR ARBITRARIO.	00007200
7300	C	NA =ARREGLO QUE CONTIENE LOS VALORES DE LOS NUMEROS DE RENGLON	00007300
7400	C	DEL PRIMER ELEMENTO DIFERENTE DE CERO QUE ENTRA EN	00007400
7500	C	CADA COLUMNA DE LA MATRIZ DE RIGIDEZ.	00007500
7600	C	ALLOW=NUMERO DE PUNTO FLOTANTE ESPERITANDO EL CRITERIO DE CON-	00007600
7700	C	VERGENCIA.	00007700
7800	C	D =ARREGLO DE ALMACENAMIENTO TEMPORAL USADO PARA ARCHIVAR LA DIA-	00007800
7900	C	AGONAL DE LA MATRIZ DE MASAS.	00007900
8000	C		00008000
8100	C	- ENTRADA DE VARIABLES -	00008100
8200	C	VAL,VEC,D,CO,NA, Y C QUEBEN CONTENER VALORES ARBITRARIOS.	00008200
8300	C		00008300
8400	C	- SALIDA DE VARIABLES -	00008400
8500	C	VAL =CONTIENE LAS FRECUENCIAS.	00008500
8600	C	VEC =CONTIENE LAS FORMAS MODALES.	00008600
8700	C	NA =CONTIENE LOS VALORES DEL NUMERO DE RENGLON DEL PRIMER	00008700
8800	C	ELEMENTO DIFERENTE DE CERO QUE ENTRA EN CADA COLUMNA	00008800
8900	C	DE LA MATRIZ DE RIGIDEZ A.	00008900
9000	C		00009000
9100	C		00009100
9200	C	DIMENSION A (ND,NDO,SM(C),VEC(NM,LM),VAL(C),D(ND,ND)	00009200
9300	C	*CO(C),NA(C),C(C)	00009300
9400	C		00009400

9500	C	ARCHIVA LA DIAGONAL DE A EN C.	00009500
9600	C		00009600
9700		DO 10 I=1,NEQ	00009700
9800		C(I)=A(I,I)	00009800
9900		10 CONTINUE	00009900
10000	C		00010000
10100	C	FORMA LA MATRIZ DINAMICA.	00010100
10200	C		00010200
10300		CALL DYNQ(A,NEQ,NA,ND,D,SM,C)	00010300
10400		SH=0.	00010400
10500		K=1	00010500
10600		LL=0	00010600
10700		R1=1.E-04	00010700
10800		NVT=0	00010800
10900	C		00010900
11000	C	INICIALIZA UN VECTOR PRUEBA.	00011000
11100	C		00011100
11200		20 DO 30 I=1,NEQ	00011200
11300		CO(I)=1.	00011300
11400		30 CONTINUE	00011400
11500		NT=1	00011500
11700	C		00011700
11800	C	OBTIENE UN NUEVO VECTOR DE PRUEBA.	00011800
11900	C		00011900
12000		40 DO 50 I=1,NEQ	00012000
12100		VEC(I,K)=0.	00012100
12200		DO 50 J=1,NEQ	00012200
12300		VEC(I,K)=VEC(I,K)+(C(I,J)*CO(J))	00012300
12400		50 CONTINUE	00012400
12500	C		00012500
12600	C	NORMALIZA EL VECTOR DE PRUEBA Y DIVIDE ENTRE EL	00012600
12700	C	ELEMENTO MAS GRANDE.	00012700
12800	C		00012800
12900		NT=1	00012900
13000		CMAX=ABS(VEC(I,K))	00013000
13100		DO 60 I=2,NEQ	00013100
13200		IF (ABS(VEC(I,K)) .GT. CMAX) GO TO 60	00013200
13300		CMAX=ABS(VEC(I,K))	00013300
13400		NT=1	00013400
13500		60 CONTINUE	00013500
13600		TT=CO(NT)/VEC(NT,K)	00013600
13700		TTS=TT/ABS(TT)	00013700
13800		CMAX=CMAX*TTS	00013800
13900		DO 70 I=1,NEQ	00013900
14000		VEC(I,K)=VEC(I,K)/CMAX	00014000
14100		70 CONTINUE	00014100
14200	C		00014200
14300	C	PRUEBA SI EL CAMBIO DE ORIGEN ES REQUERIDO.	00014300
14400	C		00014400
14500		IF (LL .LT. LM) GO TO 80	00014500
14600		IF (NVT .EQ. 1) GO TO 80	00014600
14700		IF (NI .LT. 10) GO TO 80	00014700
14800		VT=SH*ABS(L./CMAX)	00014800
14900		K=K+1	00014900
15000		VVT=SORT(VT)	00015000
15100		IF (ABS(VVT-VAL(VT)) .GT. .05*VAL(VT)) GO TO 110	00015100
15200		NVT=1	00015200
15300	C		00015300
15400	C	PRUEBA LA CONVERGENCIA.	00015400

15500	90	ERRMAX=0.	00015500
15600		DO 90 I=1,NED	00015600
15700		IF (ABS(VEC(I,K))-1.1*ERR) GO TO 90	00015700
15800		ERR=(QQ(I)-VEC(I,K))/VEC(I,K)	00015800
15900		ERR=ABS(ERR)	00015900
16000		IF (ERR.GT.ERMAX) ERMAX=ERR	00016000
16100	90	CONTINUE	00016100
16200		IF (ERMAX.LE.ALLOW) GO TO 150	00016200
16300		IF (NI.GT.MI) GO TO 150	00016300
16400		DO 100 I=1,NED	00016400
16500		QQ(I)=VEC(I,K)	00016500
16600	100	CONTINUE	00016600
16700		NI=NI+1	00016700
16800		GO TO 40	00016800
16900	C		00016900
17000	C	ADUI OBTIENE EL CAMBIO DE ORIGEN Y LA NUEVA MATRIZ DINAMICA.	00017000
17100	C		00017100
17200	110	SH=(VT*VAL(KT)+VAL(PT))/Z.	00017200
17300		DO 120 I=1,NED	00017300
17400		A(I,I)=A(I,I)-SH*SM(I)	00017400
17500		QQ(I)=VEC(I,K)	00017500
17600	130	CONTINUE	00017600
17700		CALL DYNQ(A,NED,NA,NL,TS,MI)	00017700
17800	C		00017800
17900	C	REVISLA LA MATRIZ DINAMICA PARA LLEVAR FUERA LAS FORMAS MODALES	00017900
18000	C	YA DETERMINADAS.	00018000
18100	C		00018100
18200		KK=K-1	00018200
18300		DO 140 N=1,KK	00018300
18400		HAT=1./((VAL(N)*VAL(N)-SH)	00018400
18500		DO 140 I=1,NED	00018500
18600		DO 140 J=1,NED	00018600
18700		B(I,J)=B(I,J)-HAT*VEC(I,N)*VEC(I,N)*SM(I)	00018700
18800	140	CONTINUE	00018800
18900		II=0	00018900
19000		NI=NI+1	00019000
19100		GO TO 40	00019100
19200	C		00019200
19300	C	ADUI CUANDO LA ITERACION HA CONVERGIDO CALCULA LA MI DIMENSIONAL	00019300
19400	C	FRECUENCIA Y EL VECTOR MODAL NORMALIZADO.	00019400
19500	C		00019500
19600	150	VECTOR=1	00019600
19700		VAL(K)=SORT(SH)/CHMAX	00019700
19800		ALFA=0.	00019800
19900		DO 160 I=1,NED	00019900
20000		ALFA=ALFA+VEC(I,I)*VEC(I,I)*SM(I)	00020000
20100	160	CONTINUE	00020100
20200		BETA=SORT(ALFA)	00020200
20300		DO 170 I=1,NED	00020300
20400		VEC(I,K)=VEC(I,I)/BETA	00020400
20500	170	CONTINUE	00020500
20600		TE(K,NM)=180,200,200	00020600
20700	180	CONTINUE	00020700
20800	C		00020800
20900	C	REVISLA LA MATRIZ DINAMICA PARA LLEVAR FUERA LAS ULTIMAS FORMAS	00020900
21000	C	MODALES A SER DETERMINADAS.	00021000
21100	C		00021100
21200		DO 120 I=1,NED	00021200
21300		DO 190 J=1,NED	00021300
21400		TCT(I,J)=B(I,J)-CHMAX*VEC(I,I)*VEC(I,J)*SM(I)	00021400

21500	190	CONTINUE	00021500
21600		K=K+1	00021600
21700		II=II+1	00021700
21800		NVT=0	00021800
21900		GO TO 20	00021900
22000	200	RETURN	00022000
22100	301	FORMAT (" MODULO NUMERO.....=" ,I4)	00022100
22200	302	FORMAT (" NUMERO DE ITERACION.....=" ,I4)	00022200
22300	303	FORMAT (" DIFERENCIA MAXIMA ENTRE SUSCRIPTOS.....=" ,I4)	00022300
22400		* " VECTORES DE PRIERA ...=" ,F10.7,7)	00022400
22500		END	00022500
22600		SUBROUTINE DYNQ(A,NFQ,NA,ND,D,SM,C)	00022600
22700			00022700
22800	C	FORMA LA MATRIZ DINAMICA	00022800
22900	C		00022900
23000		DIMENSION A(1),D(1),NA(1),SM(1),C(1)	00023000
23100		CALL FORE(A,NFQ,NA,ND)	00023100
23200		IT=0	00023200
23300		DO 20 J=2,NFQ	00023300
23400		II=J-1	00023400
23500		IT=IT+ND	00023500
23600		JADR=1	00023600
23700		DO 20 I=1,II	00023700
23800		IADR=IT+I	00023800
23900		D(IADR)=0.	00023900
24000		D(JADR)=0.	00024000
24100		JADR=JADR+ND	00024100
24200	20	CONTINUE	00024200
24300		JADR=1	00024300
24400		DO 30 J=1,NFQ	00024400
24500		D(JADR)=SM(J)	00024500
24600		JADR=JADR+ND+1	00024600
24700	30	CONTINUE	00024700
24800		NK=1	00024800
24900		DO 40 J=1,NFQ	00024900
25000		NR=1	00025000
25100		CALL BACK(A,NFQ,IA,NF,ND,D(NK))	00025100
25200		NK=NK+ND	00025200
25300	40	CONTINUE	00025300
25400		IT=0	00025400
25500		DO 50 J=2,NFQ	00025500
25600		II=J-1	00025600
25700		IT=IT+ND	00025700
25800		JADR=1	00025800
25900		DO 50 I=1,II	00025900
26000		IADR=IT+I	00026000
26100		A(IADR)=A(JADR)	00026100
26200		JADR=JADR+ND	00026200
26300	50	CONTINUE	00026300
26400		JADR=1	00026400
26500		DO 60 J=1,NFQ	00026500
26600		A(JADR)=C(J)	00026600
26700		JADR=JADR+ND+1	00026700
26800	60	CONTINUE	00026800
26900		RETURN	00026900
27000		END	00027000
27100		SUBROUTINE FORE(A,NFQ,NA,ND)	00027100
27200			00027200
27300	C	CONSTRUYE EL VECTOR NA Y LLEVA FUERA LA REDUCCION ESPECIAL	00027300
27400	C	DE LA MATRIZ A USANDO EL METODO DE PRODUIS.	00027400

27500				00027500
27600				00027600
27700				00027700
27800				00027800
27900				00027900
28000				00028000
28100				00028100
28200				00028200
28300				00028300
28400				00028400
28500				00028500
28600				00028600
28700				00028700
28800				00028800
28900				00028900
29000				00029000
29100				00029100
29200				00029200
29300				00029300
29400				00029400
29500				00029500
29600				00029600
29700				00029700
29800				00029800
29900				00029900
30000				00030000
30100				00030100
30200				00030200
30300				00030300
30400				00030400
30500				00030500
30600				00030600
30700				00030700
30800				00030800
30900				00030900
31000				00031000
31100				00031100
31200				00031200
31300				00031300
31400				00031400
31500				00031500
31600				00031600
31700				00031700
31800				00031800
31900				00031900
32000				00032000
32100				00032100
32200				00032200
32300				00032300
32400				00032400
32500				00032500
32600				00032600
32700				00032700
32800				00032800
32900				00032900
33000				00033000
33100				00033100
33200				00033200
33300				00033300
33400				00033400

```

C
27500      DIMENSION A(1),NA(1)
27600      NADR=1
27700      DO 20 J=2,NFO
27800      NADR=NADR+ND
27900      LIMIT=NADR+J-1
28000      NJ=1
28100      DO 10 I=NADR,LIMIT
28200      IF (A(I).NE.0) GO TO 20
28300      NI=NJ+1
28400      10 CONTINUE
28500      20 NA(J)=NI
28600      NA(1)=1
28700      JT=0
28800      DO 30 J=2,NFO
28900      JT=JT+ND
29000      I=NA(J)
29100      II=J-1
29200      IF (II.GT.1) GO TO 20
29300      IN=II+1
29400      JJ=IT+1
29500      IF (IN.GT.1) GO TO 50
29600      DO 40 I=IN,II
29700      M=NA(I)
29800      KI=MAXO(II,M)
29900      KI=I-1
30000      IT=KI+ND
30100      SUM=0.
30200      DO 30 K=KI,KL
30300      JADR=IT+K
30400      JADR=JIT+K
30500      SUM=SUM+A(JADR)*A(JADR)
30600      30 CONTINUE
30700      I.=IT+1
30800      A(I.)=A(I.)-SUM
30900      40 CONTINUE
31000      50 SUM=0.
31100      NADR=(I.-1)*ND+1
31200      DO 60 K=I.,II
31300      JADR=JIT+K
31400      C=A(JADR)/A(NADR)
31500      SUM=SUM+A(JADR)*C
31600      A(JADR)=C
31700      NADR=NADR+ND+1
31800      60 CONTINUE
31900      A(I.)=A(I.)-SUM
32000      70 CONTINUE
32100      RETURN
32200      END
32300      SUBROUTINE BACI (A,NFO,NA,NB,ND,I)
32400      C
32500      D
32600      E
32700      F
32800      G
32900      DIMENSION A(1),NA(1),NB(1)
33000      IN=NB(1)
33100      IF (IN.GT.NFO) GO TO 30
33200      DO 20 I=IN,NFO
33300      M=NA(I)
33400      KI=MAXO(M,NB)

```

```

135000 K1=1-1
136000 IF(K1-GT,K1) GO TO 20
137000 I=K1*ND
138000 SUM=0.
139000 DO 10 K=I,K1
140000 IADR=I+K
141000 SUM=SUM+A(IADR)*R(K)
142000 CONTINUE
143000 D(I)=R(I)-SUM
144000 CONTINUE
145000 20 IADR=(NB-1)*ND+I
146000 DO 40 I=NR,NEO
147000 F(I)=R(I)/A(IADR)
148000 IADR=IADR+ND
149000 CONTINUE
150000 I=NEO
151000 I=NEO*ND
152000 KI=NA(I)
153000 KI=1-1
154000 I=I-ND
155000 IF(K1-GT,K1) GO TO 70
156000 DO 40 K=KI,K1
157000 IADR=I+K
158000 R(K)=R(K)-A(IADR)*R(I)
159000 CONTINUE
160000 IF(I.EQ.2) GO TO 80
161000 I=1-1
162000 GO TO 50
163000 RETURN
164000 END
000337500
000336000
000337000
000338000
000339000
000340000
000341000
000342000
000343000
000344000
000345000
000346000
000347000
000348000
000349000
000350000
000351000
000352000
000353000
000354000
000355000
000356000
000357000
000358000
000359000
000360000
000361000
000362000
000363000
000364000

```

```

000365000
000366000
000367000
000368000
000369000
000370000
000371000
000372000
000373000
000374000
000375000
000376000
000377000
000378000
000379000
000380000
000381000
000382000
000383000
000384000
000385000
000386000
000387000
000388000
000389000
000390000
000391000
000392000
000393000
000394000
000395000
000396000
000397000
000398000
000399000
000400000

```

PROGRAMA PARA RESOLVER EL PROBLEMA GENERALIZADO DE  
VALORES Y VECTORES CARACTERISTICOS USANDO EL METODO  
GENERALIZADO DE ITERACION DE JACOBI

```

3100      SUBROUTINE JACOBI(A,D,X,EIGV,D,N,RTOL,NMAX,IERR,IOUT)
3200      C
3300      C
3400      C
3500      C
3600      C
3700      C
3800      C
3900      C
4000      C
4100      C
4200      C
4300      C
4400      C
4500      C
4600      C
4700      C
4800      C
4900      C
5000      C
5100      C
5200      C
5300      C
5400      C
5500      C
5600      C
5700      C
5800      C
5900      C
6000      C
6100      C
6200      C
6300      C
6400      C
6500      C
6600      C
6700      C
6800      C
6900      C
7000      C
7100      C
7200      C
7300      C
7400      C
7500      C
7600      C
7700      C
7800      C
7900      C
8000      C
8100      C
8200      C
8300      C
8400      C
8500      C
8600      C
8700      C

```

SUBROUTINE JACOBI(A,D,X,EIGV,D,N,RTOL,NMAX,IERR,IOUT)

PROGRAMA  
PARA RESOLVER EL PROBLEMA GENERALIZADO DE VALORES Y VECTORES  
CARACTERISTICOS USANDO EL METODO GENERALIZADO DE ITERACION DE  
JACOBI.

REFERENCIA  
KLAUS-JURGEN RATHE AND EDWARD L. HILTON  
"NUMERICAL METHODS IN FINITE ELEMENT ANALYSIS"  
ENGLEWOOD CLIFFS, NEW JERSEY,  
1976 BY PRENTICE HALL, INC.

LA LLAMADA A LA SUBROUTINA ES HECHA POR  
CALL JACOBI(A,D,X,EIGV,D,N,RTOL,NMAX,IERR,IOUT)

--- ENTRADA DE VARIABLES ---  
A(N,N) =MATRIZ DE RIGIDEZ(POSITIVA DEFINIDA)  
D(N,N) =MATRIZ DE MASAS(POSITIVA DEFINIDA)  
X(N,N) =MATRIZ QUE CONTIENE LOS VECTORES CARACTERISTICOS EN LA  
SALIDA  
EIGV(N) =VECTOR QUE CONTIENE LOS VALORES CARACTERISTICOS EN LA  
SALIDA  
D(N) =VECTOR DE TRABAJO  
N =ORDEN DE LAS MATRICES A Y B  
RTOL =TOLERANCIA EN LA CONVERGENCIA(CUASALMENTE 10\*\*(-12))  
NMAX =MAXIMO NUMERO DE ROTACIONES(CUASALMENTE 15)  
IERR =BANDERA PARA ESCRIBIR DURANTE LA ITERACION  
EQ.=0 NO ESCRIBIR  
EQ.=1 ESCRIBIR LOS RESULTADOS INTERMEDIOS  
IOUT =NUMERO CORRESPONDIENTE A LA UNIDAD DE SALIDA

--- SALIDA DE VARIABLES ---  
A(N,N) =MATRIZ DE RIGIDEZ DIAGONALIZADA  
D(N,N) =MATRIZ DE MASAS DIAGONALIZADA  
X(N,N) =VECTORES CARACTERISTICOS ALMACENADOS POR COLUMNAS  
EIGV(N) =VECTORES CARACTERISTICOS

IMPLICIT REAL\*8(A-H,O-Z)  
ABS(X)=ABS(X)  
SORT(X)=DSORT(X)  
DIMENSION A(N,N),D(N,N),X(N,N),EIGV(N),D(N)

INICIALIZA LAS MATRICES DE VALORES Y VECTORES CARACTERISTICOS.

DO 10 I=1,N  
IF(A(I,I).GT.0. .AND. D(I,I).LT.0.) GO TO 4  
WRITE (IOUT,\*)  
\*TOP  
4 D(I,I)=A(I,I)/D(I,I)  
10 EIGV(I)=D(I,I)  
DO 30 J=1,N  
DO 20 L=1,N  
20 X(L,I)=0.  
30 X(I,I)=1.  
IF(N.EQ.1) RETURN

8800	C		00008800
8900	C	INICIA LA CUENTA DE LAS ROTACIONES Y COMIENZA LA ITERACION	00008900
9000	C		00009000
9100		NSWEEP=0	00009100
9200		NR=N-1	00009200
9300	40	NSWEEP=NSWEEP+1	00009300
9400		IF ((EPR.FO.1)WRITE (OUT,2000) NSWEEP	00009400
9500	C		00009500
9600	C	REVISAR SI EL PRESENTE ELEMENTO FUERA DE LA DIAGONAL ES LO BASTANTE	00009600
9700	C	GRANDE PARA HACERLO CERO	00009700
9800	C		00009800
9900		FPS=(0.01*NSWEEP)**2	00009900
10000		DO 210 J=1, NR	00010000
10100		JJ=J+1	00010100
10200		DO 210 K=JJ, N	00010200
10300		EPTOLA=(A(J,K)*A(J,K))/(A(J,J)*A(K,K))	00010300
10400		EPTOLB=(B(J,K)*B(J,K))/(B(J,J)*B(K,K))	00010400
10500		IF ((EPTOLA.LT.FPS).AND.(EPTOLB.LT.FPS)) GO TO 210	00010500
10600	C		00010600
10700	C	SI SE REQUIERE HACER CERO, CALCULA LA MATRIZ DE ROTACION ELEMENTO	00010700
10800	C	CA Y CG	00010800
10900	C		00010900
11000		AKK=A(K,K)*B(J,K) - B(K,K)*A(J,J)	00011000
11100		AJJ=A(J,J)*B(J,K) - B(J,J)*A(K,K)	00011100
11200		AB=A(J,J)*B(K,K) - A(K,K)*B(J,J)	00011200
11300		CHECK=(AB**2+.4*AKK**2+.4*AJJ**2)	00011300
11400		IF (CHECK) 50, 60, 70	00011400
11500	50	WRITE (OUT, 2020)	00011500
11600		STOP	00011600
11700	60	SQCH=SQRT(CHECK)	00011700
11800		D1=AB/SQCH	00011800
11900		D2=AB/2.-SQCH	00011900
12000		DEN=D1	00012000
12100		IF (ABS(D2).GT.ABS(D1)) DEN=D2	00012100
12200		IF (DEN) 80, 70, 80	00012200
12300	70	CA=0.	00012300
12400		CG=-A(J,K)/A(K,K)	00012400
12500		GO TO 90	00012500
12600	80	CA=AKK/DEN	00012600
12700		CG=AJJ/DEN	00012700
12800	C		00012800
12900	C	ENCUENTA LA ROTACION GENERALIZADA A CERO DEL PRESENTE ELEMENTO	00012900
13000	C	FUERA DE LA DIAGONAL	00013000
13100	C		00013100
13200	20	IF (N-2) 100, 190, 190	00013200
13300	100	JP1=J+1	00013300
13400		JMI=J-1	00013400
13500		KP1=K+1	00013500
13600		KMI=K-1	00013600
13700		IF (N) 110, 130, 110, 110	00013700
13800	110	DO 120 I=1, JMI	00013800
13900		AJ=A(I,J)	00013900
14000		BJ=B(I,J)	00014000
14100		AK=A(I,K)	00014100
14200		BK=B(I,K)	00014200
14300		A(I,J)=A.I+CG*AK	00014300
14400		B(I,J)=B.I+CG*BK	00014400
14500		A(I,K)=AK+CA*AJ	00014500
14600	120	B(I,K)=BK+CA*BJ	00014600
14700	130	IF (N) 140, 140, 140	00014700

14800	140	DO 150 I=KPI..J	00014800
14900		A,I=A(I,I)	00014900
15000		B,J=B(J,I)	00015000
15100		AK=A(K,I)	00015100
15200		BK=B(K,I)	00015200
15300		A(I,I)=A.I*CG*AK	00015300
15400		D(I,I)=R.J*CG*BK	00015400
15500		A(K,I)=AK+CA*AJ	00015500
15600	150	R(K,I)=BK+CA*B.I	00015600
15700	160	IF (JPI-KM1) 170,170,190	00015700
15800	170	DO 180 I=JPI..KM1	00015800
15900		A,I=A(I,I)	00015900
16000		R,I=R(I,I)	00016000
16100		AK=A(I,K)	00016100
16200		BK=B(I,K)	00016200
16300		A(I,I)=A.I*CG*AK	00016300
16400		R(I,I)=BK+CA*B.I	00016400
16500		A(I,K)=AK+CA*AJ	00016500
16600	180	R(I,K)=BK+CA*B.I	00016600
16700	190	AK=A(K,K)	00016700
16800		BK=B(K,K)	00016800
16900		A(K,I)=AK+2.*CA*A(I,I)*CG+CA*B(K,I)*CG*AK	00016900
17000		B(K,I)=BK+2.*CA*B(I,I)*CG+CA*R(K,I)*CG*BK	00017000
17100		A(I,I)=A(I,I)+2.*CG*A(I,K)*CG*CG*AK	00017100
17200		B(I,I)=B(I,I)+2.*CG*B(I,K)*CG*CG*BK	00017200
17300		A(I,I)=0.	00017300
17400		R(I,I)=0.	00017400
17500	C		00017500
17600	C	CALCULA LA NUEVA MATRIZ DE VALORES CARACTERISTICOS DESPUES DE	00017600
17700	C	CADA ROTACION	00017700
17800	C		00017800
17900		DO 200 I=1..N	00017900
18000		X,I=X(I,I)	00018000
18100		XK=X(I,K)	00018100
18200		X(I,I)=X.I*CG*YK	00018200
18300	200	X(I,K)=XK+CA*X.I	00018300
18400	310	CONTINUE	00018400
18500	C		00018500
18600	C	CALCULA LOS ULTIMOS VALORES CARACTERISTICOS DESPUES DE CADA	00018600
18700	C	ROTACION	00018700
18800	C		00018800
18900		DO 220 I=1..N	00018900
19000		IF (A(I,I).GT.0. .AND. B(I,I).GT.0.) GO TO 230	00019000
19100		WRITE (IOUT,20701)	00019100
19200		STOP	00019200
19300	220	EIGV(I)=A(I,I)/B(I,I)	00019300
19400		IF (I.FPR.EQ.0) GO TO 230	00019400
19500		WRITE (IOUT,20702)	00019500
19600		DO 1234 J=1..N	00019600
19700		WRITE (IOUT,20703) EIGV(I)	00019700
19800	1234	CONTINUE	00019800
19900	C		00019900
20000	C	PRUEBA LA CONVERGENCIA	00020000
20100	C		00020100
20200		DO 240 I=1..N	00020200
20300		TOL=RTOL*B(I)	00020300
20400		DIF=ABS(EIGV(I)-B(I))	00020400
20500		IF (DIF.GT.TOL) GO TO 250	00020500
20600	240	CONTINUE	00020600
20700	C		00020700

```

20800 C CHECA TODOS LOS ELEMENTOS FUERA DE LA DIAGONAL PARA VER SI CITA
20900 C ROTACION ES REQUERIDA
21000 C
21100 EPS=RTOL**2
21200 DO 250 J=1,NR
21300 J1=J+1
21400 DO 250 K=L1,N
21500 EPSA=(A(J,K)*A(J,K))/(A(J,J)*A(K,K))
21600 EPSB=(B(J,K)*B(J,K))/(B(J,J)*B(K,K))
21700 IF ((EPSA.LT.EPS).AND.(EPSB.LT.EPS)) GO TO 250
21800 GO TO 260
21900 250 CONTINUE
22000 255 DO 260 J=1,N
22100 DO 260 J=1,N
22200 A(J,1)=A(I,J)
22300 260 B(J,1)=B(I,J)
22400 DO 270 J=1,N
22500 BB=SQRT(B(J,1))
22600 DO 270 K=1,N
22700 X(K,1)=X(K,1)/BB
22800 RETURN
22900 C
23000 C OBTIENE LA ULTIMA MATRIZ D Y COMIENZA UNA NUEVA BARRIDA. SI
23100 C ES REQUERIDA
23200 C
23300 280 DO 290 I=1,N
23400 290 D(I)=EIGV(I)
23500 IF(NSWEEP.LT.NSMAX) GO TO 40
23600 GO TO 255
23700 2000 FORMAT("BARRIDA NUMERO EN *JACOBI* =",I4)
23800 2010 FORMAT(1H0,6E20,12)
23900 2020 FORMAT("***ERROR EN LA SOLUCION FIN * /
24000 * " MATRIZ NO DEFINIDA POSITIVA")
24100 2030 FORMAT(" LOS VALORES EN *JACOBI* SON" /)
24200 END

```

```

0000000000 00 00 00
0000000000 00 000 00
00 00 000 00
00 000 00
00000 00 000 00
00000 00 000 00
00 00 00 000
00 00 00 000
00 00 00 000
00 00 00 000

```

PROGRAMA PARA CALCULAR LAS FORMAS MODALES Y FRECUEN-  
CIAS DE UN SISTEMA ESTRUCTURAL POR EL METODO DE  
LANCZOS

```

100 SUBROUTINE F766RCA(LAM1,ZZ,INEQS,IPN,ITL,NECS,NR) 0000100
200 C 0000200
300 C 0000300
400 C 0000400
500 C 0000500
600 C 0000600
700 C 0000700
800 C 0000800
900 C 0000900
1000 C 0001000
1100 C 0001100
1200 C 0001200
1300 C 0001300
1400 C 0001400
1500 C 0001500
1600 C 0001600
1700 C 0001700
1800 C 0001800
1900 C 0001900
2000 C 0002000
2100 C 0002100
2200 C 0002200
2300 C 0002300
2400 C 0002400
2500 C 0002500
2600 C 0002600
2700 C 0002700
2800 C 0002800
2900 C 0002900
3000 C 0003000
3100 C 0003100
3200 C 0003200
3300 C 0003300
3400 C 0003400
3500 C 0003500
3600 C 0003600
3700 C 0003700
3800 C 0003800
3900 C 0003900
4000 C 0004000
4100 C 0004100
4200 C 0004200
4300 C 0004300
4400 C 0004400
4500 C 0004500
4600 C 0004600
4700 C 0004700
4800 C 0004800
4900 C 0004900
5000 C 0005000
5100 C 0005100
5200 C 0005200
5300 C 0005300
5400 C 0005400
5500 C 0005500
5600 C 0005600
5700 C 0005700
5800 C 0005800
5900 C 0005900
6000 C 0006000
6100 C 0006100
6200 C 0006200
6300 C 0006300
6400 C 0006400
6500 C 0006500
6600 C 0006600
6700 C 0006700
6800 C 0006800
6900 C 0006900
7000 C 0007000
7100 C 0007100
7200 C 0007200
7300 C 0007300
7400 C 0007400
7500 C 0007500
7600 C 0007600
7700 C 0007700
7800 C 0007800
7900 C 0007900
8000 C 0008000
8100 C 0008100
8200 C 0008200
8300 C 0008300
8400 C 0008400
8500 C 0008500
8600 C 0008600
8700 C 0008700
8800 C 0008800
8900 C 0008900
9000 C 0009000
9100 C 0009100
9200 C 0009200
9300 C 0009300
9400 C 0009400
9500 C 0009500
9600 C 0009600
9700 C 0009700
9800 C 0009800
9900 C 0009900
10000 C 0010000

```

PROGRAMA  
EN ESTA SUBROUTINA SE CALCULAN LAS FRECUENCIAS NATURALES Y LAS FORMAS MODALES DE LA ESTRUCTURA EN CUESTION. ESTA SUBROUTINA CONSTA DE UN PROGRAMA PRINCIPAL QUE LLAMA A LAS SUBROUTINAS DE DONDE SE TRANSFORMA EL PROBLEMA DE LA FORMA GENERAL A LA FORMA ESTANDAR POR EL METODO DE LANCZOS, Y DE DONDE SE CALCULAN LOS VALORES CARACTERISTICOS POR EL METODO DE LA SECCION. LOS VECTORES CARACTERISTICOS SE CALCULAN POR EL METODO DE ITERACION INVERSA.  
LAS MATRICES DE RIGIDEZ Y DE MASAS ENTREGAN A LA SUBROUTINA DE AQUIENDO AL SUBROUO DE ALGORITMO DE BENDIGO 1.

REFERENCIA  
1. J. LAM, DEREY, JR., AND DAVID M. YORRIDA  
"AN EIGENVALUE PROBLEM FOR BANDED MATRICES",  
COMPUTERS & MATHEMATICS, VOL. 1, PP. 251-262,  
PERGAMON PRESS, 1971, PRINTED IN GREAT BRITAIN.  
2. J. LAM, DEREY, JR., AND DAVID M. YORRIDA  
"AN EIGENVALUE PROBLEM FOR BANDED MATRICES",  
COMPUTERS & MATHEMATICS, VOL. 1, PP. 251-262,  
PERGAMON PRESS, 1971, PRINTED IN GREAT BRITAIN.  
3. J. LAM, DEREY, JR., AND DAVID M. YORRIDA  
"AN EIGENVALUE PROBLEM FOR BANDED MATRICES",  
COMPUTERS & MATHEMATICS, VOL. 1, PP. 251-262,  
PERGAMON PRESS, 1971, PRINTED IN GREAT BRITAIN.

--- ENTRADA DE VARIABLES ---  
AKI(NECS,NR) = MATRIZ DE RIGIDEZ EN BANDA  
AM1(NECS,IPN) = MATRIZ DE MASAS EN BANDA  
NECS = NUMERO DE ECUACIONES  
NR = ANCHO DE BANDA DE LA MATRIZ DE RIGIDEZ  
IPN = ANCHO DE BANDA DE LA MATRIZ DE MASAS  
MI = VALOR CARACTERISTICO MENOR A ENCONTRAR  
ME = VALOR CARACTERISTICO MAYOR A SER ENCONTRADO

--- SALIDA ---  
W(ME) = VECTOR QUE CONTIENE LAS FRECUENCIAS NATURALES  
ZZ(NECS,ME) = MATRIZ QUE CONTIENE LAS FORMAS MODALES

COMMON /TRES/ MI,ME  
COMMON /CUATRO/ W(100)  
DIMENSION R(150,150),R1(150,150),R2(150,150),R3(150,150),R4(150,150)  
R5(150,150),Z(150,150),Z1(150,150),Z2(150,150),Z3(150,150),Z4(150,150)  
R5AM1(NECS,NR)  
WRITE(6,1000)  
CALL DRSOL(NECS,NR,IPN,AKI,AM1,MI,ME,R,R1,R2,R3,R4,R5)  
IF(NALPHA.EQ.1) GO TO 200  
CALL TRISE(NECS,NR)  
WRITE(6,2000) MI,ME  
CALL TRINIT(NECS,NR,C,R)  
WRITE(6,4000)  
DO 400 I=MI,ME  
U1=MI\*ME+I  
N(I)=SORT(N(I),I)

5800		WRITE(6,5000) I,W(11)	00005800
5900	400	CONTINUE	00005900
6000		KK=0.	00006000
6100		DO 600 J=1,MF	00006100
6200		DO 700 I=1,NECS	00006200
6300	700	ZK(I,J)=Z(KK*NECS(I))	00006300
6400	600	KK=KK+1	00006400
6500		CALL ASF(NECS,ZI,R,ZZ)	00006500
6600		MM=MF-MJ+1	00006600
6700		DO 800 II=1,MM	00006700
6800		I=MM-II+1	00006800
6900		CALL UNIDAD(ZZ,NECS,I)	00006900
7000	500	CONTINUE	00007000
7100		WRITE(6,6000)	00007100
7200		WRITE(6,7000) (I,I=1,MM)	00007200
7300		GO TO 900	00007300
7400		WRITE(6,8000)	00007400
7500		WRITE(6,7000) (I,I=1,MM)	00007500
7600		WRITE(6,9000)	00007600
7700		DO 900 K=1,NECS	00007700
7800		K3=(UNIMP*2+K(1))/2	00007800
7900		IF(MOD(K,2).EQ.0) GO TO 250	00007900
8000		WRITE(6,9500) K3,(ZZ(K,MM-I+1),I=1,MM)	00008000
8100		GO TO 200	00008100
8200	950	WRITE(6,9400) K3,(ZZ(K,MM-I+1),I=1,MM)	00008200
8300	900	CONTINUE	00008300
8400		GO TO 500	00008400
8500	200	CONTINUE	00008500
8600		WRITE(6,9700)	00008600
8700	1000	FORMAT(///25X,"RESULTADOS DEL PROBLEMA DE VALORES CARACTERISTICOS"	00008700
8800		*,/24X,52(" ")///)	00008800
8900	2000	FORMAT(10X,"MI =" ,15,20X,"MF =" ,15///)	00008900
9000	4000	FORMAT(//25X,"FRECUENCIAS NATURALES NORMALIZADAS",//24X,	00009000
9100		"RESPECTO A V5/H")	00009100
9200	5000	FORMAT(/35X,"W(" ,12," )=" ,F9,2)	00009200
9300	6000	FORMAT(///10X,"VECTOR CARACTERISTICO PARA CADA MODO DE VIBRACION	00009300
9400		*TRANSVERSAL",/9X,43(" ")///)	00009400
9500	7000	FORMAT(3X,"MODO",3X,10(12,3X))	00009500
9600	8000	FORMAT(30X,"VECTORES CARACTERISTICOS",///)	00009600
9700	9000	FORMAT(//)	00009700
9800	9500	FORMAT(/14,3H ,11,3X,10F12,6)	00009800
9900	9600	FORMAT(/14,3H ,11,3X,10F12,6)	00009900
10000	9700	FORMAT(30X,"MATRIZ MALA")	00010000
10100		RETURN	00010100
10200		END	00010200
10300	C		00010300
10400		SUBROUTINE HBSO (N,MA,MB,A,DEL,A,FIL,ETA,R,MM,N,II)	00010400
10500	C		00010500
10600	C		00010600
10700	C	ESTA SUBROUTINA REDUCE EL PROBLEMA DE VALORES CARACTERISTICOS	00010700
10800	C	DE LA FORMA NO ESTANDAR (A**+LAMBDA**X) A LA FORMA	00010800
10900	C	(T**+LAMBDA**Y) DONDE T ES TRIANGULAR EMPLEANDO EL METODO	00010900
11000	C	GENERAL DE LANZOS.	00011000
11100	C		00011100
11200	C	PARAMETROS GLOBALES	00011200
11300	C		00011300
11400	C	N = ORDEN DE LAS MATRICES BANDADAS	00011400
11500	C	MA = ANCHO DE LA BANDA SUPERIOR DE LA MATRIZ A	00011500
11600	C	MB = ANCHO DE LA BANDA SUPERIOR DE LA MATRIZ B	00011600
11700	C	A = BANDA SUPERIOR DE LA MATRIZ A	00011700
11800	C	B = BANDA SUPERIOR DE LA MATRIZ B	00011800

11800	D	F = MATRIZ DESCOMPLETA D	00011800
11900	C	ALPH = DIAGONAL PRINCIPAL DE F	00011900
12000	C	BETA = SUBDIAGONAL DE F	00012000
12100	C	R = MATRIZ DE TRANSFORMACION	00012100
12200	C	W = SALIDA NO ESTANDAR	00012200
12300	C	W.U = VECTORES DE ALMACENAMIENTO TEMPORAL	00012300
12400	C		00012400
12500		DIMENSION A(N,NA),L(N,MB)	00012500
12600		REAL F(150,1),R(150,1),W(1),U(1),ALPH(1),BETA(1)	00012600
12700		REAL MB,NM,TD,DCORT	00012700
12800			00012800
12900	C	SELECCIONA LA COLUMNA INICIAL DE R,DETERMINA ALPH(1) Y	00012900
13000	C	DESCOMPONE D POR EL METODO DE CHOLSKY	00013000
13100	C		00013100
13200		NM1=N-1	00013200
13300		R(1,1)=1.0/SQR(R(1,1))	00013300
13400		DO 100 J=2,N	00013400
13500	100	R(1,J)=0.0	00013500
13600		ALPH(1)=A(1,1)/R(1,1)	00013600
13700		BETA(N)=0.0	00013700
13800		CALL DDCOMP(N,MB,R,F,NN)	00013800
13900		IF(NN.EQ.1) GO TO 1400	00013900
14000		CALL DMULT(N,MA,A,R(1,1),W)	00014000
14100		CALL DMULT(N,MB,B,R(1,1),U)	00014100
14200		DO 200 J=1,N	00014200
14300	200	W(J)=W(J)-ALPH(1)*U(J)	00014300
14400		DO 300 I=2,N	00014400
14500		IM1=I-1	00014500
14600		CALL BSOL1(N,MB,F,W)	00014600
14700		SUM=0.0	00014700
14800		DO 150 J=1,N	00014800
14900		TD=W(J)	00014900
15000	150	SUM=SUM+TD*TD	00015000
15100		SUM=DSORT(SUM)	00015100
15200		BETA(IM1)=SUM	00015200
15300		IF(SUM.LE.0.0) GO TO 400	00015300
15400		DO 300 J=1,N	00015400
15500	300	R(1,J)=W(J)/SUM	00015500
15600		GO TO 200	00015600
15700	400	DO 500 J=1,N	00015700
15800	500	R(1,J)=0.0	00015800
15900		WRITE(6,111)	00015900
16000	111	FORMAT(" PASO")	00016000
16100		R(1,1)=1./SQRT(R(1,1))	00016100
16200	200	CALL BSOL2(N,MB,F,R(1,1))	00016200
16300	C		00016300
16400	C	EL SIGUIENTE ELEMENTO DE BETA Y LA SIGUIENTE COLUMNA DE R VAN	00016400
16500	C	SIENDO CALCULADOS. LAS COLUMNAS DE R DEBEN SER REORDENADAS	00016500
16600	C	PARA MINIMIZAR EL ERROR	00016600
16700	C		00016700
16800		CALL DMULT(N,MB,R,R(1,1),W)	00016800
16900		DO 600 J=1,IM1	00016900
17000		SUM=0.	00017000
17100		DO 700 K=1,N	00017100
17200		TD=W(K)	00017200
17300	700	SUM=SUM+TD*R(1,K)	00017300
17400		DO 600 K=1,N	00017400
17500	600	R(1,K)=R(1,K)-SUM/R(1,1)	00017500
17600		CALL DMULT(N,MB,R,R(1,1),W)	00017600
17700		SUM=0.	00017700

```

17500      DO 700 J=1,N
17600      TD=R(J,I)
17700      SUM=SUM+W(J)*TD
17800      SUM=DSORT(SUM)
17900      DO 1000 J=1,N
18000      R(J,I)=R(J,I)/SUM
18100      C
18200      C      CALCULA EL SIGUIENTE ELEMENTO DE ALFA
18300      CALL DMULT(N,MA,A,R(J,I),D)
18400      SUM=0.
18500      DO 1100 J=1,N
18600      TD=R(J,I)
18700      SUM=SUM+W(J)*TD
18800      ALFA(I)=SUM
18900      F
19000      C      REINICIALIZA W PARA EL SIGUIENTE PASO
19100      C
19200      IF (I.EQ.N) RETURN
19300      CALL DMULT(N,NB-R(J,I),D)
19400      DO 1200 J=1,N
19500      W(J)=W(J)-ALFA(I)*R(J,I)
19600      CALL DMULT(N,NS-R(J,I),D)
19700      DO 1300 J=1,N
19800      W(J)=W(J)+BETA(IM)*R(J,I)
19900      1400      RETURN
20000      END
20100      *
20200      SUBROUTINE TRBISE (C,B)
20300      *
20400      C      TRBISE ENCUENTRA LOS VALORES CARACTERISTICOS DE UNA MATRIZ
20500      C      TRIANGULAR SIMETRICA POR EL METODO DE BISECCION
20600      *
20700      DIMENSION C(150),B(150)
20800      LOGICAL IFAIL
20900      COMMON /CLISE/      YMAX,BETA(250),UM(150),N1,FEB,XI,XII,YO
21000      *11,EPG1,EPG2,XMIN,1,1,1,1,1,1,1,1,NM1
21100      COMMON /TRES/  MI,MF
21200      COMMON /CUATRO/ X(100)
21300      NM1=N-1
21400      EPG1=1.0E-011
21500      RELEFEN=1.0E-011
21600      ID=500 N=1,NM1
21700      I=1 J
21800      DO 100 B(I+1)=B(I)
21900      DO 100 J=2,N
22000      10 BETA(I)=B(I)*B(I)
22100      BETA(I)=0.
22200      B(I)=0.
22300      XMIN=C(N)-ABS(B(N))
22400      XMAX=C(N)+ABS(B(N))
22500      DO 100 J=1,NM1
22600      I=N-J
22700      H=ABS(B(I))+ABS(B(I+1))
22800      IF (C(I) > H.GT.XMAX) XMAX=C(I)+H
22900      100 IF (C(I) < -H.LT.XMIN) XMIN=C(I)-H
23000      EPG2=RELEFEN*XMIN
23100      IF (XMAX-XMIN < 1.E-11)
23200      1 EPG2=RELEFEN*XMIN
23300      2 IF (EPG1) 3,3,1

```

```

23000      2 EPS1=EPS2
23100      3 EPS2=0.5*EPS1/7.0*EPS2
24000      X0=XMAX
24100      DO 200 I=M1,M2
24200      X(I)=XMAX
24300      200 W(I)=XMIN
24400      IZ=0
24500      DO 300 J=M1,M2
24600      K=M2+M1-J
24700      XJ=XMIN
24800      DO 400 JJ=M1,K
24900      I=M1+K-JJ
25000      IF(XJ-W(I)) 5,400,400
25100      5 XJ=W(I)
25200      GO TO 6
25300      400 CONTINUE
25400      6 IF(X0.GT.X(K)) X0=X(K)
25500      7 AST=0.4*RELEEN*(ABS(XJ)-ABS(X0))/(EPS1-X0)*XJ
25600      IF(AST) 8,300,300
25700      8 X1=(XJ+X0)/2.
25800      CALL STISEQ(N,C,B)
25900      GO TO 7
26000      300 X(K)=(X0+X1)/2.
26100      DO 400 I=M1,M2
26200      W(I)=X(I)
26300      DO 700 J=M1,M2
26400      Y(I)=W(I)+M2-1)
26500      DO 800 I=1,NM1
26600      800 B(I)=B(1+I)
26700      B(N)=0.
26800      RETURN
26900      END
C
27000      C
27100      SUBROUTINE TRINIT(N,Z,C,R)
C
27200      C
27300      C
27400      C
27500      C
27600      C
27700      C
27800      C
27900      C
28000      C
28100      C
28200      C
28300      C
28400      C
28500      C
28600      C
28700      C
28800      C
28900      C
29000      C
29100      C
29200      C
29300      C
29400      C
29500      C
29600      C
29700      C
DIMENSION Z(1),C(1),B(1)
COMMON/CUSE/ GORM,P(250),AM(250),B(250),FAM
*ITN(250),X(252),SLAM,EPS,ILV,B1,BIL,BLTA,NI,LL,J,II
COMMON /TRES/M3,M2
COMMON /CHATRO/ W(100)
FAM=1.0E-011
M1=M2-M3+1
N1=N-1
EPS=FAM*GORM
ETA=1./N
DO 1 J=M3,M2
SLAM=GORM-EPS
IF(W(J).LT.SLAM) SLAM=W(J)

```

29800	H=C(I)-SLAM	00029800
29900	V=R(I)	00029900
30000	IF(V.EQ.O.) V=EPS	00030000
30100		00030100
30200	PARA ASEGURAR LA INDEPENDENCIA EN CASOS DEGENERADOS	00030200
30300		00030300
30400	DO 2 I=1,N	00030400
30500	R1=R(I)	00030500
30600	IF(B1.EQ.O.) B1=EPS	00030600
30700	R11=R(I+1)	00030700
30800	IF(R11.EQ.O.) R11=EPS	00030800
30900	IF(ABS(B1).LE.ABS(U)) GO TO 4	00030900
31000	3 AM(I+1)=U/B1	00031000
31100	IF((AM(I+1).EQ.O.).AND.(B1.LE.EPS)) AM(I+1)=1.	00031100
31200		00031200
31300	PARA EVITAR EL FACTOR CERO EN EL CASO DESCOMPUESTO	00031300
31400		00031400
31500	P(I)=B1	00031500
31600	Q(I)=C(I+1)-SLAM	00031600
31700	R(I)=B11	00031700
31800	U=V-AM(I+1)*Q(I)	00031800
31900	V=-AM(I+1)*R(I)	00031900
32000	TIN(I+1)=1.	00032000
32100	GO TO 2	00032100
32200	4 AM(I+1)=R1/U	00032200
32300	P(I)=U	00032300
32400	Q(I)=V	00032400
32500	R(I)=0.	00032500
32600	U=C(I+1)-SLAM-AM(I+1)*V	00032600
32700	V=B11	00032700
32800	TIN(I+1)=-1.	00032800
32900	2 CONTINUE	00032900
33000	P(N)=U	00033000
33100	Q(N)=0.	00033100
33200	R(N)=0.	00033200
33300	X(N+1)=0.	00033300
33400	X(N+2)=0.	00033400
33500	H=0.	00033500
33600	DO 5 I1=1,N	00033600
33700	I=N+1-I1	00033700
33800	H=ETA-Q(I)*X(I+1)-R(I)*X(I+2)	00033800
33900	IF(P(I).EQ.O.) X(I)=H/EPS	00033900
34000	IF(P(I).NE.O.) X(I)=H/P(I)	00034000
34100	H=H+ABS(X(I))	00034100
34200	5 CONTINUE	00034200
34300	H=1./H	00034300
34400	DO 6 I=1,N	00034400
34500	6 Y(I)=X(I)*H	00034500
34600	DO 7 I=2,N	00034600
34700	IF(TIN(I).LE.O.) GO TO 8	00034700
34800	U=Y(I-1)	00034800
34900	X(I-1)=X(I)	00034900
35000	X(I)=U-AM(I)*X(I-1)	00035000
35100	GO TO 7	00035100
35200	8 X(I)=X(I)-AM(I)*X(I-1)	00035200
35300	7 CONTINUE	00035300
35400	H=0.	00035400
35500	DO 2 I1=1,N	00035500
35600	I=N+1-I1	00035600
35700	U=Y(I)-Q(I)*X(I+1)-R(I)*X(I+2)	00035700

```

35800      IF (P(I).EQ.0.) X(I)=H/EPS
35900      IF (P(I).NE.0.) X(I)=H/P(I)
36000      H=H*X(I)*X(I)
36100      ? CONTINUE
36200      H=1./SQRT(H)
36300      DO 100 I=1,N
36400 100 X(I)=X(I)*H
36500      H=0.
36600      DO 101 II=1,N
36700      II=II-1
36800      H=X(II)-G(II)*X(1)+F(II)*X(1+2)
36900      IF (P(II).EQ.0.) X(II)=H/EPS
37000      IF (P(II).NE.0.) X(II)=H/P(II)
37100      H=H+X(II)*X(II)
37200 101 CONTINUE
37300      H=1./SQRT(H)
37400      DO 11 I=1,N
37500 11 Z(I+(J-1)*N)=X(I)*H
37600      ? CONTINUE
37700      RETURN
37800      END
37900      C
38000      C SUBROUTINE ASF(N,ZK,R,ZZ)
38100      C
38200      COMMON /TRES/M1,M2
38300      DIMENSION ZK(150,1),R(150,1),ZZ(150,1)
38400      DO 2 J=1,N
38500      DO 2 K=1,M2
38600      ZZ(J,K)=0.
38700      DO 1 K=1,N
38800      1 ZZ(J,K)=ZZ(J,K)+R(1,K)*ZK(K,J)
38900      2 CONTINUE
39000      RETURN
39100      END
39200      C
39300      C SUBROUTINE UNIDAD(ZZ,N,II)
39400      C
39500      C NORMALIZA LAS AMPLITUDES DEL VECTOR MODAL CON RESPECTO AL VALOR
39600      C MAXIMO ENCONTRADO EN DICHO VECTOR
39700      C
39800      DIMENSION ZZ(150,150)
39900      I=1
40000      AMAY=ABS(ZZ(I,1))
40100      DO 1 K=2,N
40200      IF (AMAY.GE.ABS(ZZ(I,K))) GO TO 1
40300      I=K
40400      AMAY=ABS(ZZ(K,1))
40500      ? CONTINUE
40600      F=1./ZZ(I,1)
40700      DO 2 K=1,N
40800      2 ZZ(K,1)=ZZ(K,1)*F
40900      RETURN
41000      END
41100      C
41200      C SUBROUTINE BDCOMP(N,IBW,A,UBALFA)
41300      C
41400      C ESTA SUBROUTINA DESCOMPONE UNA MATRIZ SIMETRICA POSITIVA DEFINIDA A
41500      C LA FORMA (A-HB*H) DONDE H ES UNA MATRIZ DE BANDA SUPERIOR
41600      C
41700      DIMENSION A(N,160)

```

```

00035800
00035900
00036000
00036100
00036200
00036300
00036400
00036500
00036600
00036700
00036800
00036900
00037000
00037100
00037200
00037300
00037400
00037500
00037600
00037700
00037800
00037900
00038000
00038100
00038200
00038300
00038400
00038500
00038600
00038700
00038800
00038900
00039000
00039100
00039200
00039300
00039400
00039500
00039600
00039700
00039800
00039900
00040000
00040100
00040200
00040300
00040400
00040500
00040600
00040700
00040800
00040900
00041000
00041100
00041200
00041300
00041400
00041500
00041600
00041700

```

41800	REAL U(150,1)	00041800
41900	REAL *8 SUM,DSORT,DD	00041900
42000	INTEGER P,Q	00042000
42100	DO 400 I=1,N	00042100
42200	P=N-1+I	00042200
42300	IF(IRW.LT.P) P=IRW	00042300
42400	DO 400 J=1,P	00042400
42500	Q=IRW-1	00042500
42600	IF(1-I.LT.Q) Q=1-I	00042600
42700	SUM=A(I,J)	00042700
42800	IF(Q.LT.1) GO TO 200	00042800
42900	DO 100 K=1,Q	00042900
43000	UD=U(I-K,J+K)	00043000
43100	100 SUM=SUM-U(I-K,J+K)*UD	00043100
43200	200 IF(J.NE.1) GO TO 300	00043200
43300	IF(SUM.GT.0.) GO TO 201	00043300
43400	NAIFA=1	00043400
43500	RETURN	00043500
43600	201 TEMP=1.0/DSORT(SUM)	00043600
43700	U(I,J)=TEMP	00043700
43800	GO TO 400	00043800
43900	300 U(I,J)=SUM*TEMP	00043900
44000	400 CONTINUE	00044000
44100	RETURN	00044100
44200	END	00044200
44300		00044300
44400	C SUBROUTINE DMULT(N,IRW,A,X,H)	00044400
44500		00044500
44600	C ESTA SUBROUTINA MULTIPLICA UNA MATRIZ SIMETRICA BANDA POR UN	00044600
44700	C VECTOR. SOLO SE ALMACENA LA PORCION DE LA BANDA SUPERIOR DE LA	00044700
44800	C MATRIZ	00044800
44900		00044900
45000	DIMENSION A(N,IRW)	00045000
45100	REAL X(1),B(1)	00045100
45200	REAL *8 SUM,XD	00045200
45300	N1=N-IRW+1	00045300
45400	DO 300 I=1,N	00045400
45500	SUM=0.	00045500
45600	J1=IRW	00045600
45700	IF(I.GT.N1) J1=N-1+I	00045700
45800	DO 100 J=1,J1	00045800
45900	XD=X(J)-1	00045900
46000	100 SUM=SUM+A(I,J)*XD	00046000
46100	IF((I.EQ.1).OR.(IRW.EQ.1)) GO TO 300	00046100
46200	J1=I-IRW+1	00046200
46300	J2=J1+IRW-2	00046300
46400	IF(I.GT.IRW) GO TO 100	00046400
46500	J1=1	00046500
46600	J2=I-1	00046600
46700	100 DO 200 J=J1,J2	00046700
46800	XD=X(J)	00046800
46900	200 SUM=SUM+A(J,I-J1)*XD	00046900
47000	300 B(I)=SUM	00047000
47100	RETURN	00047100
47200	END	00047200
47300		00047300
47400	C SUBROUTINE RESOLV(D,THU,II,0)	00047400
47500		00047500
47600	C EN ESTA SUBROUTINA SE RESUELVE UN CONJUNTO DE ECUACIONES	00047600
47700	C ALGEBRAICAS SIMULTANEAS DE LA FORMA C*(X-D)+Y, DONDE D	00047700

47800	C	ES UNA MATRIZ TRIANGULAR INFERIOR BANDA 2	00047800
47900	C		00047900
48000		REAL H(150,1),R(1)	00048000
48100		REAL *B SUM,BD	00048100
48200		DO 100 I=1,N	00048200
48300		J=1+IBW-I	00048300
48400		IF(I,J).E.1)B=.	00048400
48500		SUM=B(I)	00048500
48600		K=I-1	00048600
48700		IF(I.GT.K)GO TO 200	00048700
48800		DO 100 K=I,K1	00048800
48900		BD=B(K)	00048900
49000	100	SUM=SUM-H(B,I-K)*B	00049000
49100	200	R(I)=SUM*H(I,I)	00049100
49200		RETURN	00049200
49300		END	00049300
49400	C		00049400
49500		SUBROUTINE DSOL(N,IRW,II,YS)	00049500
49600	C		00049600
49700	C	EN ESTA SUBROUTINA SE RESUELVE UN CONJUNTO DE ECUACIONES	00049700
49800	C	ALGEBRAICAS SIMULTANEAS DE LA FORMA Y=AY+X, DONDE Y ES	00049800
49900	C	UNA MATRIZ TRIANGULAR SUPERIOR BANDA 2	00049900
50000	C		00050000
50100		REAL U(150,1),Y(1)	00050100
50200		REAL *B SUM,YD	00050200
50300		DO 200 II=1,N	00050300
50400		I=N-II+1	00050400
50500		J=1+IBW-I	00050500
50600		IF(I.GT.N)J=N	00050600
50700		SUM=Y(I)	00050700
50800		K2=I+1	00050800
50900		IF(K2.GT.I)GO TO 200	00050900
51000		DO 100 K=K2,I	00051000
51100		XD=Y(K)	00051100
51200	100	SUM=SUM-H(I,K)*XD	00051200
51300	200	X(I)=SUM*H(I,I)	00051300
51400		RETURN	00051400
51500		END	00051500
51600	C		00051600
51700		SUBROUTINE STUSP(H,C,R)	00051700
51800	C		00051800
51900	C	ADHI SE CUENTA EL NUMERO DE CAMBIOS EN EL SIGNO DE LOS	00051900
52000	C	MEMBROS SUCESIVOS.	00052000
52100	C		00052100
52200		DIMENSION C(1),R(1)	00052200
52300		COMMON/CLSF/ XNAX,I4 (A(150),D(150),R(150),X1,II,XY)	00052300
52400		COMMON /TRES/MI,MI	00052400
52500		COMMON /CUATRO/ X(100)	00052500
52600		I2=I2+1	00052600
52700		IA=0	00052700
52800		DO 100 I=1,N	00052800
52900		IF(C)1,2,1	00052900
53000	2	D=C(I)-X1-ABS(C(I-1))*MI	00053000
53100		GO TO 3	00053100
53200	1	D=C(I)-X1-BETA*(I-1)	00053200
53300	3	IF(D)4,100,100	00053300
53400	4	IA=IA+1	00053400
53500	100	CONTINUE	00053500
53600			00053600
53700			00053700



PROGRAMA PARA RESOLVER EL PROBLEMA DE VALORES Y VEC -  
TORES CARACTERISTICOS PARA MATRICES SIMETRICAS, PARA  
EL NUMERO DE VALORES REQUERIDOS POR EL METODO DE  
BISECCION

```

4000 SUBROUTINE BISECC(A,B,D,DU,E,EIGEN,FL,E2,Z,W,T,N,K),M,N?,TOL      00004000
4100 * ,DLAM,EPS,CHEPS)                                           00004100
4200                                                                00004200
4300 PROGRAMA                                                     00004300
4400 RESUELVE EL PROBLEMA DE VALORES CARACTERISTICOS PARA MATRICES 00004400
4500 SIMETRICAS, PARA EL NUMERO DE VALORES REQUERIDOS.           00004500
4600 EN ESTE PROGRAMA SE REDUCE EL PROBLEMA DE LA FORMA GENERAL A 00004600
4700 LA FORMA ESTANDAR, LA MATRIZ DE RIGIDEZ SE TRIDIAGONALIZA POR 00004700
4800 EL METODO DE HOUSEHOLDER, LOS VALORES CARACTERISTICOS SE OBTI- 00004800
4900 NEN POR EL METODO DE BISECCION, LOS VECTORES CARACTERISTICOS 00004900
5000 SE OBTIENEN POR EL METODO DE ITERACION INVERSA.             00005000
5100                                                                00005100
5200 REFERENCIA                                                    00005200
5300 1-G.M.L. GLADWELL AND D.C. TONDILLO                            00005300
5400 "THE ALGEBRAIC EIGENVALUE PROBLEM"                          00005400
5500 -A SET OF FORTRAN SUBROUTINES",                              00005500
5600 SOLID MECHANICS DIVISION UNIVERSITY OF WATERLOO,            00005600
5700 REPORT NO. 18 MARCH, 1972.                                    00005700
5750                                                                00005750
5800 2-G.M.L. GLADWELL                                              00005800
5900 "THE ALGEBRAIC EIGENVALUE PROBLEM"                            00005900
6000 A COMMENTARY ON SOME COMPUTERS PROGRAMS",                    00006000
6100 SOLID MECHANICS DIVISION UNIVERSITY OF WATERLOO,            00006100
6200 REPORT NO. 25 MAY, 1973.                                      00006200
6300                                                                00006300
6400 LA LLAMADA A LA SUBROUTINA ES HECHA POR:                      00006400
6500 CALL BISECC(A,B,N,K,E,M,N?,TOL,CHEPS)                         00006500
6600                                                                00006600
6700 -- ENTRADA DE VARIABLES --                                     00006700
6800 A(N,N) = MATRIZ DE RIGIDEZ DE LA ESTRUCTURA                  00006800
6900 B(N,N) = MATRIZ DE MASAS DE LA ESTRUCTURA.                  00006900
7000 KI=0 SI A ES POSITIVA DEFINIDA, =1 SI NO LO ES.              00007000
7100 M      = NO. DE VALORES CARACTERISTICOS REQUERIDOS.         00007100
7200 M2     = NO. DE VECT. CARACTERISTICOS REQUERIDOS.            00007200
7300 TOL Y DLAM PUEDEN SER CERO.                                   00007300
7400 CHEPS APARECE EN EIO.                                        00007400
7500                                                                00007500
7600 -- SALIDA --                                                 00007600
7700 Z(N,N) = MATRIZ QUE CONTIENE LOS VECT. CARACTERISTICOS.     00007700
7800 EIGEN(N) = VECTOR QUE CONTIENE LOS VALORES CARACTERISTICOS 00007800
7900                                                                00007900
8000                                                                00008000
8100 IMPLICIT REAL*(B(A-H,O-Z))                                    00008100
8200 DIMENSION A(N,N),B(N,N),D(N),DU(N),E(N),EIGEN(N),EL(N)      00008200
8300 *,E2(N),Z(N,N),W(N),T(N)                                     00008300
8400                                                                00008400
8500 CALL REDUC(A,B,DI,N,NX)                                       00008500
8600                                                                00008600
8700 CALL TRID(A,D,E,E2,TOL,INORM,N,NX)                            00008700
8800                                                                00008800
8900 CALL BISEC(D,E,EIGEN,EPS,CHEPS,M,N)                           00008900
9000                                                                00009000
9100 CALL EIO(D,E,EIGEN,Z,CHEPS,INORM,M2,N,NX)                    00009100
9200                                                                00009200
9300 CALL BACK(A,E,Z,M2,N,NX)                                       00009300
9400                                                                00009400
9500 CALL TRANS(B,DL,Z,M2,N,NX)                                     00009500

```

9600	RETURN	00002400
9700	END	00002700
9800	SUBROUTINE BTSE(D,B2,EE,FV,RELF,EPST,MM,N)	00002800
9900		00002900
10000	CALCULA LOS VALORES CARACTERISTICOS.	00010000
10100		00010100
10200	IMPLICIT REAL*(8(A-H,D-Z))	00010200
10300	DIMENSION EE(N),B2(N),D(N),EV(N)	00010300
10400	DIMENSION WJ(200),X(200)	00010400
10500	EE(1)=0	00010500
10600	XMIN=D(N)-DABS(EE(N))	00010600
10700	XMAX=D(N)+DABS(EE(N))	00010700
10800	I,J=N-1	00010800
10900	I=N	00010900
11000	DO 1 II=1,I-1	00011000
11100	I=I-1	00011100
11200	H=DABS(EE(I))+DABS(EE(II+1))	00011200
11300	IF((D(I)+H).GT.XMAX)XMAX=D(I)+H	00011300
11400	IF((D(I)-H).LT.XMIN)XMIN=D(I)-H	00011400
11500	1 CONTINUE	00011500
11600	X=XMIN+XMAX	00011600
11700	IF(XM.GT.0)EPS2=RELF*XMAX	00011700
11800	IF(XM.LE.0)EPS2=-RELF*XMIN	00011800
11900	IF(EPST.LE.0)EPS1=EPS2	00011900
12000	EPS2=0.5#EPS1+7#EPS2	00012000
12100	XO=XMAX	00012100
12200	DO 2 I=1,MM	00012200
12300	X(I)=XMAX	00012300
12400	WJ(I)=XMIN	00012400
12500	2 CONTINUE	00012500
12600	K=MM+1	00012600
12700	DO 3 KK=1,MM	00012700
12800	K=K-1	00012800
12900	XII=XMIN	00012900
13000	I=K+1	00013000
13100	DO 4 II=1,K	00013100
13200	I=I-1	00013200
13300	IF(XI.GE.WJ(II))GO TO 4	00013300
13400	XI=WJ(II)	00013400
13500	GO TO 5	00013500
13600	4 CONTINUE	00013600
13700	5 IF(XO.GT.X(K))XO=X(K)	00013700
13800	XX=XO-XII	00013800
13900	XXX=2#RELF*(DABS(XII)+DABS(XO))+EPST	00013900
14000	XY=XX-XXX	00014000
14100	IF(XY)16,16,7	00014100
14200	7 XJ=(XII+XO)/2	00014200
14300	JA=0	00014300
14400	Q=1.0	00014400
14500	DO 8 I=1,N	00014500
14600	IF(D,NE.0)GO TO 14	00014600
14700	IF(D,EO.0)Q=D(I)-XJ-DABS(EE(I)/RELF)	00014700
14800	GO TO 15	00014800
14900	14 Q=D(I)-XJ-R2(I)/Q	00014900
15000	15 CONTINUE	00015000
15100	IF(D,LT.0)IA=IA+1	00015100
15200	8 CONTINUE	00015200
15300	IF(IA,GE,K)GO TO 9	00015300
15400	IF(IA,GE,1)GO TO 11	00015400
15500	XII=XJ	00015500

15600	WU(1)=X1	00015600
15700	GO TO 1212	00015700
15800	11 WU(IA+1)=X1	00015800
15900	XU=WU(IA+1)	00015900
16000	IF(X(IA).GT.XU)X(IA)=XU	00016000
16100	1212 K=K	00016100
16200	GO TO 10	00016200
16300	9 XO=XJ	00016300
16400	10 XX=XO-XU	00016400
16500	XXX=2*RELF*(DABS(XU)+DABS(XO))+EPSI	00016500
16600	XY=XX-XXX	00016600
16700	C *	00016700
16800	IF(XY)16,16,7	00016800
16900	16 X(K)=(XO+XU)/2	00016900
17000	3 CONTINUE	00017000
17100	DO 13 J=1,MM	00017100
17200	EV(I)=X(I)	00017200
17300	13 CONTINUE	00017300
17400	RETURN	00017400
17500	END	00017500
17600	SUBROUTINE REDUC(A,D,DL,N,NX)	00017600
17700	7700	00017700
17800	REDUC EL PROBLEMA A LA FORMA GENERAL.	00017800
17900	7900	00017900
18000	IMPLICIT REAL*8(A-H,O-Z)	00018000
18100	DIMENSION A(NX,NX),B(NX,NX),DL(NX)	00018100
18200	IF(N)1,2,2	00018200
18300	1 N=-N	00018300
18400	GO TO 3	00018400
18500	2 DO 8 I=1,N	00018500
18600	DO 8 J=1,N	00018600
18700	X=R(I,J)	00018700
18800	WRITE(6,*) X	00018800
18900	(J,I=1	00018900
19000	K=I	00019000
19100	IF(.L.EQ.0)GO TO 16	00019100
19200	DO 6 KK=1,JI	00019200
19300	K=K-1	00019300
19400	6 X=X-R(I,K)*B(I,K)	00019400
19500	16 K=K	00019500
19600	IF(.L.NE.0)GO TO 7	00019600
19700	IF(X(.E.0.0)GO TO 999	00019700
19800	Y=DSORT(X)	00019800
19900	DL(I)=DSORT(X)	00019900
20000	GO TO 8	00020000
20100	7 R(I,I)=X/Y	00020100
20200	8 CONTINUE	00020200
20300	3 DO 9 I=1,N	00020300
20400	II=I-1	00020400
20500	Y=DL(I)	00020500
20600	DO 10 J=1,N	00020600
20700	X=A(I,J)	00020700
20800	K=I	00020800
20900	IF(.L.EQ.0)GO TO 10	00020900
21000	DO 11 KK=1,II	00021000
21100	K=K-1	00021100
21200	11 X=X-R(I,K)*A(I,K)	00021200
21300	10 A(I,I)=X/Y	00021300
21400	9 CONTINUE	00021400
21500	DO 12 J=1,N	00021500

21600	DO 12 I=J,N	00021600
21700	II=J-1	00021700
21800	JL=J-1	00021800
21900	K=1	00021900
22000	X=A(I,II)	00022000
22100	IF(II.LT.J)GO TO 17	00022100
22200	DO 13 KK=J,II	00022200
22300	K=K-1	00022300
22400	13 X=X-A(K,J)*R(I,K)	00022400
22500	17 K=1	00022500
22600	IF(JL.EQ.0)GO TO 18	00022600
22700	DO 14 KK=1,JL	00022700
22800	K=K-1	00022800
22900	14 X=X-A(J,K)*R(I,K)	00022900
23000	18 A(I,II)=X/DL(I)	00023000
23100	12 CONTINUE	00023100
23200	GO TO 888	00023200
23300	888 PRINT 15	00023300
23400	15 FORMAT(2X,"LA MATRIZ NO ES DEFINIDA POSITIVA")	00023400
23500	888 K=K	00023500
23600	RETURN	00023600
23700	END	00023700
23800	SUBROUTINE TRID(G,D,E,E2,TOL,INORM,N,BX)	00023800
23900		00023900
24000	TRIDIAGONALIZA LA MATRIZ.	00024000
24100		00024100
24200	IMPLICIT REAL*(A,H,O-Z)	00024200
24300	DIMENSION A(NX,NX),D(NX),E(NX),E2(NX)	00024300
24400	DO 1 J=1,N	00024400
24500	1 D(I)=A(I,I)	00024500
24600	I=N+1	00024600
24700	DO 2 II=1,N	00024700
24800	I=II-1	00024800
24900	L=I-1	00024900
25000	H=0	00025000
25100	IF(L.EQ.0)GO TO 14	00025100
25200	DO 1212 K=1,L	00025200
25300	1212 H=H+A(I,K)*A(I,I)	00025300
25400	14 IF(H.GT.TOL)GO TO 4	00025400
25500	E(I)=0	00025500
25600	E2(I)=0	00025600
25700	GO TO 5	00025700
25800	4 E2(I)=H	00025800
25900	F=A(I,I-1)	00025900
26000	IF(F.LT.0)GO TO 6	00026000
26100	F(I)=-DSORT(H)	00026100
26200	G=-DSORT(H)	00026200
26300	GO TO 7	00026300
26400	6 F(I)=DSORT(H)	00026400
26500	G=DSORT(H)	00026500
26600	7 H=H-F*G	00026600
26700	A(I,I-1)=F-G	00026700
26800	F=0	00026800
26900	IF(L.EQ.0)GO TO 15	00026900
27000	DO 8 J=1,L	00027000
27100	G=0	00027100
27200	DO 9 K=1,J	00027200
27300	9 G=G+A(I,K)*A(I,I)	00027300
27400	JL=J+1	00027400
27500	IF(JL.GT.I)GO TO 16	00027500

27600	DO 10 K=1,L	00027600
27700	10 G=F(A(K,I))*A(I,E)	00027700
27800	14 F(I)=G/AI	00027800
27900	G=F(I)	00027900
28000	8 F=F+G*A(I,I)	00028000
28100	15 H=F/(I+H)	00028100
28200	IF(L.EQ.0)GO TO 5	00028200
28300	DO 11 I=1,L	00028300
28400	F=A(I,I)	00028400
28500	F(I)=F(I)+F	00028500
28600	G=F(I)	00028600
28700	DO 12 K=1,L	00028700
28800	12 A(J,K)=A(J,K)-F*F(K)+G*A(I,I)	00028800
28900	11 CONTINUE	00028900
29000	5 H=0(I)	00029000
29100	D(I)=A(I,I)	00029100
29200	A(I,I)=H	00029200
29300	2 CONTINUE	00029300
29400	TNORM=DABS(D(1))+DABS(E(1))	00029400
29500	NI=N-1	00029500
29600	DO 102 J=1,NI	00029600
29700	AI=DABS(E(1))+DABS(E(I+1))+DABS(D(I))	00029700
29800	102 IF(AI.GT.TNORM) TNORM=AI	00029800
29900	RETURN	00029900
30000	END	00030000
30100	SUBROUTINE EIG(C,FE,W,7,CHEPS,NORM,M,N,NX)	00030100
30200		00030200
30300	CALEULA LOS VECTORES CARACTERISTICOS DE LA MATRIZ TRIANGULAR.	00030300
30400		00030400
30500	IMPLICIT REAL*8(A-H,O-Z)	00030500
30600	DIMENSION C(NX),W(NX),FE(NX),7(NX,NX)	00030600
30700	DIMENSION P(200),R(200),X(200),X(200),INT(200)	00030700
30800	REAL*8 LAMDA,NORM,R(200)	00030800
30900	II=N-1	00030900
31000	DO 23 I=1,II	00031000
31100	23 FE(I)=FE(I+1)	00031100
31200	FE(N)=0.	00031200
31300	LAMDA=NORM	00031300
31400	EPS=CHEPS*NORM	00031400
31500	J=M(I)	00031500
31600	DO 1 JJ=1,M1	00031600
31700	1 JJ=J	00031700
31800	LAMDA=LAMDA-EPS	00031800
31900	IF(W(I),LT,LAMDA) LAMDA=W(I)	00031900
32000	H=C(I)-LAMDA	00032000
32100	V=FE(I)	00032100
32200	IF(V.EQ.0)V=EPS	00032200
32300	KK=N-1	00032300
32400	DO 2 I=1,KK	00032400
32500	II=I+1	00032500
32600	R1=FE(I)	00032600
32700	IF(R1.EQ.0)R1=EPS	00032700
32800	R11=FE(II)	00032800
32900	IF(R11.EQ.0)R11=EPS	00032900
33000	IF(DABS(R1).LT,DABS(R11))GO TO 1,12	00033000
33100	M(II)=R1/R1	00033100
33200	IF(M(II).NE.0)GO TO 4	00033200
33300	IF(R11.E.FPS)M(II)=1.0	00033300
33400	4 P(I)=R1	00033400
33500	O(I)=C(I+1)-LAMDA	00033500

C  
E  
I

33600		R(I)=R11	00033600
33700		U=V M(I+1)*Q(I)	00033700
33800		V=-M(I+1)*R(I)	00033800
33900		INT(I+1)=+1	00033900
34000		DO TO 2	00034000
34100	1212	M(I+1)=R1/U	00034100
34200		P(I)=U	00034200
34300		Q(I)=V	00034300
34400		R(I)=0	00034400
34500		U=C(I+1)-LAMDA-M(I+1)*V	00034500
34600		V=R11	00034600
34700		INT(I+1)=-1	00034700
34800	2	CONTINUE	00034800
34900		P(N)=U	00034900
35000		Q(N)=0	00035000
35100		R(N)=0	00035100
35200		X(N+1)=0	00035200
35300		X(N+2)=0	00035300
35400		H=0	00035400
35500		ETA=1.0/N	00035500
35600		I=N+1	00035600
35700		DO 5 I1=1,N	00035700
35800		I=I-1	00035800
35900		U=ETA *Q(I)*X(I+1)-R(I)*X(I+2)	00035900
36000		IF(P(I).EQ.0)X(I)=U/EP5	00036000
36100		IF(P(I).NE.0)X(I)=U/P(I)	00036100
36200	5	H=H+DABS(X(I))	00036200
36300		H=1/H	00036300
36400		DO 6 J=1,N	00036400
36500	6	X(I)=X(I)*H	00036500
36600		DO 7 I=2,N	00036600
36700		IF(INT(I).LE.0)GO TO 8	00036700
36800		U=X(I-1)	00036800
36900		X(I-1)=X(I)	00036900
37000		X(I)=U-M(I)*X(I-1)	00037000
37100		GO TO 7	00037100
37200	8	X(I)=X(I)-M(I)*X(I-1)	00037200
37300	7	CONTINUE	00037300
37400		H=0	00037400
37500		I=N+1	00037500
37600		DO 9 I1=1,N	00037600
37700		I=I-1	00037700
37800		U=X(I)-Q(I)*X(I+1)-R(I)*X(I+2)	00037800
37900		IF(P(I).EQ.0)X(I)=U/EP5	00037900
38000		IF(P(I).NE.0)X(I)=U/P(I)	00038000
38100	9	H=H+X(I)*X(I)	00038100
38200		H=1./DSQRT(H)	00038200
38300		DO 10 I=1,N	00038300
38400	10	Z(I,1)=X(I)*H	00038400
38500		1 CONTINUE	00038500
38600			00038600
38700		CAMBIA LOS INDICES DE LA SUBROUTINA HACI	00038700
38800			00038800
38900		NN=N-1	00038900
39000		I=N	00039000
39100		DO 22 I1=1,NN	00039100
39200		I=I-1	00039200
39300	22	FE(I1)=FE(I)	00039300
39400		FE(I)=0.	00039400
39500		RETURN	00039500

```

39600      END
39700      SUBROUTINE BACK(A,FF,Z,M,N,NX)
39800      C
39900      C   CALCULA LOS VECTORES CARACTERISTICOS DE LA MATRIZ REDUCIDA.
40000      C
40100      IMPLICIT REAL*(A,H,O,Z)
40200      DIMENSION Z(NX,NX),FE(NX),A(NY,NY)
40300      DO 1 J=2,N
40400      IF (FF(1).EQ.0.0)GO TO 1
40500      L=J-1
40600      H=FE(1)*A(L,L-1)
40700      DO 2 J=1,M
40800      S=0.0
40900      DO 1212 K=1,L
41000      1212 S=S+A(L,K)*Z(K,J)
41100      S=S/H
41200      DO 4 K=1,L
41300      4 Z(K,J)=Z(K,J)+S*A(L,J)
41400      7 CONTINUE
41500      1 CONTINUE
41600      RETURN
41700      END
41800      SUBROUTINE TRANS(B,DL,Z,M,N,NX)
41900      C
42000      C   CALCULA LOS VECTORES CARACT. DEL PROBLEMA ORIGINAL.
42100      C
42200      IMPLICIT REAL*(A,H,O,Z)
42300      DIMENSION Z(NY,NX),DL(NY),B(NY,NY)
42400      DO 1 J=1,M
42500      I=N+1
42600      DO 1 II=1,N
42700      I=1-I
42800      X=Z(I,J)
42900      III=I+1
43000      IF (III.GT.N)GO TO 1212
43100      DO 2 K=III,N
43200      2 X=X-B(K,I)*Z(K,J)
43300      1212 DL(I)=X
43400      1 Z(I,J)=X/DL(I)
43500      DO 5 J=1,M
43600      H=DABS(Z(I,J))
43700      DO 28 I=2,N
43800      28 IF (DABS(Z(I,J)).GT.DABS(H))H=DABS(Z(I,J))
43900      DO 29 I=1,N
44000      29 Z(I,J)=Z(I,J)/H
44100      5 CONTINUE
44200      RETURN
44300      END

```

```

0000000000  00      00      00
0000000000  00      000  00
00      00      0000  00
00      00      0000  00
000000  00      00 000  00
0000000  00      00 000  00
00      00      00 0000
00      00      00 0000
00      00      00 0000
00      00      00 0000
00      00      00 0000

```

PROGRAMA PARA RESOLVER EL PROBLEMA DE VALORES Y VEC-  
TORES CARACTERISTICOS PARA MATRICES SIMETRICAS, PARA  
EL NUMERO DE VALORES REQUERIDOS POR EL METODO QR

```

4000 SUBROUTINE RATOR(A,B,D,DL,E,EIGEN,EL,E2,Z,W,T,N,K1,M,M2,TOL
4100 * ,DLAM,EPS,CHEPS)
4200 C
4300 C
4400 C
4500 C
4600 C
4700 C
4800 C
4900 C
5000 C
5100 C
5200 C
5300 C
5400 C
5500 C
5600 C
5700 C
5800 C
5900 C
6000 C
6100 C
6200 C
6300 C
6400 C
6500 C
6600 C
6700 C
6800 C
6900 C
7000 C
7100 C
7200 C
7300 C
7400 C
7500 C
7600 C
7700 C
7800 C
7900 C
8000 C
8100 C
8200 C
8300 C
8400 C
8500 C
8600 C
8700 C
8800 C
8900 C
9000 C
9100 C
9200 C
9300 C
9400 C
9500 C
9600 C

```

SUBROUTINE RATOR(A,B,D,DL,E,EIGEN,EL,E2,Z,W,T,N,K1,M,M2,TOL  
\* ,DLAM,EPS,CHEPS)

PROGRAMA  
RESUELVE EL PROBLEMA DE VALORES CARACTERISTICOS PARA MATRICES  
SIMETRICAS, PARA EL NUMERO DE VALORES REQUERIDOS.  
EN ESTE PROGRAMA SE REDUCE EL PROBLEMA DE LA FORMA GENERAL A  
LA FORMA ESTANDAR. LA MATRIZ DE RIGIDEZ SE TRIANGULARIZA POR  
EL METODO DE HOUSEHOLDER. LOS VALORES CARACTERISTICOS SE OBTIENEN  
POR EL METODO QR. LOS VECTORES CARACTERISTICOS SE OBTIENEN  
POR EL METODO DE ITERACION INVERSA.

REFERENCIA  
1-G.M.L. GLADWELL AND U.C. TAMBURAR  
"THE ALGEBRAIC EIGENVALUE PROBLEM  
- A SET OF FORTRAN SUBROUTINES".  
SOLID MECHANICS DIVISION UNIVERSITY OF WATERLOO.  
REPORT NO. 18 MARCH, 1972

2-G.M.L. GLADWELL  
"THE ALGEBRAIC EIGENVALUE PROBLEM  
A COMMENTARY ON SOME COMPUTERS PROGRAMS".  
SOLID MECHANICS DIVISION UNIVERSITY OF WATERLOO.  
REPORT NO. 25 MAY, 1973

LA LLAMADA A LA SUBROUTINA ES HECHA POR:  
CALL RATOR(A,B,N,K1,M,M2,TOL,CHEPS,EPS)

- - ENTRADA DE VARIABLES - -  
A(N,N) = MATRIZ DE RIGIDEZ DE LA ESTRUCTURA  
B(N,N) = MATRIZ DE MASAS DE LA ESTRUCTURA.  
N1=0 SI A ES POSITIVA DEFINIDA =1 SI NO LO ES.  
M = NO. DE VALORES CARACTERISTICOS REQUERIDOS.  
M2 = NO. DE VECT. CARACTERISTICOS REQUERIDOS.  
TOL Y DLAM DEBEN SER CEROS  
CHEPS APARECE EN FIG.

- - SALIDA - -  
Z(N,N) = MATRIZ QUE CONTIENE LOS VECT. CARACTERISTICOS.  
EIGEN(N) = VECTOR QUE CONTIENE LOS VALORES CARACTERISTICOS.

IMPLICIT REAL\*(A-H,O-Z)  
DIMENSION A(N,N),B(N,N),D(N),DL(N),E(N),EIGEN(N),EL(N)  
\* ,E2(N),Z(N,N),W(N),T(N)

CALL PTHH(A,B,DL,N,NX)  
CALL TRB(A,B,E,E2,TOL,TRM,M,N,NX)  
CALL RISE(D,E2,E,EIGEN,EPS,CHEPS,N,N)  
CALL EIGD(E,EIGEN,Z,CHEPS,TRM,M2,N,NX)  
CALL BACK(A,E,Z,M2,N,NX)

00004000  
00004100  
00004200  
00004300  
00004400  
00004500  
00004600  
00004700  
00004800  
00004900  
00005000  
00005100  
00005200  
00005300  
00005400  
00005500  
00005600  
00005700  
00005800  
00005900  
00006000  
00006100  
00006200  
00006300  
00006400  
00006500  
00006600  
00006700  
00006800  
00006900  
00007000  
00007100  
00007200  
00007300  
00007400  
00007500  
00007600  
00007700  
00007800  
00007900  
00008000  
00008100  
00008200  
00008300  
00008400  
00008500  
00008600  
00008700  
00008800  
00008900  
00009000  
00009100  
00009200  
00009300  
00009400  
00009500  
00009600

9700	CALL TRANS(B,DL,7,M2,N,NX)	00009700
9800	RETURN	00009800
9900	END	00009900
10000	SUBROUTINE REDUC(A,B,DL,N,NX)	00010000
10100		00010100
10200	REDUC EL. PROBLEMA A LA FORMA GENERAL.	00010200
10300		00010300
10400	IMPLICIT REAL*8 (A-H,O-Z)	00010400
10500	DIMENSION A(NX,NX),B(NX,NX),DL(NX)	00010500
10600	IF(N)1,2,2	00010600
10700	1 N=-N	00010700
10800	GO TO 3	00010800
10900	2 DO 8 I=1,N	00010900
11000	DO 8 J=1,N	00011000
11100	X=B(I,J)	00011100
11200	JJ=J-1	00011200
11300	K=J	00011300
11400	IF(JJ.EQ.0)GO TO 16 4	00011400
11500	DO 6 KK=1,JJ	00011500
11600	K=K-1	00011600
11700	6 X=X-B(I,K)*B(K,J)	00011700
11800	16 K=K	00011800
11900	IF(I.NE.J)GO TO 7	00011900
12000	IF(X.LE.0.0)GO TO 999	00012000
12100	Y=DSORT(X)	00012100
12200	DL(I)=DSORT(X)	00012200
12300	GO TO 8	00012300
12400	7 B(J,I)=X/Y	00012400
12500	8 CONTINUE	00012500
12600	3 DO 9 I=1,N	00012600
12700	II=I-1	00012700
12800	Y=DL(I)	00012800
12900	DO 10 JJ=1,N	00012900
13000	X=A(I,J)	00013000
13100	K=J	00013100
13200	IF(II.EQ.0)GO TO 10	00013200
13300	DO 11 KK=1,II	00013300
13400	K=K-1	00013400
13500	11 X=X-B(I,K)*A(K,I)	00013500
13600	10 A(J,I)=X/Y	00013600
13700	9 CONTINUE	00013700
13800	DO 12 J=1,N	00013800
13900	DO 12 I=J,N	00013900
14000	II=I-1	00014000
14100	JJ=J-1	00014100
14200	K=I	00014200
14300	X=A(I,J)	00014300
14400	IF(II.LT.JJ)GO TO 12	00014400
14500	DO 13 KK=1,II	00014500
14600	K=K-1	00014600
14700	13 X=X-A(K,J)*B(I,K)	00014700
14800	17 K=J	00014800
14900	IF(JJ.EQ.0)GO TO 10	00014900
15000	DO 14 KK=J,JJ	00015000
15100	K=K-1	00015100
15200	14 X=X-A(J,K)*B(I,K)	00015200
15300	18 A(I,J)=X/DL(I)	00015300
15400	12 CONTINUE	00015400
15500	GO TO 808	00015500
15600	999 PRINT 15	00015600

15700	15	FORMAT(2Y,"LA MATRIZ NO ES DEFINIDA POSITIVA")	00015700
15800	RRR	K=K	00015800
15900		RETURN	00015900
16000		END	00016000
16100		SUBROUTINE TRID(A,D,E,E2,TOL,TNORM,N,NX)	00016100
16200			00016200
16300		TRIDIAGONALIZA LA MATRIZ.	00016300
16400			00016400
16500		IMPLICIT REAL*8 (A-H,O-Z)	00016500
16600		DIMENSION A(NX,NX),D(NX),E(NX),E2(NX)	00016600
16700		DO 1 I=1,N	00016700
16800	1	D(I)=A(I,I)	00016800
16900		I=N+1	00016900
17000		DO 2 II=1,N	00017000
17100		I=II	00017100
17200		L=I-	00017200
17300		H=0	00017300
17400		IF(L.EQ.0)GO TO 14	00017400
17500		DO 3 K=1,L	00017500
17600	3	H=H+A(I,K)*A(I,L)	00017600
17700	14	IF(H.GT.TOL)GO TO 4	00017700
17800		F(I)=0	00017800
17900		E2(I)=0	00017900
18000		GO TO 5	00018000
18100	4	E2(I)=H	00018100
18200		F=A(I,I-1)	00018200
18300		IF(F.LT.0.0)GO TO 6	00018300
18400		F(I)=-DSORT(H)	00018400
18500		G=-DSORT(H)	00018500
18600		GO TO 7	00018600
18700	6	F(I)=DSORT(H)	00018700
18800		G=DSORT(H)	00018800
18900	7	H=H-F*G	00018900
19000		A(I,I-1)=F-G	00019000
19100		F=0	00019100
19200		IF(L.EQ.0)GO TO 15	00019200
19300		DO 8 J=1,I	00019300
19400		G=0	00019400
19500		DO 9 K=1,J	00019500
19600	9	G=G+A(J,K)*A(I,K)	00019600
19700		J=J+1	00019700
19800		IF(J.LT.L)GO TO 16	00019800
19900		DO 10 K=J,L,L	00019900
20000	10	G=G+A(K,J)*A(I,K)	00020000
20100	16	E(I)=G/H	00020100
20200		G=F(I)	00020200
20300	8	F=F+G*A(I,I)	00020300
20400	15	H=F/(H*H)	00020400
20500		IF(L.EQ.0)GO TO 5	00020500
20600		DO 11 J=1,L	00020600
20700		F=A(I,J)	00020700
20800		E(J)=E(I)-H*F	00020800
20900		G=F(I)	00020900
21000		DO 12 K=1,J	00021000
21100	12	A(J,K)=A(J,K)-F*E(I)-G*A(I,K)	00021100
21200	11	CONTINUE	00021200
21300	5	H=T(I)	00021300
21400		D(I)=A(I,I)	00021400
21500		A(I,I)=H	00021500
21600	2	CONTINUE	00021600

21700	TNORM=ABS(D(1))+ABS(E(1))	00021700
21800	NI=N-1	00021800
21900	DO 102 I=1,NI	00021900
22000	AL =ABS(E(I))+ABS(D(I))	00022000
22100	102 IF(AL.GT.TNORM) TNORM=AL	00022100
22200	RETURN	00022200
22300	END	00022300
22400	SUBROUTINE RATOR(D,R,EIGEN,DIAM,EPS,I,I,M,N)	00022400
22500		00022500
22600	C. CALCULA LOS VALORES CARACTERISTICOS.	00022600
22700	C	00022700
22800	IMPLICIT REAL*8 (A-H,O-Z)	00022800
22900	DIMENSION B2(N),D(N),EIGEN(N)	00022900
23000	DO 100 I=1,N	00023000
23100	100 EIGEN(I)=D(I)	00023100
23200	B2(I)=0	00023200
23300	ERR=0	00023300
23400	Q=0	00023400
23500	S=0	00023500
23600	TOT=D(I)	00023600
23700	I=N+1	00023700
23800	DO 1 II=1,N	00023800
23900	I=I-II	00023900
24000	P=0	00024000
24100	Q=DSORT(B2(I))	00024100
24200	F=D(I)-P-Q	00024200
24300	1 IF(E.LT.TOT)TOT=F	00024300
24400	IF(KJ.NE.0)GO TO 3	00024400
24500	IF(TOT)2,3,3	00024500
24600	2 TOT=0	00024600
24700	GO TO 4	00024700
24800	3 DO 5 I=1,N	00024800
24900	5 D(I)=D(I)-TOT	00024900
25000	4 DO 6 K=1,M	00025000
25100	777 TOT=TOT+S	00025100
25200	DELTA=D(N)-S	00025200
25300	I=N	00025300
25400	F=DABS(EPS*TOT)	00025400
25500	IF(DIAM.LT.F)DIAM=F	00025500
25600	IF(DELTA.LE.DIAM)GO TO 888	00025600
25700	F=D2(N)/DELTA	00025700
25800	OP=DELTA+F	00025800
25900	P=1.00	00025900
26000	I=N	00026000
26100	LL=N-1	00026100
26200	IF(K.GT.LL) GO TO 999	00026200
26300	DO 7 II=K,LL	00026300
26400	I=I-II	00026400
26500	Q=D(I)-S-F	00026500
26600	R=Q/OP	00026600
26700	P=P*R+1	00026700
26800	EP=E*R	00026800
26900	D(I+1)=OP+EP	00026900
27000	DELTA=D-EP	00027000
27100	1,LL=I+1	00027100
27200	IF(DELTA.LE.DIAM) GO TO 888	00027200
27300	IF(D2(I).LT.1.D-30)D2(I)=0.10	00027300
27400	E=B2(I)/Q	00027400
27500	OP=DELTA+E	00027500
27600	B2(I+1)=OP*EP	00027600

27700	7	CONTINUE	00027700
27800	999	CONTINUE	00027800
27900		I(K)=OP	00027900
28000		S=OP/P	00028000
28100		ST=TOT+S	00028100
28200		IF(ST.GT.TOT)GO TO 777	00028200
28300		S=0	00028300
28400		I=K	00028400
28500		DELTA=OP	00028500
28600		KK=K+1	00028600
28700		IF(KK.GT.N) GO TO 888	00028700
28800		DO 8 J=KK,N	00028800
28900		IF(D(J).LT.DELTA)I=J	00028900
29000	8	IF(D(J).LT.DELTA)DELTA=D(J)	00029000
29100	888	IF(I.I.T.N)B2(I+1)=B2(I)*E/OP	00029100
29200		I=I-1	00029200
29300		J=I	00029300
29400		IF(I.I.T.K)GO TO 11	00029400
29500		DO 9 J=K,I	00029500
29600		J=J-1	00029600
29700		D(J+1)=D(J)*S	00029700
29800	9	B2(J+1)=B2(J)	00029800
29900	11	D(K)=TOT	00029900
30000		ERR=ERR+DABS(DELTA)	00030000
30100	6	B2(K)=ERR	00030100
30200			00030200
30300	C	INTERCAMBIA R Y EIGEN	00030300
30400	C		00030400
30500		DO 101 I=1,N	00030500
30600		TEM=D(I)	00030600
30700		D(I)=EIGEN(I)	00030700
30800	101	EIGEN(I)=TEM	00030800
30900		RETURN	00030900
31000		END	00031000
31100		SUBROUTINE FIG(C,FE,W,Z,CHEPS,NORM,M1,N,NY)	00031100
31200	C		00031200
31300	C	CALCULA LOS VECTORES CARACTERISTICOS DE LA MATRIZ TRIDIAGONAL	00031300
31400	C		00031400
31500		IMPLICIT REAL*8 (A-H,O-Z)	00031500
31600		DIMENSION C(NX),W(NX),EE(NX),Z(NX,NX)	00031600
31700		DIMENSION D(200),P(200),R(200),X(200),INT(200)	00031700
31800		REAL*8 LAMDA,NORM,M(200)	00031800
31900		II=N-1	00031900
32000		DO 23 I=1,II	00032000
32100	23	EE(I)=EE(I+1)	00032100
32200		EE(N)=0.	00032200
32300		EPS=CHEPS*NORM	00032300
32400		J=M+1	00032400
32500		DO 1 JJ=1,M1	00032500
32600		J=J-1	00032600
32700		LAMDA=(LAMDIA-EPS)	00032700
32800		IF(W(J).LT.LAMDA)LAMDA=W(J)	00032800
32900		H=C(I)-LAMDA	00032900
33000		V=EE(I)	00033000
33100		IF(V.EQ.0)V=EPS	00033100
33200		KK=N-1	00033200
33300		DO 2 I=1,KK	00033300
33400		I=I+1	00033400
33500		R1=FE(I)	00033500
33600		IF(R1.EQ.0)R1=EPS	00033600

33700	R1=FF(141)	00033700
33800	IF(B1.EQ.0)R1=EPS	00033800
33900	IF(DABS(R1).LT.DABS(H))GO TO 3	00033900
34000	M(1)=U/R1	00034000
34100	IF(M(1).NE.0)GO TO 4	00034100
34200	IF(R1.LE.EPS)M(1)=1.0	00034200
34300	4 P(1)=R1	00034300
34400	Q(1)=C(I+1)-LAMBDA	00034400
34500	R(1)=R11	00034500
34600	U=V-M(I+1)*Q(1)	00034600
34700	V=-M(I+1)*R(1)	00034700
34800	INT(I+1)=+1	00034800
34900	GO TO 2	00034900
35000	3 M(I+1)=R1/U	00035000
35100	P(1)=U	00035100
35200	Q(1)=V	00035200
35300	R(1)=Q	00035300
35400	U=C(I+1)-LAMBDA-M(I+1)*V	00035400
35500	V=R11	00035500
35600	INT(I+1)=+1	00035600
35700	2 CONTINUE	00035700
35800	P(N)=H	00035800
35900	Q(N)=Q	00035900
36000	R(N)=Q	00036000
36100	X(N+1)=Q	00036100
36200	X(N+2)=Q	00036200
36300	H=Q	00036300
36400	ETA=1.0/N	00036400
36500	I=N+1	00036500
36600	DO 5 J=1,N	00036600
36700	I=I-1	00036700
36800	U=ETA-Q(I)*X(I+1)-R(I)*X(I+2)	00036800
36900	IF(P(I).EQ.0)X(I)=U/EPS	00036900
37000	IF(P(I).NE.0)X(I)=U/P(I)	00037000
37100	5 H=H-DABS(X(I))	00037100
37200	H=1/H	00037200
37300	DO 6 J=1,N	00037300
37400	7 X(I)=X(I)*H	00037400
37500	DO 7 I=2,N	00037500
37600	IF(INT(I).LE.0)GO TO 8	00037600
37700	H=X(I-1)	00037700
37800	X(I-1)=X(I)	00037800
37900	X(I)=H-M(I)*X(I-1)	00037900
38000	GO TO 7	00038000
38100	8 X(I)=X(I)-M(I)*X(I-1)	00038100
38200	7 CONTINUE	00038200
38300	H=Q	00038300
38400	I=N+1	00038400
38500	DO 9 J=1,N	00038500
38600	I=I-1	00038600
38700	U=Y(I)-Q(I)*X(I+1)-R(I)*X(I+2)	00038700
38800	IF(P(I).EQ.0)X(I)=U/EPS	00038800
38900	IF(P(I).NE.0)X(I)=U/P(I)	00038900
39000	9 H=H-DABS(X(I))	00039000
39100	H=1.0/DABS(H)	00039100
39200	DO 10 I=1,N	00039200
39300	10 Z(I,1)=Y(I)+H	00039300
39400	1 CONTINUE	00039400
39500		00039500
39600	C CAMBIA LOS INDICES DE LA SUBROUTINA DOLF	00039600

```

39700      C
39800      I=N
39900      DO 22 II=1,NN
40000      I=I-1
40100      22 EE(II)=EE(I)
40200      EE(I)=0.
40300      RETURN
40400      END
40500      SUBROUTINE BACK(A,EE,Z,M1,N,NX)
40600      C
40700      C
40800      C
40900      C
41000      IMPLICIT REAL*8 (A-H,O-Z)
41100      DIMENSION Z(NX,NX),EF(NX),A(NX,NX)
41200      DO 1 I=2,N
41300      IF (EF(I),EQ,0.0)GO TO 1
41400      L=I-1
41500      H=EE(I)*A(I,I-1)
41600      DO 2 J=1,M1
41700      S=0.0
41800      DO 3 K=1,L
41900      3 S=S+A(I,K)*Z(K,J)
42000      S=S/H
42100      DO 4 K=1,L
42200      4 Z(K,J)=Z(K,J)+S*A(I,K)
42300      2 CONTINUE
42400      1 CONTINUE
42500      RETURN
42600      END
42700      SUBROUTINE TRANS(B,DL,Z,M1,N,NX)
42800      C
42900      C
43000      C
43100      IMPLICIT REAL*8 (A-H,O-Z)
43200      DIMENSION Z(NX,NX),DL(NX),R(NX,NX)
43300      DO 1 J=1,M1
43400      I=N+1
43500      DO 1 II=1,N
43600      I=I-1
43700      X=Z(I,I)
43800      III=I+1
43900      IF (III.GT.N)GO TO 3
44000      DO 2 K=III,N
44100      2 X=X-B(K,I)*Z(K,II)
44200      3 J=1
44300      1 Z(I,I)=X/DL(I)
44400      DO 5 J=1,M1
44500      H=DARS(Z(I,J))
44600      DO 28 I=2,N
44700      28 IF (DARS(Z(I,I)),GT,DARS(H))H=DARS(Z(I,I))
44800      DO 29 I=1,N
44900      29 Z(I,I)=Z(I,I)/H
45000      5 CONTINUE
45100      RETURN
45200      END

```

```

00039700
00039800
00039900
00040000
00040100
00040200
00040300
00040400
00040500
00040600
00040700
00040800
00040900
00041000
00041100
00041200
00041300
00041400
00041500
00041600
00041700
00041800
00041900
00042000
00042100
00042200
00042300
00042400
00042500
00042600
00042700
00042800
00042900
00043000
00043100
00043200
00043300
00043400
00043500
00043600
00043700
00043800
00043900
00044000
00044100
00044200
00044300
00044400
00044500
00044600
00044700
00044800
00044900
00045000
00045100
00045200

```

```

0000000000 00 00 00 00
0000000000 00 00 00 00
00 00 000 00
00 00 0000 00
000000 00 000 00
00000 00 00 000 00

```