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

P R E S E N T A

HECTOR URQUIJO PARRA

MEXICO, D. F. 1982

FACULTAD DE INGENIERIA
DIRECCION
60-1-378

UNIVERSIDAD NACIONAL AUTONOMA

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.

A t e n t a m e n t e
"POR MI RAZA HABLARA EL ESPIRITU"
Cd. Universitaria, 25 de agosto de 1982
EL DIRECTOR

ING. JAVIER JIMENEZ ESPRIU

JJE/OBLH/ser

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 Burrougs-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 a_{ii} 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).

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

$$U_{ij} = \frac{a_{ij} - \sum_{r=1}^{i-1} l_{ir} U_{rj}}{l_{ii}} \qquad i < j \qquad \text{Matriz triangular superior.}$$

$$U_{ii} = 1$$

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:

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

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

En donde M se puede obtener usando las siguientes relaciones

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

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

$$m_{ij} = 0 \qquad i < j$$ 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.

$$n_{ii} = 1$$ para los elementos de la diagonal.

$$n_{ij} = -\sum_{r=j}^{i-1} n_{ir} U_{rj} \qquad i < j$$ elementos arriba de la diagonal.

$$n_{ij} = 0 \qquad i > j$$ elementos bajo la diagonal.

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, $l_{ii} = 0$, 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 Λ 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 Λ están dados por.

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

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

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

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

$$u_{ii} = 1 / \lambda_{ii} \qquad i = j$$

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

$$u_{ij} = 0 \qquad i < j$$

La inversa de la matriz Λ es una matriz triangular inferior. Para determinar los elementos de Λ^{-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_{j1} $j=2,\ldots.N$. 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.

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

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

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

Reducción del vector de cargas

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

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.

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

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 <u>Reducción de Crout.</u> 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

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

$$b_{i}^{(i-1)} = b_{i} - \sum_{k=1}^{i-1} \frac{a_{ki}^{(k-1)}}{a_{kk}^{(k-1)}}\, b_{k}^{(k-1)} \qquad \begin{array}{l} i = 2, \ldots\ldots, j \\ i = 2, \ldots\ldots, N \end{array}$$

Procedimiento de sustitución regresiva

$$X_{N} = \frac{b_{N}^{(N-1)}}{a_{NN}^{(N-1)}} \; ; \quad X_{i} = \frac{b_{i}^{(i-1)} - \sum_{j=i+1}^{N} a_{ij}^{(i-1)} X_{j}}{a_{ii}^{(i-1)}} \; , \quad i = N-1, \ldots\ldots, 1$$

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

$$a_{ij}^{(i-1)} = a_{ij} - \sum_{k=k_0}^{i-1} \bar{a}_{ki}^{(k-1)}\, a_{kj}^{(k-1)}$$

en donde

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

$$j = 2, \ldots\ldots, N$$

$$i = L_j + 1, \ldots\ldots, j-1$$
$$K_o = \max(L_i, L_j)$$

y

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

Reducción del vector de cargas

$$b_i^{(i-1)} = b_i - \sum_{K=K_o}^{i-1} \bar{a}_{ki}^{(K-1)} b_k^{(K-1)} \qquad i = L_{N+1}+1, \ldots\ldots, N \qquad (2.2.3.2)$$

$$b_i^{(i-1)} = b_i^{(i-1)} / a_{ii}^{(i-1)} \qquad K_o = \max(L_i, L_{N+1}) \qquad (2.2.3.3)$$
$$i = L_{N+1}, \ldots\ldots, N$$

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.

$$X_k = b_k^{(K-1)} \qquad K = 1, \ldots\ldots, N$$
$$X_k = X_k - \bar{a}_{ki}^{(K-1)} X_i \qquad i = N, N-1, \ldots\ldots, 2$$
$$K = L_i, L_{i+1}, \ldots\ldots, i-1$$

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 proposito 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

$$K\phi = \lambda\phi \tag{3.1.1}$$

Dondo 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 (λ_i, ϕ_i) , donde los valores característicos son ordenados de acuerdo con sus magnitudes:

$$0 \leq \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_{n-1} < \lambda_n$$

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

$$K\phi = \phi\Lambda \tag{3.1.2}$$

Donde ϕ es una matriz de orden $n \times p$ en donde las columnas son iguales a los P vectores característicos y Λ es una matriz diagonal en donde los elementos de la diagonal corresponden a los valores característicos. En cualquier caso $\phi = [\phi_1 \ldots\ldots \phi_p]$ y $\Lambda = \mathrm{diag}(\lambda_i), i=1,..,P$.

Se recuerda que si K es positiva definida, $\lambda_i > 0, \; i=1,\ldots,n$ y si K es semidefinida positiva $\lambda_i \geq 0, \; i=1,\ldots,n$, 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.

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

Donde K y M son respectivamente, la matriz de rigidez y la de masas. Los valores característicos λ_i y vectores característicos ϕ_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 ϕ son los vectores característicos y Λ 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 forma 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

$$\phi K \phi = \Lambda \qquad (3.2.1.1)$$

$$\phi M \phi = I \qquad (3.2.1.2)$$

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 R_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 R_k es.

$$
\begin{array}{cc}
 & \begin{array}{cccc} \quad & \text{i-ésima} & \text{j-ésima columna} & \quad \end{array} \\
R_k = & \left[\begin{array}{cccc} 1 & & & \\ & \cos\theta & -\operatorname{sen}\theta & \\ & & \ddots & \\ & \operatorname{sen}\theta & \cos\theta & \\ & & & 1 \end{array}\right] \begin{array}{l} \\ \text{i-ésimo} \\ \\ \text{j-ésimo renglón} \\ \\ \end{array}
\end{array}
$$

Donde θ 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.

$$\tan 2\theta = \frac{2K_{ij}^{(k)}}{K_{ii}^{(k)} \quad K_{jj}^{(k)}} \qquad K_{ii}^{(k)} \neq K_{jj}^{(k)}$$

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.

3.2.2 <u>Método generalizado de Jacobi.</u> En la sección anterior se discutió la solución del problema estándar de valores característicos $K\phi=\lambda\phi$ usando el método convencional de rotación de matrices destinadas a reducir K a una forma diagonal. Para resolver el problema generalizado $K\phi=\lambda M\phi, M\neq I$, usando el método estándar de Jacobi es necesario transformar el problema a la forma estándar. Sin embargo, esta transformación se puede evitar usando el método generalizado de Jacobi, que opera directamente en K y M.

El algoritmo es una extensión natural de la solución estándar de Jacobi, el método generalizado se reduce al esquema presentado para el problema $K\phi=\lambda\phi$ cuando M es una matriz identidad.

Refiriéndose a la discusión en la sección previa, en el problema generalizado de iteración de Jacobi se usa la siguiente matriz P_k:

$$
\begin{array}{cc}
 & \begin{array}{ccccc} & \text{i-ésima} & \text{j-ésima columna.} & & \end{array} \\
P_k = & \left[\begin{array}{ccccc}
1 & & & & \\
 & \ddots & & & \\
 & & 1 & \alpha & \\
 & & \gamma & 1 & \\
 & & & \ddots & \\
 & & & & 1
\end{array}\right]
\begin{array}{l} \\ \\ \text{— i-ésimo} \\ \text{— j-ésimo renglón} \\ \\ \\ \end{array}
\end{array}
$$

Donde las constantes α y γ son escogidas de manera que reduzcan a cero simultáneamente los elementos (i,j) en K_k y M_k. Sin embargo, los valores de α y γ son una función de los elementos $K_{ij}^{(k)}$, $K_{ii}^{(k)}$, $K_{jj}^{(k)}$, $m_{ij}^{(k)}$, $m_{ii}^{(k)}$ $m_{jj}^{(k)}$, en donde el índice superior indica la k-ésima iteración considerada.

Efectuando la multiplicación $P_k^T K_k P_k$ y $P_k^T M_k P_k$ y usando la condición que $K_{ij}^{(k+1)}$ y $m_{ij}^{(k+1)}$ serán cero, se obtienen las dos siguientes ecuaciones para α y γ :

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

y

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

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 $=-K_{ij}^{(k)}/K_{jj}^{(k)}$.

En general, se resuelve para α y γ de las ecs(3.2.2.1) y (3.2.2.2) se define.

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

y

$$\gamma=-\bar{K}_{ii}^{(k)}/X; \qquad \alpha=\bar{K}_{jj}^{(k)}/X$$

El valor de X para obtener α y γ 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 α y γ 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 $R_k \neq 0$, 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 λ 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 λ_1 y un incremento $\Delta\lambda_1$ Se calcula Δ_1 y Δ_2 donde.

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

y

$$\Delta_1 = |K-\lambda_1 M|$$
$$\Delta_2 = |K-(\lambda_1+\Delta\lambda)M|$$

Si estos determinantes tienen signos opuestos la raíz se encontrará entre λ_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 Δ_3.

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

Se revisa el signo de los determinantes Δ_1, Δ_3 y Δ_3, Δ_2 . Si la primera de las dos parejas tiene diferente signo, entonces la raíz se encontrará entre λ_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 Δ_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 Δ_1 y Δ_2 tienen el mismo signo, entonces la raíz no se encuentra entre λ_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 Δ_3'.

$$\Delta_3' = \left| K - (\lambda_1 + 2\Delta\lambda_1)M \right|$$

Si los determinantes Δ_2 y Δ_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 <u>Cálculo de los valores característicos de una matriz tridiagonal simétrica por el método de bisección.</u> 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. [illegible]).

$$A = \begin{bmatrix} d_1 & b_2 & 0 & & 0 \\ b_2 & d_2 & b_3 & & 0 \\ 0 & b_3 & d_3 & & 0 \\ & & & & \\ & & & d_{n-1} & b_n \\ 0 & 0 & 0 & b_n & d_n \end{bmatrix}$$

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

$$P_1(\lambda)=d_1-\lambda \ , \quad P_2(\lambda)=(d_2-\lambda)(d_1-\lambda)-b_2^2 \ , \ \text{etc.}$$

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

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

Ahora, si se define $a(\lambda)$ como el número de cambios de signo en la secuencia $P_0(\lambda), P_1(\lambda), \ldots\ldots P_n(\lambda)$, 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 $a(\lambda)$ por este método es muy estable para matrices de orden modesto. La dificultad es evitada remplazando la sucesión $P_i(\lambda)$ por una sucesión $q_i(\lambda)$ definida por.

$$q(\lambda)=P(\lambda)/P(\lambda) \qquad (i=1(1)n)$$

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

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

$$q_i(\lambda)=d_i-\lambda-b_i^2/q_{i-1}(\lambda)$$

A primera vista esta relación parece muy peligrosa, porque $q_{i-1}(\lambda)$ puede ser cero para alguna i , en tal caso es necesario remplazar 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 remplazado por una división; adicionalmente, solo se tiene que detectar el signo de cada $q(\lambda)$ 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.

$$X_{max}=\max\{d_i+|b_i|+|b_{i+1}|\}$$
$$X_{min}=\min\{d_i-|b_i|-|b_{i+1}|\}$$

los valores característicos se encuentran en orden descendiente,

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 $K=1,2,\ldots$:

$$K\bar{X}_{K+1}=MX_K \qquad (3.4.1)$$

y

$$X_{K+1}=\frac{\bar{X}_{K+1}}{(\bar{X}^T_{K+1}M\bar{X}_{K+1})^{1/2}} \qquad (3.4.2)$$

con la condición que X_1 no sea ortogonal a M para ϕ_1 significa que $X_1^TM\phi_1\neq 0$, se tiene que

$$X_{K+1}\longrightarrow\phi_1 \quad \text{cuando} \quad K\longrightarrow\infty$$

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.

$$X^T_{K+1}MX_{K+1}=1 \qquad (3.4.3)$$

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 ϕ_1 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 $y_1 = Mx_1$ se evalúa para $K=1,2,\ldots\ldots,$

$$K\bar{x}_{K+1} = y_K \tag{3.4.4}$$

$$\bar{y}_{K+1} = M\bar{x}_{K+1} \tag{3.4.5}$$

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

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

donde, con la condición que $y_1^T\phi_1 \neq 0$,

$$y_{K+1} \longrightarrow M\phi_1 \qquad (\bar{x}_{K+1}) \longrightarrow \lambda_1 \text{ cuando } K \longrightarrow \infty$$

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 λ_1 dado por el cociente de Rayleigh $\rho(\bar{x}_{K+1})$. Esto es una aproximación a λ_1 la cual es convenientemente usada para determinar la convergencia en la iteración.

3.4.1 <u>Iteración Frontal.</u> 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 $K=1,2,\ldots$

$$M\bar{x}_{K+1} = Kx_K \tag{3.4.8}$$

y

$$x_{K+1} = \frac{\bar{x}_{K+1}}{(\bar{x}^T_{K+1} M \bar{x}_{K+1})^{1/2}} \tag{3.4.9}$$

con la condición que x_1 no es ortogonal a M para ϕ_n, se tiene que

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

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 $K=1,2,\ldots\ldots\ldots,$

$$M\bar{x}_{K+1} = y_K \tag{3.4.10}$$

$$\bar{y}_{K+1} = K\bar{x}_{K+1} \tag{3.4.11}$$

$$\rho(\bar{x}_{K+1}) = \frac{\bar{x}^T_{K+1}\,\bar{y}_{K+1}}{\bar{x}^T_{K+1}\,y_K} \tag{3.4.12}$$

$$y_{K+1} = \frac{\bar{y}_{K+1}}{(\bar{x}^T_{K+1}\,y_K)^{1/2}} \tag{3.4.13}$$

con la condición que $\phi_n^T y_1 \neq 0$

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

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 metodo 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

$$A_m = Q_m R_m \qquad A_{m+1} = R_m Q_m \qquad m = 1, 2, \ldots\ldots, \tag{3.5.1}$$

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.

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

A_{m+1} es ortogonal y similar a A_m , y así por inducción $A_1 = A$. De la ec(3.5.2)

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

introduciendo las matrices P_m y U_m en

$$P_m = Q_1 \ldots Q_m \qquad U_m = R_m \ldots R_1$$

y de la ec(3.5.3)

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

La matriz P_m es ortogonal y U_m es una matriz triangular superior: De la ec(3.5.2) con m remplazando a $m+1$

$$Q_1 \ldots Q_{m-1} A_m = A_1 Q_1 \ldots Q_{m-1}$$

usando esto,

$$P_m U_m = A_1 Q_1 \ldots Q_{m-1} R_{m-1} \ldots R_1 = A_1 P_{m-1} U_{m-1}$$

Después $P_1 U_1 = Q_1 R_1 = A_1$, por inducción en la última ecuación

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

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 $A_1=Q_1R_1$ y $A_2=R_1Q_1$, 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 $A_1=Q_1R_1$ con el siguiente valor para Q_1:

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

en donde cada H_K es una matriz de Householder.

$$H_K=I-2W^{(K)}W^{(K)T} \qquad 1 \leq K \leq n-1$$

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

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

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 <u>Método QR con cambios.</u> 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

$$A_m - C_m I = Q_m R_m$$
$$A_{m+1} = C_m I + R_m Q_m \qquad m = 1, 2, \ldots\ldots$$

Las matrices A_m son similares a A_1 , después

$$R_m = Q_m^T (A_m - C_m I)$$
$$A_{m+1} = C_m I + Q_m^T (A_m - CI) Q_m$$
$$= C_m I + Q_m^T A_m Q_m - C_m I$$
$$A_{m+1} = Q_m^T A_m Q_m \qquad m \geqq 1$$

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

$$A_m = \begin{bmatrix} \alpha_1^{(m)} & \beta_1^{(m)} & 0 & \cdots & 0 \\ \beta_1^{(m)} & \alpha_2^{(m)} & \beta_2^{(m)} & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \beta_{n-1}^{(m)} \\ 0 & \cdots & \cdots & \beta_{n-1}^{(m)} & \alpha_n^{(m)} \end{bmatrix}$$

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.

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

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 loa bloques tienen orden 1 o 2. En uno u otro caso la elección de $\{C_m\}$ garantiza que

$$\beta_{n-1}^{(m)} \beta_{n-2}^{(m)} \longrightarrow 0 \quad \text{cuando} \quad n \longrightarrow \infty \qquad (3.5.1.2)$$

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

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

De la ec(3.5.1.2) de $\{\beta_{n-1}^{(m)}\}$ y $\{\beta_{n-2}^{(m)}\}$, se tiene que cualquiera $\beta_{n-1}^{(m)} \longrightarrow 0$ ó $\beta_{n-2}^{(m)} \longrightarrow 0$ cuando $m \longrightarrow \infty$. 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 rapido a cero de manera que sea más rápida para $\beta_{n-1}^{(m)}$ $\beta_{n-2}^{(m)}$ 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.

$$AX=\lambda BX \tag{3.6.1}$$

En donde A y B son matrices reales y simétricas, B es definida positiva, el vector columna X representa un vector característico y λ 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.

$$R^T A R = T \tag{3.6.2}$$

y

$$R^T B R = I \tag{3.6.3}$$

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.

$$Ty=\lambda y \tag{3.6.4}$$

donde los vectores X y y están relacionados directamente por la matriz de transformación.

$$X = Ry \tag{3.6.5}$$

Premultiplicando las ecs(3.6.2) y (3.6.3) por $(R^T)^{-1}$ y combinado ambas se tiene:

$$AR = BRT \tag{3.6.6}$$

Ahora, si los vectores de R se definen de la forma:

$$R = \left[r_1,\ r_2 \ldots\ldots\ldots r_n \right] \tag{3.6.7}$$

y los elementos de la matriz tridiagonal T como

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{bmatrix} \tag{3.6.8}$$

Sustituyendo las ecs(3.6.7) y (3.6.8) en la ec(3.6.6) se tiene una expresión, de la cual A veces el iésimo vector de R se cambia a

$$Ar_i = \beta_{i-1} Br_{i-1} + \alpha_i Br_i + \beta_i B r_{i+1} \tag{3.6.9}$$

Ahora, se obtiene una expresión para el elemento de la diagonal multiplicando la ec(3.6.9) por r_i^T . El resultado es:

$$\alpha_i = r_i^T A\, r_i \tag{3.6.10}$$

porque los vectores de R son ortogonales con respecto a B . En esta etapa α_i puede calcularse si se conoce r_i .

El proceso se inicia tomando el primer vector r_1 como:

$$r_1 = \left\{ 1/\sqrt{B_{1,1}}\ ,0, \ldots\ldots\ldots ,0 \right\} \tag{3.6.11}$$

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

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

donde

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

Si β_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.

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

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

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

si se define z_i como

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

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:

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

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

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

Así, β_i se calcula como:

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

Dividiendo la ec(3.6.16) entre β_i se obtiene una expresión de donde r_{i+1} puede ser obtenida por un proceso de sustitución regresiva, dado por.

$$Ur_{i+1} = \frac{1}{\beta_i} z_i \qquad (3.6.20)$$

El proceso se repito 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:

$$r_{i+1} = \left\{0, 0, \ldots\ldots, 1/\sqrt{\beta_{i,i}}, \ldots\ldots, 0\right\}$$

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 negativos). 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:

$$\begin{aligned} K_2 &= P_1^T K_1 P_1 \\ K_3 &= P_2 K_2 P_2 \\ &\vdots \\ K_{k+1} &= P_k^T K_k P_k \end{aligned} \qquad (3.7.1.1)$$

usando $K_1 = K$, se calcula

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

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

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

con
$$\Theta = \frac{2}{W_K^T W_K} \tag{3.7.1.4}$$

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:

$$P_1 = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \bar{P} \end{array}\right]; \qquad W_1 = \left[\begin{array}{c} 0 \\ \hline \bar{W}_1 \end{array}\right]$$

$$K_1 = \left[\begin{array}{c|c} k_{11} & k_1^T \\ \hline k_1 & K_{11} \end{array}\right] \tag{3.7.1.5}$$

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

$$K_2 = \left[\begin{array}{c|c} k_{11} & k_1^T \bar{P}_1 \\ \hline \bar{P}_1^T k_1 & \bar{P}_1^T K_{11} \bar{P}_1 \end{array}\right] \tag{3.7.1.6}$$

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.

$$K_2 = \left[\begin{array}{c|cccc} k_{11} & x & 0 & \cdots & 0 \\ \hline x & & & & \\ 0 & & & & \\ \vdots & & \bar{K}_2 & & \\ 0 & & & & \end{array}\right] \tag{3.7.1.6a}$$

donde x indica un valor diferente de cero, y

$$\bar{K}_2 = \bar{P}_1^T K_{11} \bar{P}_1 \tag{3.7.1.7}$$

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 , $\overline{W}_1$ se determina de la condición.

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

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

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

donde κ_{21} es el elemento (2,1) de K_1.
Con $\overline{W}_1$ 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 $\overline{K}_2$ 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

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

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

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

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

$$P^T_{n,n-1} \cdots\cdots\cdots P^T_{3,1} P^T_{2,1} K = R \qquad (3.7.2.3)$$

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

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

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 $K_1 = K$, se forma.

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

y entonces

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

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

$$K_{k+1} \longrightarrow \Lambda \qquad Q_1 \cdots\cdots Q_{k-1} Q_k \longrightarrow \Phi \quad \text{cuando} \quad K \longrightarrow \infty$$

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 <u>Cálculo de los vectores característicos.</u> 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 i-ésimo vector característico de T_1 para ψ , se tiene, usando las matrices de transformación P_K en la ec(3.7.1.2)

$$\phi_i = P_1 P_2 \cdots\cdots P_{n-2} \psi_i \qquad (3.7.3.1)$$

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 se necesita 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 [illegible] la reparación 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 menores 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 susecivos 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

$$\left| U_{n,j} - U_{n-1,j} \right| < \varepsilon \quad \text{para toda } j$$

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

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

Sin embargo, este método puede causar problemas cuando uno de los elementos es cero o cercano a cero. En tal caso la diferencia entre susecivos 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 arri be.

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 procesado 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 procesado, 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 eliminación 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 análizar 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 leia los datos y escribia 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 Ingeniería de la UNAM 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
<u>Calculation of the Eigenvalues of a Symmetric Tridiagonal Matrix by the Method of Bisection</u>,
Num. Math. 9, PP. 386-393, 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
30.26487	30.26487	30.26487	30.26446	30.26487
624.99581	624.99581	624.99581	624.99074	624.99580
3464.35099	3464.35099	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($\frac{rad}{seg}$) del problema de la fig 10

Jacobi	Lanczos	Bisección	RATQR
25.51773	25.51773	25.53550	26.63568
96.53106	96.53106	96.56470	96.56532
302.63041	302.63041	302.83170	302.85049
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	1829.72978
2184.34296	2184.34296	2186.10097	2186.10298
2908.92073	2908.92073	2908.87998	2908.92082

Tabla 2. Frecuencias($\frac{rad}{seg}$) 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 $\left(\frac{rad}{seg}\right)$ del problema de la fig 12b

Ejemplo	1	2
Método	Matriz de 14×14	27×27
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 procesado 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° de V V obtenidos 3	8	3	10	5	30
Jacobi Ns=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 procesado de computadora (Bourroughs 6800) entre los diferentes métodos de solución del problema de valores y vectores característicos figs 10 y 12

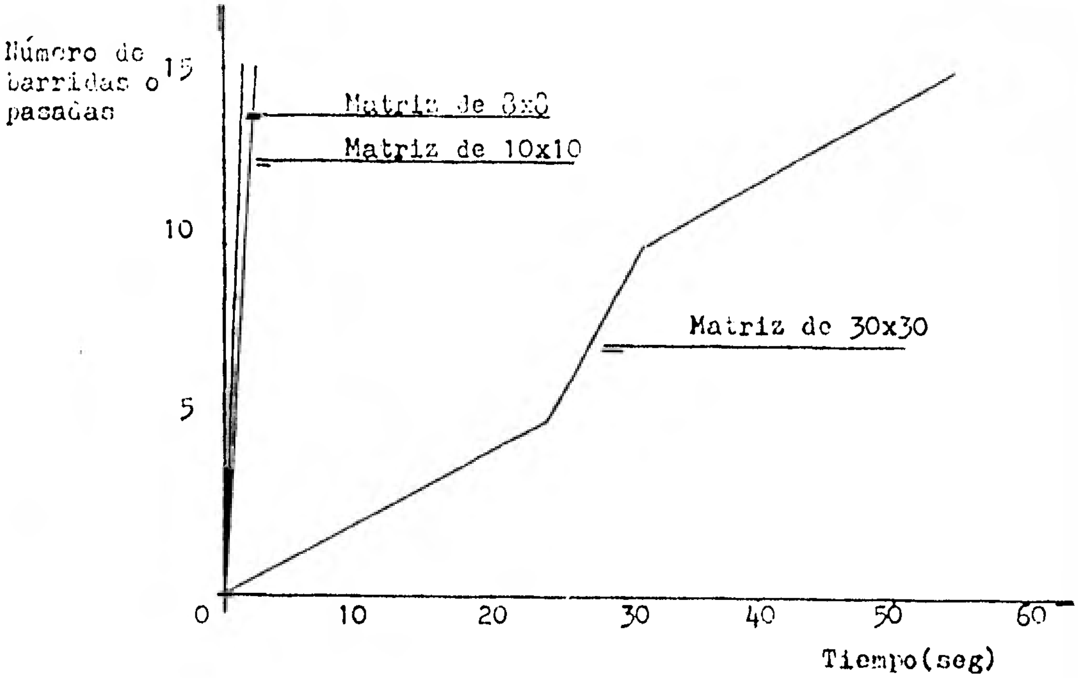


Fig. 1 Se muestra cómo varía el tiempo con el número de pasadas(barridas) a la matriz de rigidez, por el método de Jacobi, ejemplos 3,4,5.

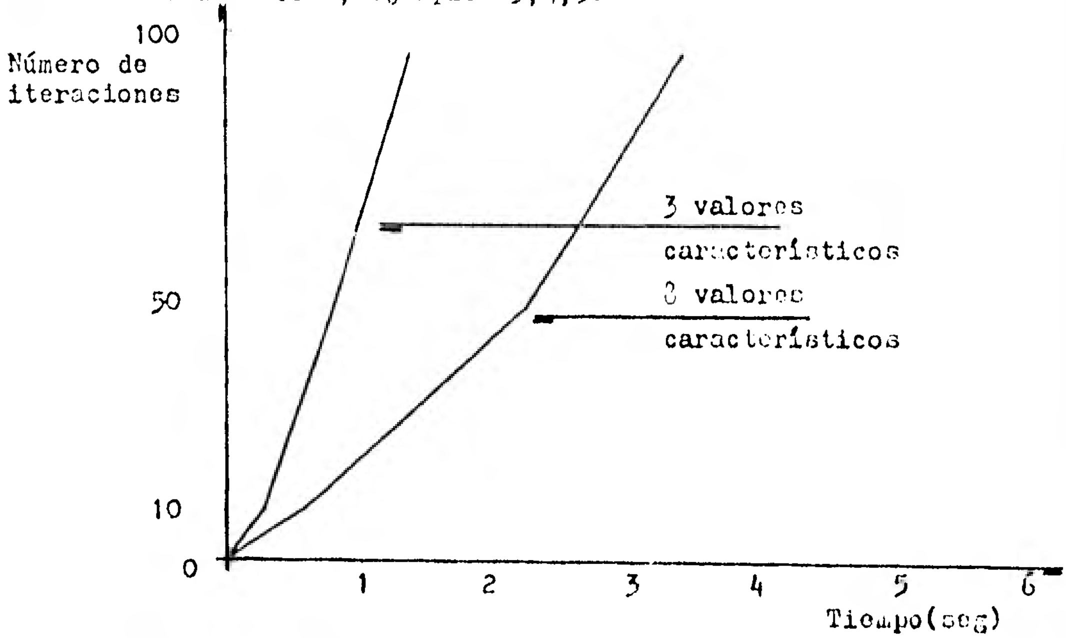
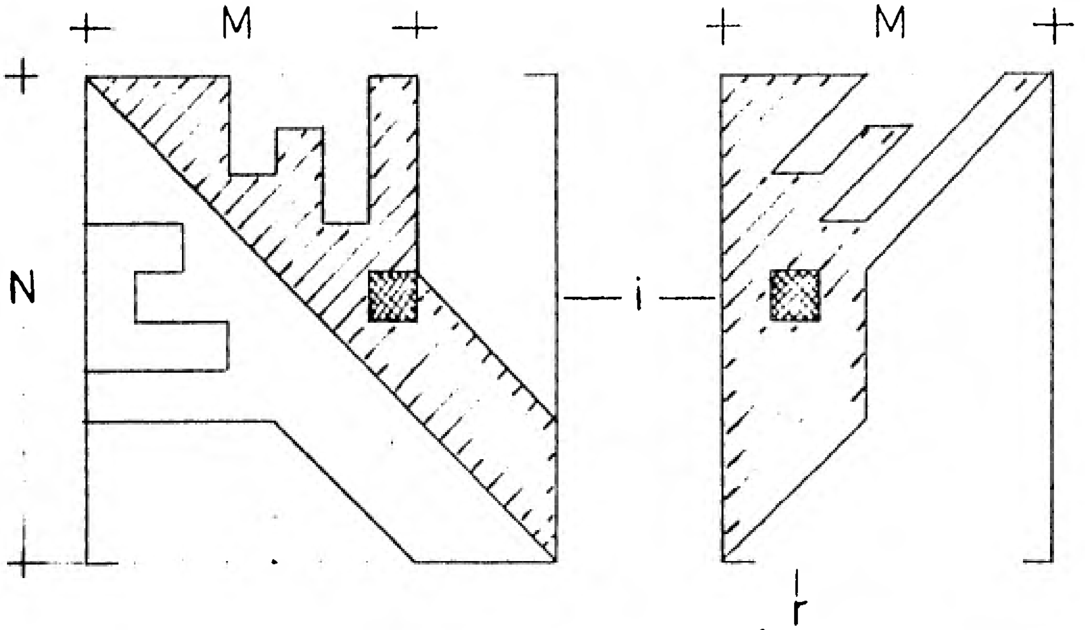


Fig. 2 Se muestra cómo aumenta el tiempo con el número de iteraciones para el programa de Iteración Inversa, para el ejemplo 3

Esquemas de Almacenaje.

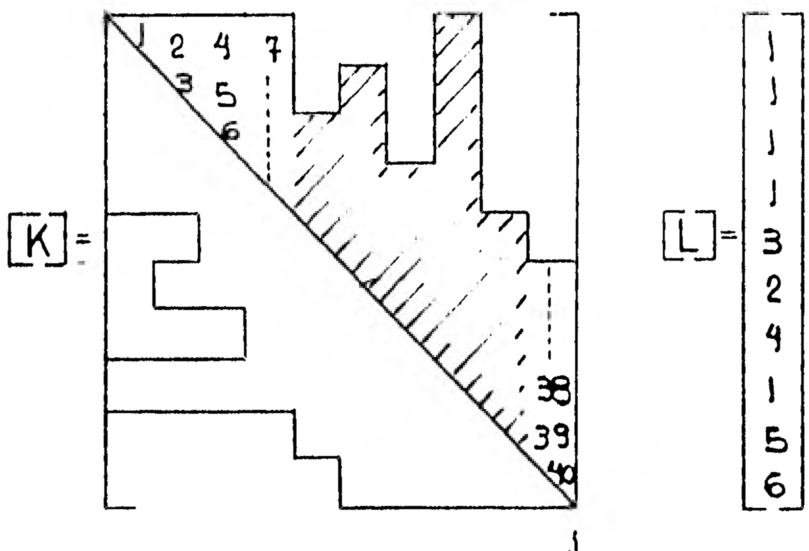


Renglón=i, Columna=j

Renglón=i, Columna=r=j-i+1

Esquema Número 1

Fig. 3 Esquema de almacenaje normal y en banda.



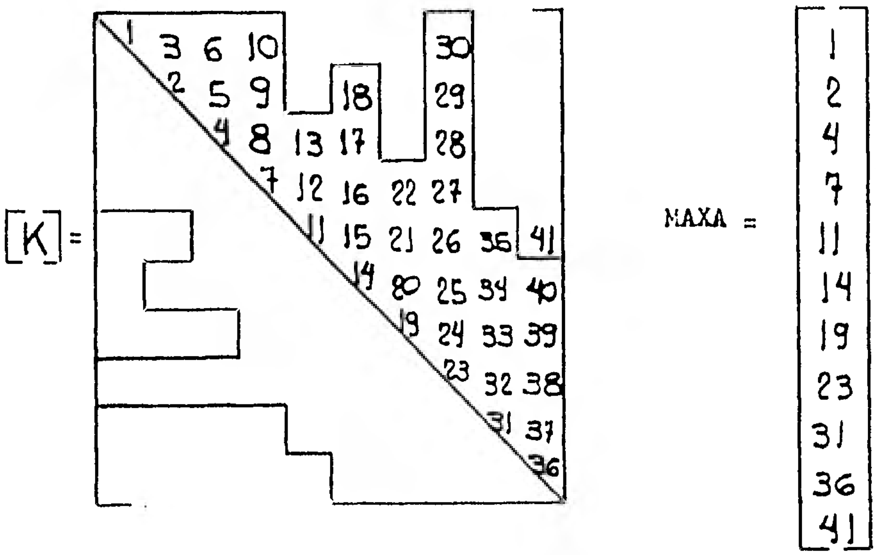
Vector que contiene el número del renglon del primer elemento diferente de cero en cada columna en la matriz de coeficientes. (Para el ejemplo)

Esquema Número 2

$$\text{Localización} = i - j + \sum_{k=1}^{j} (k - L_k + 1)$$

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

Esquemas de Almacenaje.



Vector que contiene la dirección de los elementos de la diagonal

(Para el ejemplo)

Esquema Número 3

El vector de cargas en todos los programas se almacena en forma normal.

Fig. 5 Esquema de almacenaje en columna activa.

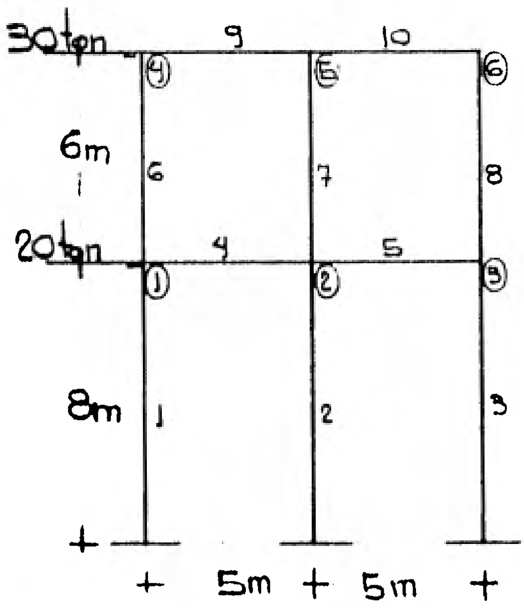


Fig. 6 Marco a ser analizado.

Ejemplo 1

Todas las barras tienen sección de 50x50 cm su área y su inercia son:

Area = 0.25 m^2

Inercia = 5.21×10^{-3} m^4

[K] =

1.967	0.4		0.333			-0.073	0.167	0.24	-0.24				
0.4	2.767	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	-0.096	
			-0.24		0.24				-8.0		-0.096	8.196	-0.096
				-0.24	0.24					-8.0		-0.096	8.096

[ton/m]

Matriz de Rigidez

[F] =

0.0
0.0
0.0
0.0
0.0
0.0
20.0
30.0
0.0
0.0
0.0
0.0
0.0
0.0

Vector de cargas [ton]

[D] =

-70.686
-37.195
-70.686
-29.955
-17.619
-29.955
954.075
1384.812
6.0
0.0
-6.0
7.344
0.0
-7.344

Vector de desplazamientos [m]

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

Ejemplo 2

Todas las barras son de sección de 50x60 cm.

Area = 0.30 m^2

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

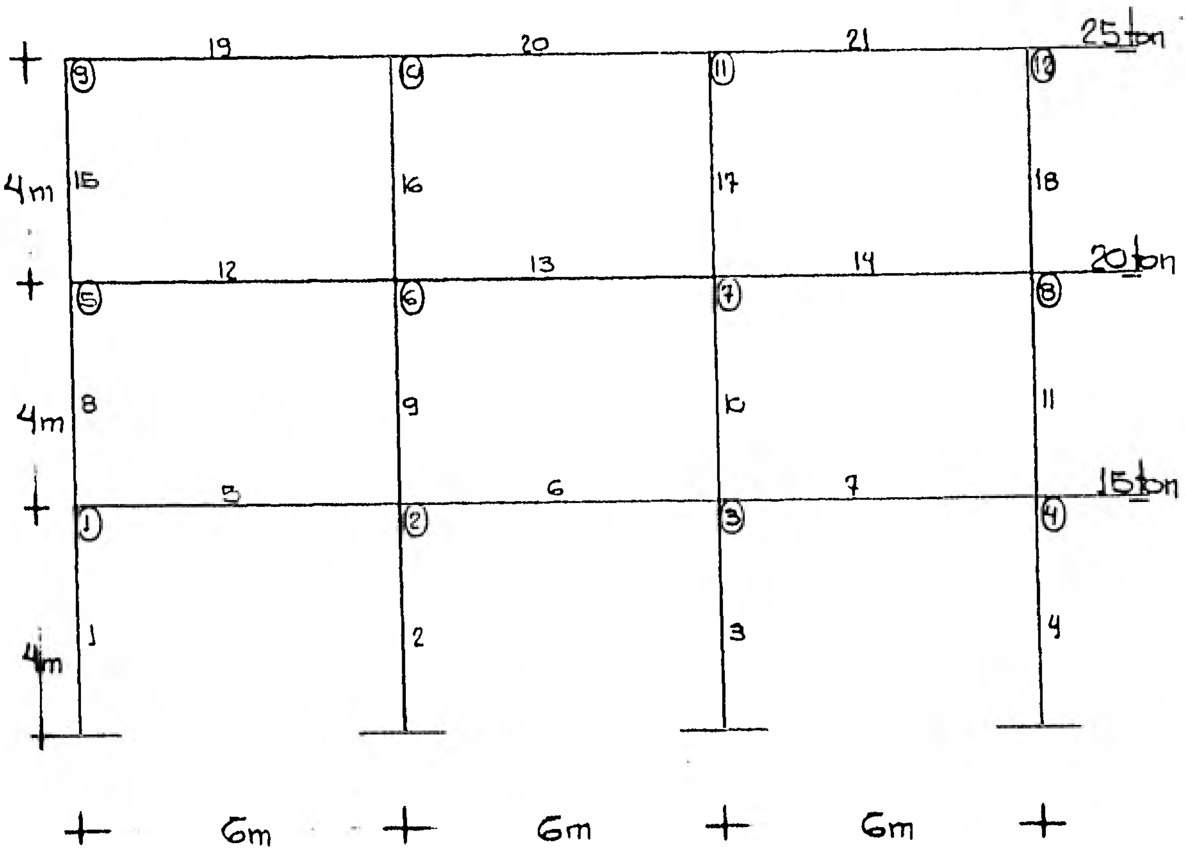


Fig. 8 Marco a ser analizado, se consideran tres grados de libertad por nudo, las trabes no se deforman bajo carga axial.

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	0.375	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	16.772	-0.056			-8.33		
	-0.167	0.073	0.24													-0.056	16.81	-0.096			-8.33	
		-0.24	-0.24														-0.096	16.756				-8.33
				0.167	0.167										-8.33				16.716	-0.056		
				-0.167		0.167										-8.33			-0.056	16.772		
					-0.167	0.073	0.24										-8.33			-0.056	16.81	-0.096
						-0.24	-0.24											-8.33			-0.096	16.756
								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 trabes no se deforma bajo carga axial.

	0.375		0.167	-0.167												0.0		-30.851	
	0.375		0.167		-0.167											0.0		-22.241	
	0.375			0.167	0.073	-0.24										0.0		-21.308	
	0.375				0.24	-0.24										0.0		-29.308	
0.375		0.375					0.167	-0.167								0.0		-18.939	
0.375		0.375					0.167		-0.167							0.0		-14.966	
0.375		0.375						0.167	0.073	-0.24						0.0		-14.435	
-0.375		0.375							0.24	-0.24						0.0		-18.333	
	-0.375	0.375									0.167	-0.167				0.0		-8.738	
	-0.375	0.375									0.167		-0.167			0.0		-5.728	
	-0.375	0.375										0.167	0.073	-0.24		0.0		-5.628	
	-0.375	0.375											0.24	-0.24		0.0		-8.496	
1.50	-0.75															25.0		131.742	
-0.75	1.50	-0.75													[Ton/m]	20.0	[Ton]	263.488	[m]
	-0.75	0.75														15.0		331.122	
			16.716	-0.056			-8.33									0.0		1.970	
			-0.056	16.772	-0.056			-8.33								0.0		-0.256	
				-0.056	16.81	-0.096			-8.33							0.0		0.924	
					-0.096	16.756				-8.33						0.0		-2.637	
			-8.33				16.716	-0.056			-8.33					0.0		2.892	
				-8.33			-0.056	16.772				-8.33				0.0		-0.340	
					-8.33			-0.056	16.81	-0.096			-8.33			0.0		1.312	
						-8.33			-0.096	16.756						0.0		-3.864	
							-8.33				8.386	-0.056				0.0		3.158	
								-8.33			-0.056	8.442	-0.056			0.0		-0.366	
									-8.33			-0.056	8.482	-0.096		0.0		1.415	
										-8.33			-0.096	8.426		0.0		-4.206	

esultante de considerar tres grados de libertad por nudo.
g. 8 las trabes no se deforma bajo carga axial.

Vector de cargas.

Vector de desplazamientos.

Ejemplo 3

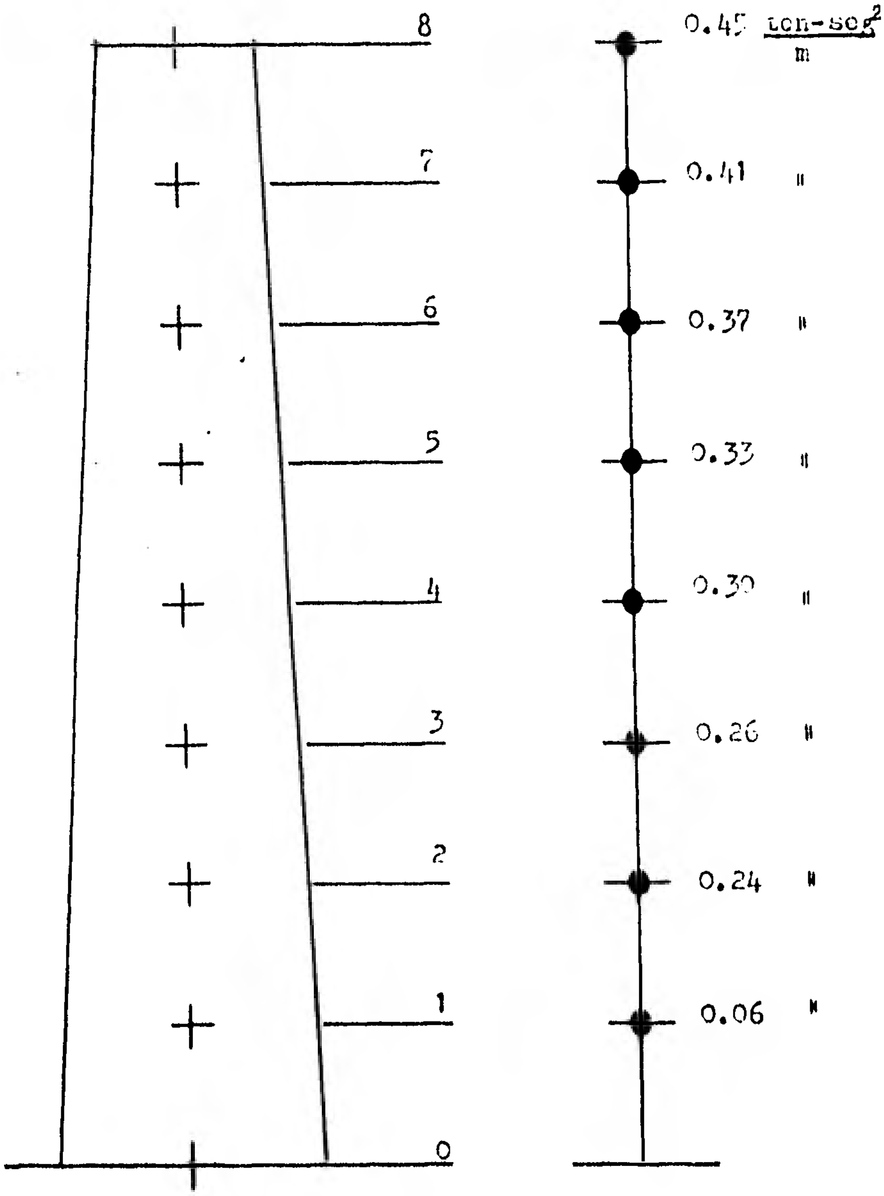


Fig. 10 Chimenea a ser analizada y concentración final de las masas.

[K] =

12791.0	-6695.5	1307.8	119.41	-12.38	-7.77	20.23	-8.11
-6695.5	9943.3	-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
119.41	1122.0	-50363	7159.4	-4249.9	850.29	37.60	0.94
-12.38	83.51	993.4	-4249.9	59500	-3519.9	724.3	20.60
-7.77	-14.24	70.85	850.29	-3519.9	48568	-2865.1	615.82
20.23	15.66	-21.02	37.60	724.30	-2865.1	3409.7	-1303.8
-8.11	-4.43	8.46	0.94	20.60	615.82	-1303.8	664.32

[Ton/m]

Matriz de Rigidez

[M] =

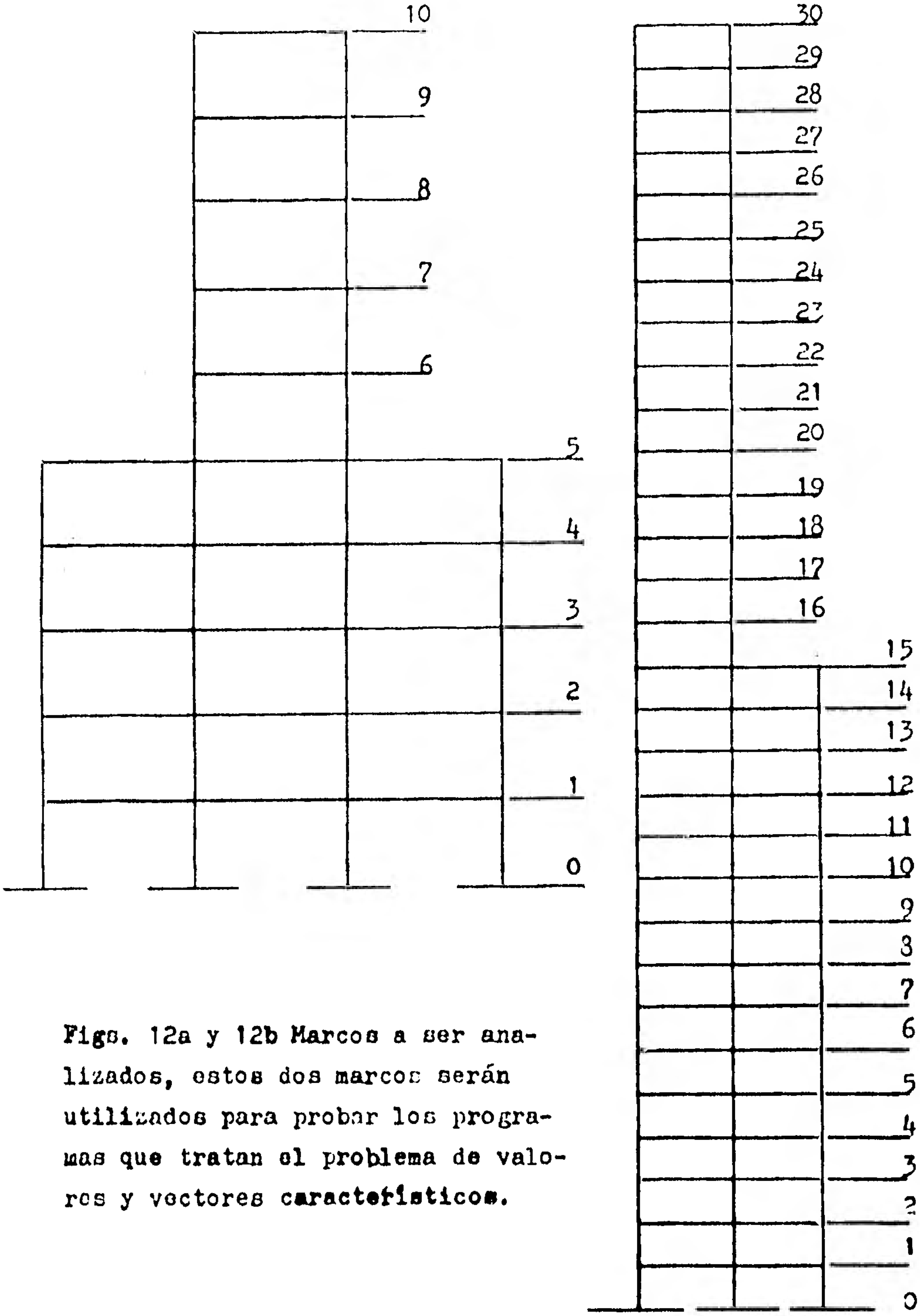
0.45
0.41
0.37
0.33
0.30
0.26
0.24
0.06

$\left[\frac{\text{Ton-seg}^2}{\text{m}}\right]$

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



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

```fortran
      SUBROUTINE CHOSIM(A,X,N1,R1,U,E1,Z1,U1,U2)
C
C     PROGRAMA
C        RESUELVE SISTEMAS DE ECUACIONES LINEALES POR EL METODO
C        GENERAL DE CHOLESKY.
C
C     REFERENCIA
C        J.S. PRZEMIENIECKI
C        "THEORY OF MATRIX STRUCTURAL ANALYSIS",
C        DEAN OF ENGINEERING AIR FORCE INSTITUTE OF TECHNOLOGY.
C        1968 BY MCGRAW-HILL, INC.
C
C     - - ENTRADA DE VARIABLES - -
C     A(N1,N1)   = MATRIZ DE COEFICIENTES
C     X(N1)      = VECTOR DE TERMINOS INDEPENDIENTES
C     N1         = NUMERO DE ECUACIONES
C
C     - - SALIDA - -
C     R1(N1)    = CONTIENE EL VECTOR SOLUCION DEL PROBLEMA
C
C
      DIMENSION A(N1,N1),U(N1,N1),E1(N1,N1)
      DIMENSION X(N1),R1(N1)
      DIMENSION U1(N1,N1),U2(N1,N1),Z1(N1,N1)
C
C     OBTIENE LA MATRIZ TRIANGULAR INFERIOR E1 Y LA MATRIZ TRIANGULAR
C     SUPERIOR U
C
      DO 10 I=1,N1
      DO 10 J=1,N1
      IF (J.NE.I) GO TO 1550
      U(I,J)=1.0
 1550 IF(J.GT.I) GO TO 1620
      S8=0.
      DO 20 K=1,J-1
   20 S8=S8+E1(I,K)*U(K,J)
      E1(I,J)=A(I,J)-S8
      GO TO 10
 1620 S9=0.
      DO 30 K1=1,I-1
   30 S9=S9+E1(I,K1)*U(K1,J)
      U(I,J)=(A(I,J)-S9)/E1(I,I)
   10 CONTINUE
C
C     AQUI SE OBTIENE LA INVERSA DE LA MATRIZ E1
C
      DO 40 I=1,N1
      DO 40 J=1,N1
      IF (J.NE.I) GO TO 1780
      Z1(I,J)=1.0/E1(I,J)
      GO TO 40
 1780 IF(J.GT.I) GO TO 1850
      S3=0.
      DO 70 L4=J,I-1
   70 S3=S3+E1(I,L4)*Z1(L4,J)
      Z1(I,J)=-S3/E1(I,I)
      GO TO 40
```

```
 1850 Z1(I,J)=0.
   40 CONTINUE
C
C     AQUI SE OBTIENE LA INVERSA DE LA MATRIZ U
C
      DO 71 I=1,N1
      DO 71 J=1,N1
      IF(J.NE.I) GO TO 2000
      U1(I,J)=1.0
      GO TO 71
 2000 IF(I.GT.J) GO TO 2070
      S4=0.
      DO 72 K=I,J-1
   72 S4=S4+U1(I,K)*U(K,J)
      U1(I,J)=-S4
      GO TO 71
 2070 U1(I,J)=0.
   71 CONTINUE
C
C     AQUI SE MULTIPLICA LAS MATRICES INVERSAS DE E1 Y U PARA OBTENER
C     LA INVERSA DE LA MATRIZ A
C
      DO 99 I=1,N1
      DO 99 J=1,N1
      S5=0.
      DO 73 K=1,N1
   73 S5=S5+U1(I,K)*Z1(K,J)
   99 U2(I,J)=S5
C
C     AQUI SE MULTIPLICA LA INVERSA DE LA MATRIZ A POR EL VECTOR DE
C     TERMINOS INDEPENDIENTES PARA OBTENER LA SOLUCION
C
      DO 74 I=1,N1
      S7=0.
      DO 75 K=1,N1
   75 S7=S7+U2(I,K)*X(K)
   74 R1(I)=S7
      RETURN
      END
```

```
@@@@@@@@@@      @@      @@      @@
@@@@@@@@@@      @@      @@@     @@
@@              @@      @@@@    @@
@@              @@      @@@@    @@
@@@@@@          @@      @@ @@@  @@
@@@@@@          @@      @@  @@@ @@
@@              @@      @@    @@@@
@@              @@      @@    @@@@
@@              @@      @@     @@@
@@              @@      @@      @@
```

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

```
      SUBROUTINE CHLSKY(A,X,N1,R1,E,B,C,D)
C
C     PROGRAMA
C        RESUELVE SISTEMAS DE ECUACIONES LINEALES POR EL METODO DE
C        CHOLESKY PARA MATRICES SIMETRICAS
C
C     REFERENCIA
C        J.S. PRZEMIENIECKI
C        "THEORY OF MATRIX STRUCTUTAL ANALYSIS",
C        DEAN OF ENGINEERING AIR FORCE INSTITUTE OF TECHNOLOGY,
C        1968 BY MCGRAW-HILL, INC.
C
C       - - ENTRADA DE VARIABLES - -
C      A(N1,N1)    = MATRIZ DE RIGIDEZ
C      X(N1)       = VECTOR DE CARGAS
C      N1          = NUMERO DE ECUACIONES
C
C     - - SALIDA - - -
C     R1(N1)       = CONTIENE EL VECTOR DE DESPLAZAMIENTOS
C
C
      DIMENSION A(N1,N1),E(N1,N1),B(N1,N1),C(N1,N1),X(N1),
     *R1(N1),D(N1)
C
C     OBTIENE LA MATRIZ TRIANGULAR SUPERIOR
C
      DO 150 I=1,N1
      DO 150 J=1,N1
      IF (J.NE.I) GO TO 25
      S=0.
      DO 20 K=1,I-1
   20 S=S+E(I,K)*E(I,K)
      IF(A(I,J).GE.S) GO TO 85
      WRITE(6,1000)
      STOP
   85 E(I,I)=SQRT(A(I,I)-S)
      GO TO 150
   25 IF(J.GT.I) GO TO 140
      S1=0.
      DO 40 K1=1,J-1
   40 S1=S1+E(I,K1)*E(J,K1)
      E(I,J)=(A(I,J)-S1)/E(J,J)
      GO TO 150
  140 E(I,J)=0.
  150 CONTINUE
C
C     OBTIENE LA INVERSA DE LA MATRIZ E Y ES B
C
      DO 50 I=1,N1
      DO 50 J=1,N1
      IF(J.NE.I) GO TO 250
      B(I,I)=1.0/E(I,I)
      GO TO 50
  250 IF(J.GT.I) GO TO 320
      S3=0.
      DO 60 L4=J,I-1
   60 S3=S3+E(I,L4)*B(L4,J)
```

```
      B(I,J)=-S3/E(I,I)
      GO TO 50
  320 B(I,J)=0.
   50 CONTINUE
C
C     SE MULTIPLICA B POR SU TRANSPUESTA PARA OBTENER A-1
C
      DO 70 I=1,N1
      DO 70 J=1,N1
      S0=0.
      DO 80 K=1,N1
   80 S0=S0+B(K,I)*B(K,J)
   70 C(I,J)=S0
C
C     MULTIPLICA A-1 POR EL VECTOR DE TERMINOS INDEPENDIENTES
C     PARA OBTENER LA SOLUCION
C
      DO 90 I=1,N1
      S7=0.
      DO 100 K=1,N1
  100 S7=S7+C(I,K)*X(K)
   90 R1(I)=S7
      RETURN
 1000 FORMAT("  SISTEMA MAL CONDICIONADO",//)
      END
```

```
@@@@@@@@@@     @@      @@      @@
@@@@@@@@@@     @@      @@@     @@
@@             @@      @@@@    @@
@@             @@      @@@@    @@
@@@@@@         @@      @@ @@@  @@
@@@@@@         @@      @@  @@@ @@
@@             @@      @@   @@@@
@@             @@      @@   @@@@
@@             @@      @@    @@@
@@             @@      @@     @@
```

PROGRAMA PARA SOLUCION DE SISTEMAS DE ECUACIONES LINEALES PARA MATRICES SIMETRICAS, USANDO UN ESQUEMA DE ALMACENAJE EN BANDA

```
      SUBROUTINE GAUBAN (A,B,NR1,MB,NC2)
C
C     PROGRAMA
C         RESUELVE SISTEMAS DE ECUACIONES LINEALES PARA MATRICES
C         SIMETRICAS Y EN BANDA, USANDO UN ESQUEMA DE ALMACENAJE
C         EN BANDA. LA MATRIZ DE RIGIDEZ ENTRA A LA SUBRUTINA DE
C         ACUERDO AL ESQUEMA DE ALMACENAJE NUMERO 1.
C         SE UTILIZA EL METODO DE GAUSS.
C
C     REFERENCIA
C         M. EN I. AYER DE LEON JUAN
C         INSTITUTO DE INGENIERIA UNAM.
C
C     LA LLAMADA A LA SUBRUTINA ES HECHA POR
C          CALL GAUBAN(A,B,NR1,MB,NC2)
C
C     - - ENTRADA DE VARIABLES
C     ES(NR1,MB)   = MATRIZ DE RIGIDEZ EN BANDA
C     B(NR1,NC2)   = MATRIZ DE CARGAS
C     NR1          = NUMERO DE ECUACIONES
C     MB           = ANCHO DE BANDA DE "A"
C     NC2          = NUMERO DE CONDICIONES DE CARGA
C
C     - - SALIDA -
C     B(NR1,NC2)   = MATRIZ DE DESPLAZAMIENTOS
C
C
      DIMENSION A(50,MB),B(50,NC2)
      N1=NR1-1
      M1=NR1-MB+1
      JS=MB
      DO 300 K=1,N1
      KK=K
      T1=A(K,1)
      T1=1./T1
      K1=K-1
      IF(K.GT.M1) JS=NR1-K1
      DO 150 J1=2,JS
      J=J1
      T2=A(K,J1)*T1
      IF(T2.EQ.0.) GO TO 150
      DO 100 I1=2,J1
      J=J-1
      I=I1+K1
  100 A(I,J)=A(I,J)-A(K,I1)*T2
  150 CONTINUE
      DO 250 J=1,NC2
      T2=B(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(K,I1)*T2
  250 CONTINUE
  300 CONTINUE
      T1=1./A(NR1,1)
      N1=NR1-1
      DO 210 K=1,NC2
```

```
      JS=1
      B(NR1,K)=B(NR1,K)*T1
      DO 155 II=1,N1
      JS=JS+1
      I=NR1-II
      IF(II.GE.MB) JS=MB
      SUM=B(I,K)
      I1=I
      DO 110 J=2,JS
      I1=I1+1
  110 SUM=SUM-A(I,J)*B(I1,K)
  155 B(I,K)=SUM/A(I,1)
  210 CONTINUE
      RETURN
      END
```

```
@@@@@@@@@@      @@      @@        @@
@@@@@@@@@@      @@      @@@       @@
@@              @@      @@@@      @@
@@              @@      @@@@      @@
@@@@@@          @@      @@ @@@    @@
@@@@@@          @@      @@   @@@  @@
@@              @@      @@    @@@@
@@              @@      @@    @@@@
@@              @@      @@     @@@
@@              @@      @@      @@
```

PROGRAMA PARA SOLUCION DE SISTEMAS DE ECUACIONES LINEALES PARA MATRICES SIMETRICAS, USANDO UN ESQUEMA DE ALMACENAJE EN BANDA

```
      SUBROUTINE SISTRI(FS,KC,NR,NC,NCC,IND)
C
C     PROGRAMA
C        RESUELVE SISTEMAS DE ECUACIONES LINEALES PARA MATRICES
C        SIMETRICAS Y EN BANDA, USANDO UN ESQUEMA DE ALMACENAJE EN
C        BANDA, LA MATRIZ DE RIGIDEZ ENTRA A LA SUBRUTINA DE ACUERDO
C        AL ESQUEMA DE ALMACENAJE NUMERO 1.
C
C     REFERENCIA
C        ING. JULIO DAMY RIOS
C        DIVISION DE ESTUDIOS DE POSGRADO UNAM.
C
C
C     LA LLAMADA ES HECHA POR
C          CALL SISTRI(FS,KC,NR,NC,NCC,IND)
C
C        - ENTRADA DE VARIABLES - -
C     FS(NR,NC)     = MATRIZ DE RIGIDEZ EN BANDA
C     KC(NR,NCC)    = MATIRZ DE CARGAS
C     NR            = NUMERO DE ECUACIONES
C     NC            = ANCHO DE BANDA DE "FS"
C     NCC           = NUMERO DE CONDICIONES DE CARGA
C     IND           = PARAMETRO DE EJECUCION
C            (1) IND=0 TRIANGULARIZA LA MATRIZ DE RIGIDEZ EN BANDA
C                      Y RESUELVE EL SISTEMA DE ECUACIONES
C            (2) IND=1 TRIANGULARIZA LA MATRIZ DE RIGIDEZ EN BANDA
C
C     - - SALIDA -
C     KC(NR,NCC)    = MATRIZ QUE CONTIENE LA SOLUCION DE SISTEMA
C
C
      DIMENSION FS(50,NC),KC(50,NCC)
      CALL TRIAN(FS,NR,NC,50,IND)
      IF(IND.EQ.1) CALL EXIT
      CALL RESUEL(FS,NR,NC,50,KC,NCC,50)
      RETURN
      END
C
C
      SUBROUTINE TRIAN(A,NN,NAN,ND1,IND)
C
C   SUBRUTINA PARA TRIANGULARIZAR MATRICES EN BANDA
C
      DIMENSION A(ND1,1)
      IND=0
      DO 20 I=1,NN
      N1=NAN
      IF(I.GT.NN-NAN+1)N1=NN-I+1
      DO 90 J=1,N1
      SUMA=0.
      K1=J+1
      K2=2
      K3=I-1
   10 IF(K3.EQ.0.OR.K1.EQ.NAN+1)GO TO 11
      SUMA=SUMA+A(K3,K1)*A(K3,K2)
      K1=K1+1
      K2=K2+1
```

```
      K3=K3-1
      GO TO 10
 11   SUMA=A(I,J)-SUMA
      IF(J.NE.1)GO TO 12
      IF(SUMA.LE.0)GO TO 13
      A(I,J)=SQRT(SUMA)
      GO TO 90
 12   A(I,J)=SUMA/A(I,1)
 90   CONTINUE
      RETURN
 13   WRITE(6,1000)I,SUMA
      IND=1
      RETURN
 1000 FORMAT(5X,"ERROR EN EL RENGLON",I5,"    SUMA=",F15.5)
      END
C
C
C
      SUBROUTINE RESUEL(A,NN,NAN,ND1,YY,NCC,ND2)
C
C    SUBRUTINA PARA RESOLVER UN SISTEMA DE ECUACIONES DADO EN BANDA
C
      DIMENSION A(ND1,1),YY(ND2,1)
      DO 200 J=1,NCC
      DO 50 I=1,NN
      SUMA=0.
      K1=2
      K3=I-1
 10   IF(K1.EQ.NAN+1.OR.K3.EQ.0)GO TO 50
      SUMA=SUMA+A(K3,K1)*YY(K3,J)
      K1=K1+1
      K3=K3-1
      GO TO 10
 50   YY(I,J)=(YY(I,J)-SUMA)/A(I,1)
      DO 100 IP=1,NN
      I=NN-IP+1
      SUMA=0.
      K1=2
      K3=I+1
 51   IF(K3.EQ.NN+1.OR.K1.EQ.NAN+1)GO TO 100
      SUMA=SUMA+A(I,K1)*YY(K3,J)
      K1=K1+1
      K3=K3+1
      GO TO 51
 100  YY(I,J)=(YY(I,J)-SUMA)/A(I,1)
 200  CONTINUE
      RETURN
      END
```

```
@@@@@@@@@@      @@      @@        @@
@@@@@@@@@@      @@      @@@       @@
@@              @@      @@@@      @@
@@              @@      @@@@      @@
@@@@@@          @@      @@ @@@    @@
@@@@@@          @@      @@  @@@   @@
@@              @@      @@   @@@@
@@              @@      @@    @@@@
@@              @@      @@     @@@
@@              @@      @@      @@
```

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

```
      SUBROUTINE COLSOL(A,V,MAXA,NN,NWK,NNM,KKK,IOUT)
C
C     PROGRAMA
C        RESUELVE SISTEMAS DE ECUACIONES LINEALES USANDO ALMACENAJE
C        COMPACTO Y ESQUEMA DE REDUCCION EN COLUMNA. LA MATRIZ DE
C        RIGIDEZ ENTRA A ESTA SUBRUTINA DE ACUERDO AL ESQUEMA DE AL
C        MACENAJE NUMERO 3.
C        SE UTILIZA EN METODO DE ELIMINACION DE GAUSS.
C
C     REFERENCIA
C        KLAUS-JURGEN BATHE AND EDWARD L. WILSON
C        "NUMERICAL METHODS IN FINITE ELEMENT ANALYSIS",
C        ENGLEWOOD CLIFFS, NEW JERSEY
C        1976 BY PRENTICE-HALL, INC.
C
C     LA LLAMADA A LA SUBRUTINA ES HECHA POR
C        CALL COLSOL(A,V,MAXA,NN,NWK,NNM,KKK,IOUT)
C
C     - - ENTRADA DE VARIABLES - -
C     A(NWK)     =MATRIZ DE RIGIDEZ ALMACENADA EN FORMA COMPACTA
C     V(NN)      =VECTOR DE CARGAS
C     MAXA(NNM)  =VECTOR QUE CONTIENE LA DIRECCION DE LOS ELEMENTOS
C                 DE LA MATRIZ DE RIGIDEZ EN A
C     NN         =NUMERO DE ECUACIONES
C     NWK        =NUMERO DE ELEMENTOS BAJO LA LINEA DE HORIZONTE
C                 DE LA MATRIZ DE RIGIDEZ
C     NNM        =NN+1
C     KKK        =SENAL DE ENTRADA
C             EQ. 1 TRIANGULARIZACION DE LA MATRIZ DE RIGIDEZ
C             EQ. 2 REDUCCION Y SUSTITUCION REGRESIVA DEL VECTOR DE CARGA
C     IOUT       =NUMERO DE DISPOSITIVO DE SALIDA
C
C     - - SALIDA - -
C     A(NWK)     =D Y L  FACTORES DE LA MATRIZ DE RIGIDEZ
C     V(NN)      =VECTOR DE DESPLAZAMIENTOS
C
C
      DIMENSION A(NWK),V(1),MAXA(1)
C
C     EJECUTA L*D*L(T) LA FACTORIZACION DE LA MATRIZ DE RIGIDEZ
C
      IF(KKK-2) 40,150,150
   40 DO 140 N=1,NN
      KN=MAXA(N)
      KL=KN+1
      KU=MAXA(N+1)-1
      KH=KU-KL
      IF (KH) 110,90,50
   50 K=N-KH
      IC=0
      KLT=KU
      DO 80 J=1,KH
      IC=IC+1
      KLT=KLT-1
      KI=MAXA(K)
      ND=MAXA(K+1)-KI-1
      IF (ND) 80,80,60
```

```
   60 KK=MIN0(IC,ND)
      C=0.0
      DO 70 L=1,KK
   70 C=C+A(KI+L)*A(KLT+L)
      A(KLT)=A(KLT)-C
   80 K=K+1
   90 K=N
      B=0.0
      DO 100 KK=KL,KU
      K=K-1
      KI=MAXA(K)
      C=A(KK)/A(KI)
      B=B+C*A(KK)
  100 A(KK)=C
      A(KN)=A(KN)-B
  110 IF (A(KN)) 120,120,140
  120 WRITE(IOUT,2000) N,A(KN)
      STOP
  140 CONTINUE
C
C     REDUCE EL VECTOR DE CARGAS
C
  150 DO 180 N=1,NN
      KL=MAXA(N)+1
      KU=MAXA(N+1)-1
      IF (KU-KL) 180,160,160
  160 K=N
      C=0.0
      DO 170 KK=KL,KU
      K=K-1
  170 C=C+A(KK)*V(K)
      V(N)=V(N)-C
  180 CONTINUE
C
C     SUSTITUCION REGRESIVA
C
      DO 200 N=1,NN
      K=MAXA(N)
  200 V(N)=V(N)/A(K)
      IF(NN.EQ.1) RETURN
      N=NN
      DO 230 L=2,NN
      KL=MAXA(N)+1
      KU=MAXA(N+1)-1
      IF (KU-KL) 230,210,210
  210 K=N
      DO 220 KK=KL,KU
      K=K-1
  220 V(K)=V(K)-A(KK)*V(N)
  230 N=N-1
      RETURN
 2000 FORMAT(//"ALTO-LA MATRIZ DE RIGIDEZ NO ES DEFINIDA POSITIVA",//
     *          "PIVOTE NEGATIVO EN LA ECUACION",I4,//
     *          "PIVOTE=",E20.12)
      END
```

```
@@@@@@@@@@     @@     @@       @@
@@@@@@@@@@     @@     @@@      @@
@@             @@     @@@@     @@
@@             @@     @@@@     @@
@@@@@@         @@     @@ @@@   @@
@@@@@@         @@     @@  @@@  @@
@@             @@     @@   @@@@
@@             @@     @@    @@@@
@@             @@     @@     @@@
```

PROGRAMA PARA SOLUCION DE SISTEMAS DE ECUACIONES LINEALES PARA MATRICES SIMETRICAS, USANDO UN ESQUEMA DE ALMACENAJE EN BANDA

```
      SUBROUTINE NUMSYM (A,B,NA,NEQ,MBAND,KEX)
C
C     PROGRAMA
C        RESUELVE SISTEMAS DE ECUACIONES LINEALES PARA MATRICES SIME
C        TRICAS USANDO ALMACENAJE EN BANDA PARA LA MATRIZ DE RIGIDEZ.
C        LA MATRIZ DE RIGIDEZ ENTRA A ESTA SUBRUTINA DE ACUERDO AL
C        ESQUEMA DE ALMACENAJE NUMERO 1.
C        SE UTILIZA EL METODO DE ELIMINACION DE GAUSS
C
C     REFERENCIA
C        DIAMBAR P. MONDKAR AND GRAHAM H. POWELL
C        "TOWARDS OPTIMAL IN CORE EQUATION SOLVING",
C        COMPUTERS & STRUCTURES, VOL. 4, PP. 531-548,
C        PERGAMON PRESS 1974, PRINTED IN GREAT BRITAIN
C
C     LA LLAMADA A LA SUBRUTINA ES HECHA POR
C        CALL NUMSYM(A,B,NA,NEQ,MBAND,KEX)
C
C     - - EMTRADA DE VARIABLES - -
C     NEQ      = NUMERO DE ECUACIONES
C     MBAND    = MAXIMO ANCHO DE LA MITAD DE LA BANDA
C     A        = ES UN ARREGLO DE DIMENSION NEQ X MBAND, LA CUAL CON-
C                TIENE LA MATRIZ DE COEFICIENTES ALMACENADA EN LA FORMA
C                NUMERO 1
C     B        = VECTOR DE DIMENSION NEQ, EL CUAL CONTIENE EL VECTOR
C                DE CARGA
C     NA       = VECTOR DE DIMENSION NEQ, EL CUAL ENTRA QUE CONTIENE VA
C                LORES ARBITRARIOS
C     KEX      = PARAMETROS DE EJECUCION.
C                 (1) KEX=1 REDUCCION DE LA MATRIZ DE COEFICIENTES
C                 (2) KEX=2 REDUCCION DEL VECTOR DE CARGA Y SUSTITUCION
C                           REGRESIVA
C                 (3) KEX=3 REDUCCION DE LA MATRIZ DE COEFICIENTES Y DEL
C                           VECTOR DE CARGAS Y SUSTITUCION REGRESIVA
C     - - SALIDA - -
C     VECTOR B QUE CONTIENE EL VECTOR SOLUCION
C     VECTOR NA QUE CONTIENE LOS VALORES DEL NUMERO DE RENGLONES DEL
C     PRIMER ELEMENTO DIFERENTE DE CERO ENTRANDO EN CADA COLUMNA
C     DE LA MATRIZ DE COEFICIENTES
C     ARREGLO A QUE CONTIENE LOS VALORES MODIFICADOS DE LA MATRIZ
C
C
      DIMENSION A(700),B(NEQ),NA(NEQ)
      NEQQ=NEQ-1
      MBAA=MBAND 1
      GO TO (10,110,10) KEX
C
C     REDUCE LA MATIZ DE COEFICIENTES A
C
   10 NADR=1
      DO 30 N=2,MBAA
      NADR=NADR+NEQ
      JADR=NADR
      N1=N-1
      DO 20 I=1,N1
      IF(A(JADR).NE.0) GO TO 30
   20 JADR=JADR-NEQQ
```

```
      I=N1
   30 NA(N)=I
      M=0
      NADR=NEQ*MBAA
      DO 50 N=MBAND,NEQ
      M=M+1
      NADR=NADR+1
      JADR=NADR
      N1=N-1
      DO 40 I=M,N1
      IF (A(JADR).NE.0.) GO TO 50
   40 JADR=JADR-NEQQ
      I=N1
   50 NA(N)=I
      NA(1)=1
      JI=0
      DO 100 J=2,NEQ
      IF=NA(J)
      IF1=IF+1
      IL=J-1
      JI=JI+NEQ
      JK=JI+NEQ
      IFN=IF*NEQQ
      IF (IF1.GT.IL) GO TO 80
      JIA=JI-IFN+1
      IKA=JK-IFN
      NN=IF1*NEQ-JK
      DO 70 I=IF1,IL
      KF=MAX0(NA(I),IF)
      JKA=JK-KF*NEQQ
      AA=0.0
      DO 60 K=IKA,JKA,NEQQ
   60 AA=AA+A(K)*A(NN+K)
      A(JIA)=A(JIA)-AA
      NN=NN+NEQ
      IKA=IKA-NEQQ
   70 JIA=JIA-NEQQ
   80 JKA=JK-IFN
      AA=0.0
      DO 90 K=IF,IL
      CC=A(JKA)/A(K)
      AA=AA+A(JKA)*CC
      A(JKA)=CC
   90 JKA=JKA-NEQQ
      A(J)=A(J)-AA
  100 CONTINUE
      GO TO (190,190,110) KEX
C
C     REDUCCION DEL VECTOR DE CARGA, Y SUSTITUCION REGRESIVA
C
  110 DO 120 N=1,NEQQ
      IF(B(N).NE.0.) GO TO 130
  120 CONTINUE
      N=NEQQ
  130 N1=N+1
      NADR=N1*NEQ
      DO 150 I=N1,NEQ
      KF=MAX0(NA(I),N)
      KL=I-1
      JADR=NADR-KF*NEQQ
```

```
      BB=0.0
      DO 140 K=KF,KL
      BB=BB+A(JADR)*B(K)
  140 JADR=JADR-NEQQ
      B(I)=B(I)-BB
  150 NADR=NADR+NEQ
      DO 160 I=N,NEQ
  160 B(I)=B(I)/A(I)
      NN=NEQ
      NADR=NN*NEQ
      DO 180 N=1,NEQQ
      KF=NA(NN)
      KL=NN-1
      JADR=NADR-KF*NEQQ
      BB=B(NN)
      DO 170 K=KF,KL
      B(K)=B(K)-A(JADR)*BB
  170 JADR=JADR-NEQQ
      NADR=NADR-NEQ
  180 NN=NN-1
  190 RETURN
      END
```

```
@@@@@@@@@@      @@      @@      @@
@@@@@@@@@@      @@      @@@     @@
@@              @@      @@@@    @@
@@              @@      @@@@    @@
@@@@@@          @@      @@ @@@  @@
@@@@@@          @@      @@  @@@ @@
@@              @@      @@    @@@@
@@              @@      @@    @@@@
@@              @@      @@     @@@
@@              @@      @@      @@
```

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

```
      SUBROUTINE OPTSOL (A,B,NA,NEQ,KEX)
C
C     PROGRAMA
C         RESUELVE SISTEMAS DE ECUACIONES LINEALES PARA MATRICES
C         SIMETRICAS USANDO ALMACENAJE COMPACTO Y ESQUEMA DE REDUC-
C         CION EN COLUMNA. LA MATRIZ DE RIGIDEZ ENTRA A ESTA SUBRUTI-
C         NA DE ACUERDO AL ESQUEMA DE ALMACENAJE NUMERO 2.
C         SE UTILIZA EL METODO DE ELIMINACION DE GAUSS.
C
C     REFERENCIA
C         DIGAMBAR P. MONDKAR AND GRAHAM H. POWELL.
C         "TOWARDS OPTIMAL IN-CORE EQUATION SOLVING",
C         COMPUTERS & STRUCTURES, VOL. 4 PP. 531-540,
C         PERGAMON PRESS 1974. PRINTED IN GREAT BRITAIN
C
C     LA LLAMADA A LA SUBRUTINA ES HECHA POR
C        CALL OPTSOL(A,B,NA,NEQ,KEX)
C
C     - - ENTRADA DE VARIABLES - -
C     NEQ    = NUMERO DE ECUACIONES
C     A      = VECTOR QUE CONTIENE LA MATRIZ DE COEFICIENTES USANDO EL
C              ESQUEMA DE ALMACENAJE NUMERO 2
C     B      = VECTOR DE DIMENSION NEQ, EL CUAL CONTIENE EL VECTOR
C              DE CARGA
C     NA     = VECTOR DE DIMENSION NEQ, CONTIENE LOS VALORES DE NUMERO
C              DEL RENGLON DEL PRIMER ELEMENTO DIFERENTE DE CERO DE CADA
C              COLUMNA EN LA MATRIZ DE COEFICIENTES
C     KEX    = PARAMETROS DE EJECUCION
C              (1) KEX=1 REDUCCION DE LA MATRIZ DE COEFICIENTES
C              (2) KEX=2 REDUCCION DEL VECTOR DE CARGA Y SUSTITUCION
C                        REGRESIVA
C              (3) KEX=3 REDUCCION DE LA MATRIZ DE COEFICIENTES Y EL VEC
C                        TOR DE CARGA Y SUSTITUCION REGRESIVA
C     - - SALIDA - -
C       VECTOR B QUE CONTIENE EL VECTOR SOLUCION
C       VECTOR NA QUE CONTIENE LAS DIRECCIONES DE LOS ELEMENTOS DE LA
C       DIAGONAL DE LA MATRIZ DE COEFICIENTES EN EL VECTOR A
C       VECTOR A QUE CONTIENE LA MATRIZ DE COEFICIENTES MODIFICADA
C
C
      DIMENSION A(1),B(NEQ),NA(NEQ)
      NEQQ=NEQ-1
      GO TO (10,100,10) KEX
C
C     REDUCE LA MATRIZ DE COEFICIENTES
C
   10 NSTO=1
      DO 20 J=2,NEQ
      NSTO=NSTO+J-NA(J)+1
   20 NA(J)=NSTO
      NA(1)=1
      II=1
      NAJP=NA(1)
      DO 90 J=2,NEQ
      NAJ=NA(J)
      JK=NAJ-J
      IF=1-JK+NAJP
```

```
      IF (IF.GE.J) GO TO 80
      IF1=IF+1
      IF (IF1.GT.IL) GO TO 60
      JIA=2+NAJP
      NAIP=NA(IF)
      KL=IF+JK
      DO 50 I=IF1,IL
      NAI=NA(I)
      IK=NAI-I
      II=I+1-NAI+NAIP
      IF(II.GE.I) GO TO 40
      KF=MAX0(II,IF)+JK
      JJ=IK-JK
      AA=0.0
      DO 30 K=KF,KL
   30 AA=AA+A(K)*A(JJ+K)
      A(JIA)=A(JIA)-AA
   40 JIA=JIA+1
      KL=KL+1
   50 NAIP=NAI
   60 KF=JK+IF
      KL=NAJ-1
      AA=0.0
      DO 70 K=KF,KL
      NAI=NA(IF)
      CC=A(K)/A(NAI)
      AA=AA+A(K)*CC
      A(K)=CC
   70 IF=IF+1
      A(NAJ)=A(NAJ)-AA
   80 IL=IL+1
   90 NAJP=NAJ
      GO TO (200,200,100),KEX
C
C     REDUCCION DEL VECTOR DE CARGA Y SUSTITUCION REGRESIVA
C
  100 DO 110 N=1,NEQQ
      IF(B(N).NE.0.) GO TO 120
  110 CONTINUE
      N=NEQQ
  120 N1=N+1
      II=N1+1
      KL=N
      NAIP=NA(N)
      DO 150 I=N1,NEQ
      NAI=NA(I)
      II=II-NAI+NAIP
      IF(II.GE.I) GO TO 140
      KF=MAX0(II,N)
      IK=NAI-I
      IKA=IK+KF
      BB=0.0
      DO 130 K=KF,KL
      BB=BB+A(IKA)*B(K)
  130 IKA=IKA+1
      B(I)=B(I)-BB
  140 II=II+1
      KL=KL+1
  150 NAIP=NAI
      DO 160 I=N,NEQ
```

```
      NAI=NA(I)
  160 B(I)=B(I)/A(NAI)
      J=NEQ
      NAJ=NA(NEQ)
      DO 190 I=1,NEQQ
      NAJP=NA(J-1)
      JKA=NAJP+1
      II=J-NAJ+JKA
      IF (II.GE.J) GO TO 180
      KL=J-1
      BB=B(J)
      DO 170 K=II,KL
      B(K)=B(K)-A(JKA)*BB
  170 JKA=JKA+1
  180 J=J-1
  190 NAJ=NAJP
  200 RETURN
      END
```

```
@@@@@@@@@@      @@      @@       @@
@@@@@@@@@@      @@      @@@      @@
@@              @@      @@@@     @@
@@              @@      @@@@     @@
@@@@@@          @@      @@ @@@   @@
@@@@@@          @@      @@  @@@ @@
@@              @@      @@    @@@@
@@              @@      @@    @@@@
@@              @@      @@     @@@
@@              @@      @@      @@
```

PROGRAMA PARA CALCULAR LAS FORMAS MODALES Y FRECUENCIAS DE UN SISTEMA ESTRUCTURAL CON UNA MATRIZ DE MASAS DIAGONAL

```
      SUBROUTINE EVAL1(A,SM,VAL,VEC,NEQ,NM,LM,MI,D,CO,NA,
     *ALLOW,ND,IW,C)
C
C     PROGRAMA
C      CALCULA LAS FORMAS MODALES Y FRECUENCIAS DE UN SISTEMA ESTRU
C      CTURAL CON UNA MATRIZ DE MASAS DIAGONAL.
C      ESTA SUBRUTINA UTILIZA EL METODO DE PODER INVERSO CON CAMBIOS(THE
C      POWER METHOD WITH SHIFTS), PARA CALCULAR LOS VALORES Y VECTORES
C      CARACTERISTICOS.
C
C     REFERENCIA
C      J.L. HUMAR
C      "EIGENVALUE PROGRAMS FOR BUILDING STRUCTURES",
C      COMPUTER & STRUCTURES, VOL. 8, PP. 75-91,
C      PERGAMON PRESS 1978, PRINTED IN GREAT BRITAIN
C
C     LA LLAMADA A LA SUBRUTINA ES HECHA POR
C      CALL EVAL1(A,SM,VAL,VEC,NEQ,NM,LM,MI,D,CO,NA,ALLOW,ND,IW,C)
C
C         - ENTRADA DE VARIABLES -
C     ND    =DIMENSION ASIGNADA A LA MATRIZ DE RIGIDEZ EN LA LLAMADA
C             AL PROGRAMA.
C     NEQ   =ORDEN DE LA MATRIZ DE RIGIDEZ.
C     NM    =NO. DE MODOS A SER CALCULADOS.
C     LM    =NO. DE MODOS A SER ENCONTRADOS DE UN ORIGEN.
C     MI    =MAXIMO NUMERO DE ITERACIONES A SER USADAS EN EL CALCULO
C             DE UNA FORMA MODAL.
C     IW    =NUMERO CORRESPONDIENTE A LA UNIDAD DE SALIDA.
C     A     =MATRIZ DE RIGIDEZ.
C     SM    =VECTOR QUE CONTIENE LA DIAGONAL DE LA MATRIZ DE MASAS.
C     VAL   =ARREGLO QUE CONTIENE LAS FRECUENCIAS.
C     VEC   =MATRIZ QUE CONTIENE LAS FORMAS MODALES; CADA FORMA MODAL
C             FORMA UNA COLUMNA DE LA MATRIZ.
C     D     =MATRIZ DINAMICA.
C     CO    =ARREGLO QUE CONTIENE UN VECTOR ARBITRARIO.
C     NA    =ARREGLO QUE CONTIENE LOS VALORES DE LOS NUMEROS DE RENGLON
C             DEL PRIMER ELEMENTO DIFERENTE DE CERO QUE ENTRA EN
C             CADA COLUMNA DE LA MATRIZ DE RIGIDEZ.
C     ALLOW=NUMERO DE PUNTO FLOTANTE ESPECIFICANDO EL CRITERIO DE CON
C             VERGENCIA.
C     C     =ARREGLO DE ALMACENAJE TEMPORAL USADO PARA ARCHIVAR LA DIA
C             GONAL DE LA MATRIZ DE MASAS.
C
C         - ENTRADA DE VARIABLES -
C     VAL,VEC,D,CO,NA, Y C PUEDEN CONTENER VALORES ARBITRARIOS.
C
C         - SALIDA DE VARIABLES -
C     VAL   =CONTIENE LAS FRECUENCIAS.
C     VEC   =CONTIENE LAS FORMAS MODALES.
C     NA    =CONTIENE LOS VALORES DEL NUMERO DE RENGLON DEL PRIMER
C             ELEMENTO DIFERENTE DE CERO QUE ENTRA EN CADA COLUMNA
C             DE LA MATRIZ DE RIGIDEZ A.
C
C
      DIMENSION A(ND,ND),SM(1),VEC(ND,NM),VAL(1),D(ND,ND)
     *,CO(1),NA(1),C(1)
C
```

```
C     ARCHIVA LA DIAGONAL DE A EN C.
C
      DO 10 I=1,NEQ
      C(I)=A(I,I)
   10 CONTINUE
C
C     FORMA LA MATRIZ DINAMICA.
C
      CALL DYNO(A,NEQ,NA,NB,D,SM,C)
      SH=0.
      K=1
      LL=0
      R1=1.E-06
      NVT=0
C
C     INICIALIZA UN VECTOR PRUEBA.
C
   20 DO 30 I=1,NEQ
      CO(I)=1.
   30 CONTINUE
      NT=1
C
C     OBTIENE UN NUEVO VECTOR DE PRUEBA.
C
   40 DO 50 I=1,NEQ
      VEC(I,K)=0.
      DO 50 J=1,NEQ
      VEC(I,K)=VEC(I,K)+D(I,J)*CO(J)
   50 CONTINUE
C
C     NOMALIZA EL VECTOR DE PRUEBA Y DIVIDE ENTRE EL
C     ELEMENTO MAS GRANDE.
C
      NT=1
      CMAX=ABS(VEC(1,K))
      DO 60 I=2,NEQ
      IF(ABS(VEC(I,K)).LT.CMAX) GO TO 60
      CMAX=ABS(VEC(I,K))
      NT=I
   60 CONTINUE
      TT=CO(NT)/VEC(NT,K)
      TTS=TT/ABS(TT)
      CMAX=CMAX*TTS
      DO 70 I=1,NEQ
      VEC(I,K)=VEC(I,K)/CMAX
   70 CONTINUE
C
C     PRUEBA SI EL CAMBIO DE ORIGEN ES REQUERIDO.
C
      IF(LL.LT.LM) GO TO 80
      IF(NVT.EQ.1) GO TO 80
      IF(NI.LT.10) GO TO 80
      VT=SH+ABS(1./CMAX)
      KT=K-1
      VVT=SQRT(VT)
      IF(ABS(VVT-VAL(KT)).GT..05*VAL(KT)) GO TO 110
      NVT=1
C
C     PRUEBA LA CONVERGENCIA.
C
```

```
   80 ERMAX=0.
      DO 90 I=1,NEQ
      IF(ABS(VEC(I,K)).LT.R1) GO TO 90
      ERR=(CO(I)-VEC(I,K))/VEC(I,K)
      ERR=ABS(ERR)
      IF(ERR.GT.ERMAX) ERMAX=ERR
   90 CONTINUE
      IF(ERMAX.LE.ALLOW) GO TO 150
      IF(NI.GT.MI) GO TO 150
      DO 100 I=1,NEQ
      CO(I)=VEC(I,K)
  100 CONTINUE
      NI=NI+1
      GO TO 40
C
C     AQUI OBTIENE EL CAMBIO DE ORIGEN Y LA NUEVA MATRIZ DINAMICA.
C
  110 SH=(VT+VAL(KT)+VAL(KT))/2.
      DO 130 I=1,NEQ
      A(I,I)=A(I,I)-SH*SM(I)
      CO(I)=VEC(I,K)
  130 CONTINUE
      CALL DYNO(A,NEQ,NA,ND,D,SM,E)
C
C     REVISA LA MATRIZ DINAMICA PARA LLEVAR FUERA LAS FORMAS MODALES
C     YA DETERMINADAS.
C
      KK=K-1
      DO 140 N=1,KK
      HAT=1./(VAL(N)*VAL(N)-SH)
      DO 140 I=1,NEQ
      DO 140 J=1,NEQ
      D(I,J)=D(I,J)-HAT*VEC(J,N)*VEC(I,N)*SM(J)
  140 CONTINUE
      LI=0
      NI=NI+1
      GO TO 40
C
C     AQUI CUANDO LA ITERACION HA CONVERGIDO CALCULA LA NO DIMENSIONAL
C     FRECUENCIA Y EL VECTOR MODAL NORMALIZADO.
C
  150 HECTOR=1
      VAL(K)=SQRT(SH+1./CMAX)
      ALFA=0.
      DO 160 I=1,NEQ
      ALFA=ALFA+VEC(I,K)*VEC(I,K)*SM(I)
  160 CONTINUE
      BETA=SQRT(ALFA)
      DO 170 I=1,NEQ
      VEC(I,K)=VEC(I,K)/BETA
  170 CONTINUE
      IF(K-NM) 180,200,200
  180 CONTINUE
C
C     REVISA LA MATRIZ DINAMICA PARA LLEVAR FUERA LAS ULTIMAS FORMAS
C     MODALES A SER DETERMINADAS.
C
      DO 190 I=1,NEQ
      DO 190 J=1,NEQ
      D(I,J)=D(I,J)-CMAX*VEC(I,K)*VEC(J,K)*SM(J)
```

```
  190 CONTINUE
      K=K+1
      LL=LL+1
      NVT=0
      GO TO 20
  200 RETURN
  301 FORMAT (" MODO NUMERO.......................=",I4)
  302 FORMAT (" NUMERO DE ITERACION..................=",I4)
  303 FORMAT (" DIFERENCIA MAXIMA ENTRE SUSECIVOS",
     * " VECTORES DE PRUEBA ...=",E10.7,//)
      END
      SUBROUTINE DYNO(A,NEQ,NA,ND,D,SM,C)
C
C     FORMA LA MATRIZ DINAMICA
C
      DIMENSION A(1),D(1),NA(1),SM(1),C(1)
      CALL FORE(A,NEQ,NA,ND)
      IT=0
      DO 20 J=2,NEQ
      IL=J-1
      IT=IT+ND
      JADR=J
      DO 20 I=1,IL
      IADR=IT+I
      D(IADR)=0.
      D(JADR)=0.
      JADR=JADR+ND
   20 CONTINUE
      JADR=1
      DO 30 J=1,NEQ
      D(JADR)=SM(J)
      JADR=JADR+ND+1
   30 CONTINUE
      NK=1
      DO 40 J=1,NEQ
      NB=J
      CALL BACK(A,NEQ,NA,NB,ND,D(NK))
      NK=NK+ND
   40 CONTINUE
      IT=0
      DO 50 J=2,NEQ
      IL=J-1
      IT=IT+ND
      JADR=J
      DO 50 I=1,IL
      IADR=IT+I
      A(IADR)=A(JADR)
      JADR=JADR+ND
   50 CONTINUE
      JADR=1
      DO 60 J=1,NEQ
      A(JADR)=C(J)
      JADR=JADR+ND+1
   60 CONTINUE
      RETURN
      END
      SUBROUTINE FORE(A,NEQ,NA,ND)
C
C     CONSTRUYE EL VECTOR NA Y LLEVA FUERA LA REDUCCION FRONTAL
C     DE LA MATRIZ A USANDO EL METODO DE CROUT.
```

```
C
      DIMENSION A(1),NA(1)
      NADR=1
      DO 20 J=2,NEQ
      NADR=NADR+ND
      LIMIT=NADR+J-1
      N1=1
      DO 10 I=NADR,LIMIT
      IF (A(I).NE.0) GO TO 20
      N1=N1+1
   10 CONTINUE
   20 NA(J)=N1
      NA(1)=1
      JT=0
      DO 70 J=2,NEQ
      JT=JT+ND
      II=NA(J)
      IL=J-1
      IF (II.GT.IL) GO TO 70
      IN=II+1
      JJ=JT+J
      IF(IN.GT.IL) GO TO 50
      DO 40 I=IN,IL
      M=NA(I)
      KI=MAX0(II,M)
      KL=I-1
      IT=KL*ND
      SUM=0.
      DO 30 K=KI,KL
      IADR=IT+K
      JADR=JT+K
      SUM=SUM+A(IADR)*A(JADR)
   30 CONTINUE
      IJ=JT+I
      A(IJ)=A(IJ)-SUM
   40 CONTINUE
   50 SUM=0.
      NADR=(II-1)*ND+II
      DO 60 K=II,IL
      JADR= JT+K
      C=A(JADR)/A(NADR)
      SUM=SUM+A(JADR)*C
      A(JADR)=C
      NADR=NADR+ND+1
   60 CONTINUE
      A(JJ)=A(JJ)-SUM
   70 CONTINUE
      RETURN
      END
      SUBROUTINE BACK (A,NEQ,NA,NB,ND,B)
C
C     LLEVA FUERA LA SUSTITUCION REGRESIVA Y CAMBIA EL VECTOR
C     SOLUCION EN B.
C
      DIMENSION A(1),NA(1),B(1)
      IN=NB+1
      IF (IN.GT.NEQ) GO TO 30
      DO 20 I=IN,NEQ
      M=NA(I)
      KI=MAX0(M,NB)
```

```
      KL=I-1
      IF(KI.GT.KL) GO TO 20
      IT=KL*ND
      SUM=0.
      DO 10 K=KI,KL
      IADR=IT+K
      SUM=SUM+A(IADR)*B(K)
   10 CONTINUE
      B(I)=B(I)-SUM
   20 CONTINUE
   30 IADR= (NB-1)*ND+NB
      DO 40 I=NB,NEQ
      B(I)=B(I)/A(IADR)
      IADR=IADR+ND+1
   40 CONTINUE
      I=NEQ
      IT=NEQ*ND
   50 KI=NA(I)
      KL=I-1
      IT=IT-ND
      IF(KI.GT.KL) GO TO 70
      DO 60 K=KI,KL
      IADR=IT+K
      B(K)=B(K)-A(IADR)*B(I)
   60 CONTINUE
   70 IF(I.EQ.2) GO TO 80
      I=I-1
      GO TO 50
   80 RETURN
      END
```

FIN

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

```
      SUBROUTINE JACOBI(A,B,X,EIGV,D,N,RTOL,NSMAX,IFPR,IOUT)
C
C     PROGRAMA  .
C        PARA RESOLVER EL PROBLEMA GENERALIZADO DE VALORES Y VECTORES
C        CARACTERISTICOS USANDO EL METODO GENERALIZADO DE ITERACION DE
C        JACOBI.
C
C      REFERENCIA
C         KLAUS-JURGEN BATHE AND EDWARD L. WILSON
C         "NUMERICAL METHODS IN FINITE ELEMENT ANALYSIS",
C         ENGLEWOOD CLIFFS, NEW JERSEY,
C         1976 BY PRENTICE HALL, INC.
C
C      LA LLAMADA A LA SUBRUTINA ES HECHA POR
C         CALL JACOBI(A,B,X,EIGV,D,N,RTOL,NSMAX,IFPR,IOUT)
C
C      - ENTRADA DE VARIABLES
C     A(N,N)    =MATRIZ DE RIGIDEZ(POSITIVA DEFINIDA)
C     B(N,N)    =MATRIZ DE MASAS(POSITIVA DEFINIDA)
C     X(N,N)    =MATRIZ QUE CONTIENE LOS VECTORES CARACTERISTICOS EN LA
C                SALIDA
C     EIGV(N)   =VECTOR QUE CONTIENE LOS VALORES CARACTERISTICOS EN LA
C                SALIDA
C     D(N)      =VECTOR DE TRABAJO
C     N         =ORDEN DE LAS MATRICES A Y B
C     RTOL      =TOLERANCIA EN LA CONVERGENCIA(USUALMENTE 10** 12)
C     NSMAX     =MAXIMO NUMERO DE ROTACIONES (USUALMENTE 15)
C     IFPR      =BANDERA PARA ESCRIBIR DURANTE LA ITERACION
C         EQ.=0 NO ESCRIBIR
C         EQ.=1 ESCRIBE LOS RESULTADOS INTERMEDIOS
C     IOUT      =NUMERO CORRESPONDIENTE A LA UNIDAD DE SALIDA
C
C     - - SALIDA DE VARIABLES - -
C     A(N,N)    =MATRIZ DE RIGIDEZ DIAGONALIZADA
C     B(N,N)    =MATRIZ DE MASAS DIAGONALIZADA
C     X(N,N)    =VECTORES CARACTERISTICOS ALMACENADOS POR COLUMNAS
C     EIGV(N)   =VECTORES CARACTERISTICOS
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      ABS(X)=DABS(X)
      SQRT(X)=DSQRT(X)
      DIMENSION A(N,N),B(N,N),X(N,N),EIGV(N),D(N)
C
C     INICIALIZA LAS MATRICES DE VALORES Y VECTORES CARACTERISTICOS.
C
      DO 10 I=1,N
      IF(A(I,I).GT.0. .AND. B(I,I).GT.0) GO TO 4
      WRITE (IOUT,2020)
      STOP
    4 D(I)=A(I,I)/B(I,I)
   10 EIGV(I)=D(I)
      DO 31 I=1,N
      DO 20 J=1,N
   20 X(I,J)=0.
   31 X(I,I)=1.
      IF(N.EQ.1) RETURN
```

```
      C
      C     INICIA LA CUENTA DE LAS ROTACIONES Y COMIENZA LA ITERACION
      C
            NSWEEP=0
            NR=N-1
         40 NSWEEP=NSWEEP+1
            IF(IFPR.EQ.1)WRITE(IOUT,2000) NSWEEP
      C
      C     REVISA SI EL PRESENTE ELEMENTO FUERA DE LA DIAGONAL ES LO BASTANTE
      C     GRANDE PARA HACERLO CERO
      C
            EPS=(0.01**NSWEEP)**2
            DO 210 J=1,NR
            JJ=J+1
            DO 210 K=JJ,N
            EPTOLA=(A(J,K)*A(J,K))/(A(J,J)*A(K,K))
            EPTOLB=(B(J,K)*B(J,K))/(B(J,J)*B(K,K))
            IF((EPTOLA.LT.EPS).AND.(EPTOLB.LT.EPS)) GO TO 210
      C
      C     SI SE REQUIERE HACER CERO,CALCULA LA MATRIZ DE ROTACION ELEMENTOS
      C     CA Y CG
      C
            AKK=A(K,K)*B(J,K)-B(K,K)*A(J,K)
            AJJ=A(J,J)*B(J,K)-B(J,J)*A(J,K)
            AB=A(J,J)*B(K,K)-A(K,K)*B(J,J)
            CHECK=(AB*AB+4.*AKK*AJJ)/4.
            IF(CHECK) 50,60,60
         50 WRITE(IOUT,2020)
            STOP
         60 SQCH=SQRT(CHECK)
            D1=AB/2.+SQCH
            D2=AB/2.-SQCH
            DEN=D1
            IF(ABS(D2).GT.ABS(D1))DEN=D2
            IF(DEN) 80,70,80
         70 CA=0.
            CG=-A(J,K)/A(K,K)
            GO TO 90
         80 CA=AKK/DEN
            CG=-AJJ/DEN
      C
      C     EJECUTA LA ROTACION GENERALIZADA A CERO DEL PRESENTE ELEMENTO
      C     FUERA DE LA DIAGONAL
      C
         90 IF(N-2) 100,190,100
        100 JP1=J+1
            JM1=J-1
            KP1=K+1
            KM1=K-1
            IF(JM1-1) 130,110,110
        110 DO 120 I=1,JM1
            AJ=A(I,J)
            BJ=B(I,J)
            AK=A(I,K)
            BK=B(I,K)
            A(I,J)=AJ+CG*AK
            B(I,J)=BJ+CG*BK
            A(I,K)=AK+CA*AJ
        120 B(I,K)=BK+CA*BJ
        130 IF(KP1-N) 140,140,160
```

```
  140 DO 150 I=KP1,N
      AJ=A(J,I)
      BJ=B(J,I)
      AK=A(K,I)
      BK=B(K,I)
      A(J,I)=AJ+CG*AK
      B(J,I)=BJ+CG*BK
      A(K,I)=AK+CA*AJ
  150 B(K,I)=BK+CA*BJ
  160 IF(JP1-KM1) 170,170,190
  170 DO 180 I=JP1,KM1
      AJ=A(J,I)
      BJ=B(J,I)
      AK=A(I,K)
      BK=B(I,J)
      A(J,I)=AJ+CG*AK
      B(J,I)=BK+CA*BJ
      A(I,K)=AK+CA*AJ
  180 B(I,K)=BK+CA*BJ
  190 AK=A(K,K)
      BK=B(K,K)
      A(K,K)=AK+2.*CA*A(J,K)+CA*CA*A(J,J)
      B(K,K)=BK+2.*CA*B(J,K)+CA*CA*B(J,J)
      A(J,J)=A(J,J)+2.*CG*A(J,K)+CG*CG*AK
      B(J,J)=B(J,J)+2.*CG*B(J,K)+CG*CG*BK
      A(J,K)=0.
      B(J,K)=0.
C
C     CALCULA LA NUEVA MATRIZ DE VECTORES CARACTERISTICOS DESPUES DE
C     CADA ROTACION
C
      DO 200 I=1,N
      XJ=X(I,J)
      XK=X(I,K)
      X(I,J)=XJ+CG*XK
  200 X(I,K)=XK+CA*XJ
  210 CONTINUE
C
C     CALCULA LOS ULTIMOS VALORES CARACTERISTICOS DESPUES DE CADA
C     ROTACION
C
      DO 220 I=1,N
      IF (A(I,I).GT.0. .AND. B(I,I).GT.0.) GO TO 220
      WRITE(IOUT,2020)
      STOP
  220 EIGV(I)=A(I,I)/B(I,I)
      IF(IFPR.EQ.0) GO TO 230
      WRITE(IOUT,2030)
      DO 1234 I=1,N
      WRITE(IOUT,2010) EIGV(I)
 1234 CONTINUE
C
C     PRUEBA LA CONVERGENCIA
C
  230 DO 240 I=1,N
      TOL=RTOL*D(I)
      DIF=ABS(EIGV(I)-D(I))
      IF(DIF.GT.TOL) GO TO 280
  240 CONTINUE
C
```

```
C      CHECA TODOS LOS ELEMENTOS FUERA DE LA DIAGONAL PARA VER SI OTRA
C      ROTACION ES REQUERIDA
C
       EPS=RTOL**2
       DO 250 J=1,NR
       JJ=J+1
       DO 250 K=JJ,N
       EPSA=(A(J,K)*A(J,K))/(A(J,J)*A(K,K))
       EPSB=(B(J,K)*B(J,K))/(B(J,J)*B(K,K))
       IF ((EPSA.LT.EPS).AND.(EPSB.LT.EPS)) GO TO 250
       GO TO 280
   250 CONTINUE
   255 DO 260 I=1,N
       DO 260 J=1,N
       A(J,I)=A(I,J)
   260 B(J,I)=B(I,J)
       DO 270 J=1,N
       BB=SQRT(B(J,J))
       DO 270 K=1,N
   270 X(K,J)=X(K,J)/BB
       RETURN
C
C      OBTIENE LA ULTIMA MATRIZ D Y COMIENZA UNA NUEVA BARRIDA, SI
C      ES REQUERIDA
C
   280 DO 290 I=1,N
   290 D(I)=EIGV(I)
       IF(NSWEEP.LT.NSMAX) GO TO 40
       GO TO 255
  2000 FORMAT("BARRIDA NUMERO EN *JACOBI* =",I4)
  2010 FORMAT(1HO,6E20.12)
  2020 FORMAT("***ERROR EN LA SOLUCION EIN ",/
      *, " MATRIZ NO DEFINIDA POSITIVA")
  2030 FORMAT(" LOS VALORES EN *JACOBI* SON",/)
       END
```

```
@@@@@@@@@@    @@     @@      @@
@@@@@@@@@@    @@     @@@     @@
@@            @@     @@@@    @@
@@            @@     @@@@    @@
@@@@@@        @@     @@ @@@  @@
@@@@@@        @@     @@  @@@ @@
@@            @@     @@   @@@@
@@            @@     @@    @@@
@@            @@     @@     @@
@@            @@     @@      @
```

PROGRAMA PARA CALCULAR LAS FORMAS MODALES Y FRECUENCIAS DE UN SISTEMA ESTRUCTURAL POR EL METODO DE LANCZOS

```
      SUBROUTINE EIGEN(AK1,AM1,ZZ,IREGIO,IPN,IT,I,NECS,NB)
C
C     PROGRAMA
C           EN ESTA SUBRUTINA  SE CALCULAN LAS FRECUENCIAS NATURALES Y
C           LAS FORMAS MODALES DE LA ESTRUCTURA EN CUESTION
C           ESTA SUBRUTINA CONSTA DE UN PROGRAMA PRINCIPAL QUE LLAMA A LAS
C           SUBRUTINAS EN DONDE SE TRANSFORMA EL PROBLEMA DE LA FORMA GE-
C           NERAL A LA FORMA ESTANDAR POR EL METODO DE LANCZOS Y EN DONDE
C           SE CALCULAN LOS VALORES CARACTERISTICOS POR EL METODO DE BI-
C           SECCION, LOS VECTORES CARACTERISTICOS SE CALCULAN POR EL ME-
C           TODO DE ITERACION INVERSA.
C           LAS MATRICES DE RIGIDEZ Y DE MASAS ENTRAN A LA SUBRUTINA DE
C           ACUERDO AL ESQUEMA DE ALMACENAJE NUMERO 1.
C
C     REFERENCIA
C           WILLIAM WEAVER  JR. AND DAVID M. YOSHIDA
C           "THE EIGENVALUE PROBLEM FOR BANDED MATRICES".
C           COMPUTERS & STRUCTURES, VOL. 1, PP. 651-664.
C           PERGAMON PRESS 1971. PRINTED IN GREAT BRITAIN
C                      * * *
C           TIENE MODIFICACIONES HECHAS POR
C             GUSTAVO AYALA MILIAN
C                        Y
C             GUSTAVO ALONDA HERNANDEZ
C           INSTITUTO DE INGENIERIA UNAM.
C
C
C     - - ENTRADA DE VARIABLES - -
C     AK1(NECS,NB)    = MATRIZ DE RIGIDEZ EN BANDA
C     AM1(NECS,IPN)   = MATRIZ DE MASAS EN BANDA
C     NECS            = NUMERO DE ECUACIONES
C     NB              = ANCHO DE BANDA DE LA MATRIZ DE RIGIDEZ
C     IPN             = ANCHO DE BANDA DE LA MATRIZ DE MASAS
C     MI              = VALOR CARACTERISTICO MENOR A ENCONTRAR
C     MF              = VALOR CARACTERISTICO MAYOR A SER ENCONTRADO
C
C     - - SALIDA - -
C     W(MF)           = VECTOR QUE CONTIENE LAS FRECUENCIAS NATURALES
C     ZZ(NECS,MF)     = MATRIZ QUE CONTIENE LAS FORMAS MODALES
C
C
C
      COMMON /TRES/  MI,MF
      COMMON /CUATRO/ W(100)
      DIMENSION R(150,150),B(150,150),D(150),E(150,150),C(150)
     *,B(150),Z(2500),ZI(150,150),ZZ(150,150),AK1(NECS,NB),
     *AM1(NECS,NB)
      WRITE(6,1000)
      CALL DBSOL(NECS,NB,IPN,AK1,AM1,E,C,R,R,NALPHA,R,D)
      IF(NALPHA.EQ.1) GO TO 200
      CALL TRBISE(NECS,C,B)
      WRITE(6,2000) MI,MF
      CALL TRINIT(NECS,Z,C,B)
      WRITE(6,4000)
      DO 400 I=MI,MF
      II=MI+MF-I
      W(I)=SQRT(W(II))
```

```
      WRITE(6,5000) I,W(II)
  400 CONTINUE
      KK=0.
      DO 600 J=1,MF
      DO 700 I=1,NECS
  700 ZK(I,J)=Z(KK*NECS+I)
  600 KK=KK+1
      CALL ASF(NECS,ZK,R,ZZ)
      MM=MF-MI+1
      DO 800 II=1,MM
      I=MM-II+1
      CALL UNIDAD(ZZ,NECS,I)
  800 CONTINUE
      WRITE(6,6000)
      WRITE(6,7000) (I,I=1,MM)
      GO TO 900
      WRITE(6,8000)
      WRITE(6,7000) (I,I=1,MM)
      WRITE(6,9000)
      DO 900 K=1,NECS
      K3=(NNUDP*2+K+1)/2
      IF(MOD(K,2).EQ.0) GO TO 950
      WRITE(6,9500) K3,(ZZ(K,MM-I+1),I=1,MM)
      GO TO 900
  950 WRITE(6,9600) K3,(ZZ(K,MM-I+1),I=1,MM)
  900 CONTINUE
      GO TO 500
  200 CONTINUE
      WRITE(6,9700)
 1000 FORMAT(///25X,"RESULTADOS DEL PROBLEMA DE VALORES CARACTERISTICOS"
     *,/24X,52("*")///)
 2000 FORMAT(10X,"MI =",I5,20X,"MF =",I5///)
 4000 FORMAT(//25X,"FRECUENCIAS NATURALES NORMALIZADAS",//24X,
     *       "RESPECTO A VS/H"/)
 5000 FORMAT(/35X,"W(",I2,")=",F9.4)
 6000 FORMAT(///10X,"VECTOR CARACTERISTICO PARA CADA MODO DE VIBRACION
     *TRANSVERSAL",/9X,63("*")///)
 7000 FORMAT(3X,"MODO",3X,10(I2,3X))
 8000 FORMAT(30X,"VECTORES CARACTERISTICOS",///)
 9000 FORMAT(//)
 9500 FORMAT(/I4,3H  H,3X,10F12.6)
 9600 FORMAT(/I4,3H  V,3X,10F12.6)
 9700 FORMAT(30X,"MATRIZ MALA")
      RETURN
      END
C
      SUBROUTINE HBSG (N,MA,MB,A,B,E,ALFH,BETA,R,NN,W,IP)
C
C     ESTA SUBRUTINA REDUCE EL PROBLEMA DE VALORES CARACTERISTICOS
C     DE LA FORMA NO ESTANDAR (A*X=LAMBDA*B*X) A LA FORMA
C     (T*Y=LAMBDA*Y) DONDE T ES TRIDIAGONAL,EMPLEANDO EL METODO
C     GENERAL DE LANCZOS
C
C     PARAMETROS GLOBALES
C
C     N = ORDEN DE LAS MATRICES BANDEADAS
C     MA = ANCHO DE LA BANDA SUPERIOR DE LA MATRIZ A
C     MB = ANCHO DE LA BANDA SUPERIOR DE LA MATRIZ B
C     A = BANDA SUPERIOR DE LA MATRIZ A
C     B = BANDA SUPERIOR DE LA MATRIZ B
```

```
C       F = MATRIZ DESCOMPUESTA B
C       ALPH = DIAGONAL PRINCIPAL DE T
C       BETA = SUBDIAGONAL DE T
C       R = MATRIZ DE TRANSFORMACION
C       * = SALIDA NO ESTANDAR
C       W,U = VECTORES DE ALMACENAMIENTO TEMPORAL
C
        DIMENSION A(N,MA),B(N,MB)
        REAL F(150,1),R(150,1),W(1),U(1),ALPH(1),BETA(1)
        REAL*8 SUM,TD,DSQRT
C
C       SELECCIONA LA COLUMNA INICIAL DE R,DETERMINA ALPH(1) Y
C       DESCOMPONE B POR EL METODO DE CHOLESKY
C
        NM1=N-1
        R(1,1)=1.0/SQRT(B(1,1))
        DO 100 I=2,N
    100 R(I,1)=0.0
        ALPH(1)=A(1,1)/B(1,1)
        BETA(N)=0.0
        CALL BDCOMP(N,MB,B,F,NN)
        IF(NN.EQ.1) GO TO 1400
        CALL BMULT(N,MA,A,R(1,1),W)
        CALL BMULT(N,MB,B,R(1,1),U)
        DO 200 I=1,N
    200 W(I)=W(I)-ALPH(1)*U(I)
        DO 1300 I=2,N
        IM1=I-1
        CALL BSOL1(N,MB,F,W)
        SUM=0.0
        DO 250 J=1,N
        TD=W(J)
    250 SUM=SUM+TD*TD
        SUM=DSQRT(SUM)
        BETA(IM1)=SUM
        IF(SUM.EQ.0.0) GO TO 400
        DO 300 J=1,N
    300 R(J,I)=W(J)/SUM
        GO TO 600
    400 DO 500 J=1,N
    500 R(J,I)=0.0
        WRITE(6,111)
    111 FORMAT("    PASO")
        R(I,I)=1./SQRT(B(I,I))
    600 CALL BSOL2(N,MB,F,R(1,I))
C
C       EL SIGUIENTE ELEMENTO DE BETA Y LA SIGUIENTE COLUMNA DE R HAN
C       SIDO CALCULADOS. LAS COLUMNAS DE R DEBEN SER REORTOGONALIZADAS
C       PARA MINIMIZAR EL ERROR
C
        CALL BMULT(N,MB,B,R(1,I),W)
        DO 800 J=1,IM1
        SUM=0.
        DO 700 K=1,N
        TD=W(K)
    700 SUM=SUM+TD*R(K,J)
        DO 800 K=1,N
    800 R(K,I)=R(K,I)-SUM*R(K,J)
        CALL BMULT(N,MB,B,R(1,I),W)
        SUM=0
```

```
      DO 900 J=1,N
      TD=R(J,I)
  900 SUM=SUM+W(J)*TD
      SUM=DSQRT(SUM)
      DO 1000 J=1,N
 1000 R(J,I)=R(J,I)/SUM
C
C     CALCULA EL SIGUIENTE ELEMENTO DE ALFA
C
      CALL DMULT(N,MA,A,R(1,I),U)
      SUM=0.
      DO 1100 J=1,N
      TD=R(J,I)
 1100 SUM=SUM+W(J)*TD
      ALFH(I)=SUM
C
C     REINICIALIZA W PARA EL SIGUIENTE PASO
C
      IF(I.EQ.N)RETURN
      CALL DMULT(N,MB,B,R(1,I),U)
      DO 1200 J=1,N
 1200 W(J)=W(J)-ALFH(I)*U(J)
      CALL DMULT(N,MB,B,R(1,IM1),U)
      DO 1300 J=1,N
 1300 W(J)=W(J)-BETA(IM1)*U(J)
 1400 RETURN
      END

      SUBROUTINE TRBISE(N,C,B)

C     TRBISE ENCUENTRA LOS VALORES CARACTERISTICOS DE UNA MATRIZ
C     TRIDIAGONAL SIMETRICA POR EL METODO DE BISECCION
C
      DIMENSION C(150),B(150)
      LOGICAL IFAIL
      COMMON/CUSE/          XMAX,BETA(250),W(150),RELFEH,X1,XU,X0
     *,H,EPS1,EPS2,XMIN,I,L,J,K,IZ,U,NM1
      COMMON /TRES/ M1,M2
      COMMON /CUATRO/ X(100)
      NM1=N-1
      EPS1=1.0E-011
      RELFEH=1.0E-011
      DO 500 J=1,NM1
      I=N-J
  500 B(I+1)=B(I)
      DO 13 I=2,N
   13 BETA(I)=B(I)*B(I)
      BETA(1)=0.
      B(1)=0.
      XMIN=C(N)-ABS(B(N))
      XMAX=C(N)+ABS(B(N))
      DO 100 J=1,NM1
      I=N-J
      H=ABS(B(I))+ABS(B(I+1))
      IF(C(I)+H.GT.XMAX)   XMAX=C(I)+H
  100 IF(C(I)-H.LT.XMIN)  XMIN=C(I)-H
      EPS2=RELFEH*XMIN
      IF(XMAX+XMIN  )1,1,2
    1 EPS2=-RELFEH*XMAX
    2 IF(EPS1) 3,3,4
```

```
    3 EPS1=EPS2
    4 EPS2=0.5*EPS1+7.0*EPS2
      XO=XMAX
      DO 200 I=M1,M2
      X(I)=XMAX
  200 WU(I)=XMIN
      IZ=0
      DO 300 J=M1,M2
      K=M2+M1-J
      XU=XMIN
      DO 400 JJ=M1,K
      I=M1+K-JJ
      IF(XU-WU(I)) 5,400,400
    5 XU=WU(I)
      GO TO 6
  400 CONTINUE
    6 IF(XO.GT.X(K)) XO=X(K)
    7 AST=2.*RELFEH*(ABS(XU)+ABS(XO))+EPS)-XO+XU
      IF(AST) 8,300,300
    8 X1=(XU+XO)/2.
      CALL STUSEQ(N,C,B)
      GO TO 7
  300 X(K)=(XO+XU)/2.
      DO 600 I=M1,M2
  600 WU(I)=X(I)
      DO 700 I=M1,M2
  700 Y(I)=WU(M1+M2-I)
      DO 800 I=1,NM1
  800 B(I)=B(I+1)
      B(N)=0.
      RETURN
      END
C
      SUBROUTINE TRINIT(N,Z,C,B)
C
C     CALCULO DE LOS VECTORES CARACTERISTICOS DE UNA MATRIZ TRI-
C     DIAGONAL, A PARTIR DE LOS VALORES CARACTERISTICOS CONOCIDOS
C
C     C ES LA DIAGONAL Y B LA SUBDIAGONAL DE UNA MATRIZ TRIDIAGONAL
C     SIMETRICA. EL VECTOR W DE ORDEN M1 CONTIENE LOS VALORES CARACTE
C     RISTICOS EN ORDEN DECRECIENTE. GORM ES LA NORMA INFINITA DE LA
C     MATRIZ,FAM ES LA PRECISION RELATIVA DE LA COMPUTADORA. LOS M1
C     VECTORES CARACTERISTICOS SE CALCULAN POR ITERACION INVERSA Y SE
C     ALMACENAN EN LA MATRIZ Z. Z(I,J) ES EL I ESIMO COMPONENTE DEL
C     VECTOR CARACTERISTICO CORRESPONDIENTE AL VALOR CARACTERISTICO
C     W(J)
C
      DIMENSION Z(1),C(1),B(1)
      COMMON/CUSE/            GORM,P(250),AM[illegible](250),[illegible](250),FAM
     *TIN(250),X(252),SLAM,EPS,U,V,B1,B11,DELTA,N1,I,J,I1
      COMMON /TRES/M3,M2
      COMMON /CUATRO/ 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)
```

```
      U=C(1)-SLAM
      V=B(1)
      IF(V.EQ.0.) V=EPS
C
C     PARA ASEGURAR LA INDEPENDENCIA EN CASOS DEGENERADOS
C
      DO 2 I=1,N1
      BI=B(I)
      IF(BI.EQ.0.) BI=EPS
      BI1=B(I+1)
      IF(BI1.EQ.0.) BI1=EPS
      IF(ABS(BI).LE.ABS(U)) GO TO 4
    3 AM(I+1)=U/BI
      IF((AM(I+1).EQ.0.).AND.(BI.LE.EPS)) AM(I+1)=1.
C
C     PARA EVITAR EL FACTOR CERO EN EL CASO DESCOMPUESTO
C
      P(I)=BI
      Q(I)=C(I+1)-SLAM
      R(I)=BI1
      U=V-AM(I+1)*Q(I)
      V=-AM(I+1)*R(I)
      TIN(I+1)=1.
      GO TO 2
    4 AM(I+1)=BI/U
      P(I)=U
      Q(I)=V
      R(I)=0.
      U=C(I+1)-SLAM-AM(I+1)*V
      V=BI1
      TIN(I+1)=-1.
    2 CONTINUE
      P(N)=U
      Q(N)=0.
      R(N)=0.
      X(N+1)=0.
      X(N+2)=0.
      H=0.
      DO 5 II=1,N
      I=N+1-II
      U=ETA-Q(I)*X(I+1)-R(I)*X(I+2)
      IF(P(I).EQ.0.) X(I)=U/EPS
      IF(P(I).NE.0.) X(I)=U/P(I)
      H=H+ABS(X(I))
    5 CONTINUE
      H=1./H
      DO 6 I=1,N
    6 X(I)=X(I)*H
      DO 7 I=2,N
      IF(TIN(I).LE.0.) GO TO 8
      U=X(I-1)
      X(I-1)=X(I)
      X(I)=U-AM(I)*X(I-1)
      GO TO 7
    8 X(I)=X(I)-AM(I)*X(I-1)
    7 CONTINUE
      H=0.
      DO 9 II=1,N
      I=N+1-II
      U=X(I)-Q(I)*X(I+1)-R(I)*X(I+2)
```

```
      IF(P(I).EQ.0.) X(I)=U/EPS
      IF(P(I).NE.0.) X(I)=U/P(I)
      H=H+X(I)*X(I)
    9 CONTINUE
      H=1./SQRT(H)
      DO 100 I=1,N
  100 X(I)=X(I)*H
      H=0.
      DO 101 II=1,N
      I=N+1-II
      U=X(I)-Q(I)*X(I+1)-R(I)*X(I+2)
      IF(P(I).EQ.0.) X(I)=U/EPS
      IF(P(I).NE.0.) X(I)=U/P(I)
      H=H+X(I)*X(I)
  101 CONTINUE
      H=1/SQRT(H)
      DO 11 I=1,N
   11 Z(I+(J-1)*N)=X(I)*H
    1 CONTINUE
      RETURN
      END
C
      SUBROUTINE ASF(N,ZK,R,ZZ)
C
      COMMON /TRES/M1,M2
      DIMENSION ZK(150,1),R(150,1),ZZ(150,1)
      DO 2 I=1,N
      DO 2 J=1,M2
      ZZ(I,J)=0.
      DO 1 K=1,N
    1 ZZ(I,J)=ZZ(I,J)+R(I,K)*ZK(K,J)
    2 CONTINUE
      RETURN
      END
C
      SUBROUTINE UNIDAD(ZZ,N,II)
C
C     NORMALIZA LAS AMPLITUDES DEL VECTOR MODAL CON RESPECTO AL VALOR
C     MAXIMO ENCONTRADO EN DICHO VECTOR
C
      DIMENSION ZZ(150,150)
      L=1
      AMAY=ABS(ZZ(1,II))
      DO 1 K=2,N
      IF(AMAY.GE.ABS(ZZ(K,II))) GO TO 1
      L=K
      AMAY=ABS(ZZ(K,II))
    1 CONTINUE
      R=1.0/ZZ(L,II)
      DO 2 K=1,N
    2 ZZ(K,II)=ZZ(K,II)*R
      RETURN
      END
C
      SUBROUTINE DECOMP(N,IBW,A,U,NALFA)
C
C     ESTA SUBRUTINA DESCOMPONE UNA MATRIZ SIMETRICA POSITIVA DEFINIDA A
C     LA FORMA (A=UTU), DONDE U ES UNA MATRIZ DE BANDA SUPERIOR
C
      DIMENSION A(N,IBW)
```

```
      REAL U(150,1)
      REAL*8 SUM,DSQRT,UD
      INTEGER P,Q
      DO 400 I=1,N
      P=N-I+1
      IF(IBW.LT.P) P=IBW
      DO 400 J=1,P
      Q=IBW-J
      IF(I-1.LT.Q) Q=I-1
      SUM=A(I,J)
      IF(Q.LT.1) GO TO 200
      DO 100 K=1,Q
      UD=U(I-K,J+K)
  100 SUM=SUM-U(I-K,1+K)*UD
  200 IF(J.NE.1) GO TO 300
      IF(SUM.GT.0.) GO TO 201
      NALFA=1
      RETURN
  201 TEMP=1.0/DSQRT(SUM)
      U(I,J)=TEMP
      GO TO 400
  300 U(I,J)=SUM*TEMP
  400 CONTINUE
      RETURN
      END
C
      SUBROUTINE DMULT(N,IBW,A,X,B)
C
C     ESTA SUBRUTINA MULTIPLICA UNA MATRIZ SIMETRICA BANDEADA POR UN
C     VECTOR. SOLO SE ALMACENA LA PORCION DE LA BANDA SUPERIOR DE LA
C     MATRIZ
C
      DIMENSION A(N,IBW)
      REAL X(1),B(1)
      REAL*8 SUM,XD
      N1=N-IBW+1
      DO 300 I=1,N
      SUM=0.
      J1=IBW
      IF(I.GT.N1) J1=N-I+1
      DO 100 J=1,J1
      XD=X(I+J-1)
  100 SUM=SUM+A(I,J)*XD
      IF((I.EQ.1).OR.(IBW.EQ.1)) GO TO 300
      J1=I-IBW+1
      J2=J1+IBW-2
      IF(I.GT.IBW) GO TO 120
      J1=1
      J2=I-1
  120 DO 200 J=J1,J2
      XD=X(J)
  200 SUM=SUM+A(J,I-J+1)*XD
  300 B(I)=SUM
      RETURN
      END
C
      SUBROUTINE BSOL1(N,IBW,U,B)
C
C     EN ESTA SUBRUTINA SE RESUELVE UN CONJUNTO DE ECUACIONES
C     ALGEBRAICAS SIMULTANEAS DE LA FORMA (UD) (B), DONDE U
```

```
C     ES UNA MATRIZ TRIANGULAR INFERIOR BANDEADA
C
      REAL U(150,1),B(1)
      REAL*8 SUM,BD
      DO 200 I=1,N
      J=I-IBW+1
      IF(I+1.LE.IBW) J=1
      SUM=B(I)
      K1=I-1
      IF(J.GT.K1) GO TO 200
      DO 100 K=J,K1
      BD=B(K)
  100 SUM=SUM-U(K,I-K+1)*BD
  200 B(I)=SUM*U(I,1)
      RETURN
      END
C
      SUBROUTINE DSOL2(N,IBW,U,X)
C
C     EN ESTA SUBRUTINA SE RESUELVE UN CONJUNTO DE ECUACIONES
C     ALGEBRAICAS SIMULTANEAS DE LA FORMA ( U*Y=X ), DONDE U ES
C     UNA MATRIZ TRIANGULAR SUPERIOR BANDEADA
C
      REAL U(150,1),X(1)
      REAL*8 SUM,XD
      DO 200 I1=1,N
      I=N-I1+1
      J=I+IBW-1
      IF(J.GT.N) J=N
      SUM=X(I)
      K2=I+1
      IF(K2.GT.J) GO TO 200
      DO 100 K=K2,J
      XD=X(K)
  100 SUM=SUM-U(I,K-I+1)*XD
  200 X(I)=SUM*U(I,1)
      RETURN
      END
C
      SUBROUTINE STURM(N,C,B)
C
C     AQUI SE CUENTA EL NUMERO DE CAMBIOS EN EL SIGNO DE LOS
C     MENORES SUCESIVOS
C
      DIMENSION C(1),B(1)
      COMMON/CUSE/           XMAX,BETA(150),UU(150),RELFEH,X1,ML,XO
     *,Q,EPS1,EPS2,XMIN,L,I,J,K,IZ,KI,NM1
      COMMON /TRES/M1,M
      COMMON /CUATRO/ Y(100)
      IZ=IZ+1
      IA=0
      Q=1
      DO 100 I=1,N
      IF(Q) 1,2,1
    2 Q=C(I)-X1-ABS(B(I))/RELFEH
      GO TO 3
    1 Q=C(I)-X1-BETA(I)/Q
    3 IF(Q) 4,100,100
    4 IA=IA+1
  100 CONTINUE
```

```
      IF(IA-K) 6,5,5
    5 XO=X1
      RETURN
    6 XU=X1
      IF(IA-M1) 7,8,8
    7 WU(M1)=X1
      RETURN
    8 WU(IA+1)=X1
      IF(X(IA).GT.X1) X(IA)=X1
      RETURN
      END
C
```

```
@@@@@@@@@@      @@      @@       @@
@@@@@@@@@@      @@      @@@      @@
@@              @@      @@@@     @@
@@              @@      @@@@     @@
@@@@@@          @@      @@ @@@   @@
@@@@@@          @@      @@  @@@  @@
@@              @@      @@    @@@@
@@              @@      @@    @@@@
@@              @@      @@     @@@
@@              @@      @@      @@
```

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

```
      SUBROUTINE BISECC(A,B,D,DL,E,EIGEN,EL,E2,Z,W,T,N,KI,M,M2,TOL
     *,DLAM,EPS,CHEPS)
C
C       PROGRAMA
C         RESUELVE EL PROBLEMA DE VALORES CARACTERISTICOS PARA MATRICES
C         SIMETRICAS, PARA EL NUMERO DE VALORES REQUERIDOS.
C         EN ESTE PROGRAMA SE REDUCE EL PROBLEMA DE LA FORMA GENERAL A
C         LA FORMA ESTANDAR, LA MATRIZ DE RIGIDEZ SE TRIDIAGONALIZA POR
C         EL METODO DE HOUSEHOLDER, LOS VALORES CARACTERISTICOS SE OBTIE-
C         NEN POR EL METODO DE BISECCION, LOS VECTORES CARACTERISTICOS
C         SE OBTIENEN POR EL METODO DE ITERACION INVERSA.
C
C     REFERENCIA
C        1-G.M.L. GLADWELL AND U.C. TAHBILDAR
C          "THE ALGEBRAIC EIGENVALUE PROBLEM
C          -A SET OF FORTRAN SUBROUTINES",
C          SOLID MECHANICS DIVISION UNIVERSITY OF WATERLOO.
C          REPORT NO. 18 MARCH, 1972
C
C        2-G.M.L. GLADWELL
C          "THE ALGEBRAIC EIGENVALUE PROBLEM)
C          A COMMENTARY ON SOME COMPUTERS PROGRAMS",
C          SOLID MECHANICS DIVISION UNIVERSITY OF WATERLOO.
C          REPORT NO. 25 MAY, 1973
C
C       LA LLAMADA A LA SUBROUTINA  ES HECHA POR.
C       CALL BISECC(A,B,N,KI,M,M2,TOL,CHEPS)
C
C       - - ENTRADA DE VARIABLES - -
C       A(N,N)   = MATRIZ DE RIGIDEZ DE LA ESTRUCTURA
C       B(N,N)   = MATRIZ DE MASAS DE LA ESTRUCTURA.
C       KI=0 SI A ES POSITIVA DEFINIDA. =1 SI NO LO ES.
C       M        = NO. DE VALORES CARACTERISTICOS REQUERIDOS.
C       M2       = NO. DE VECT. CARACTERISTICOS REQUERIDOS.
C       TOL Y DLAM PUEDEN SER  CERO
C       CHEPS APARECE EN EIG.
C
C      - - SALIDA - -
C      Z(N,N)    = MATRIZ QUE CONTIENE LOS VECT.CARACTERISTICOS
C      EIGEN(N) = VECTOR QUE CONTIENE LOS VALORES CARACTERISTICOS
C
C
      IMPLICIT REAL*8(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)
C
      CALL REDUC(A,B,DL,N,NX)
C
      CALL TRID(A,D,E,E2,TOL,TNORM,N,NX)
C
      CALL BISE(D,E2,E,EIGEN,EPS,CHEPS,M,N)
C
      CALL EIG(D,E,EIGEN,Z,CHEPS,TNORM,M2,N,NX)
C
      CALL BACK (A,E,Z,M2,N,NX)
C
      CALL TRANS(B,DL,Z,M2,N,NX)
```

```
      RETURN
      END
      SUBROUTINE BISE(D,B2,EE,EV,RELF,EPS1,MM,N)
C
C     CALCULA LOS VALORES CARACTERISTICOS.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION EE(N),B2(N),D(N),EV(N)
      DIMENSION WU(200),X(200)
      EE(1)=0
      XMIN=D(N)-DABS(EE(N))
      XMAX=D(N)+DABS(EE(N))
      IJ=N-1
      I=N
      DO 1 II=1,IJ
      I=I-1
      H=DABS(EE(I))+DABS(EE(I+1))
      IF((D(I)+H).GT.XMAX)XMAX=D(I)+H
      IF((D(I)-H).LT.XMIN)XMIN=D(I)-H
    1 CONTINUE
      XM=XMIN+XMAX
      IF(XM.GT.0)EPS2=RELF*XMAX
      IF(XM.LE.0)EPS2=-RELF*XMIN
      IF(EPS1.LE.0)EPS1=EPS2
      EPS2=0.5*EPS1+7*EPS2
      XO=XMAX
      DO 2 I=1,MM
      X(I)=XMAX
      WU(I)=XMIN
    2 CONTINUE
      K=MM+1
      DO  3 KK=1,MM
      K=K-1
      XU=XMIN
      I=K+1
      DO 4 II=1,K
      I=I-1
      IF(XU.GE.WU(I))GO TO 4
      XU=WU(I)
      GO TO 5
    4 CONTINUE
    5 IF(XO.GT.X(K))XO=X(K)
      XX=XO-XU
      XXX=2*RELF*(DABS(XU)+DABS(XO))+EPS1
      XY=XX-XXX
      IF(XY)16,16,7
    7 X1=(XU+XO)/2
      IA=0
      Q=1.0
      DO 8 I=1,N
      IF(Q.NE.0)GO TO 14
      IF(Q.EQ.0)Q=D(I)-X1-DABS(EE(I)/RELF)
      GO TO 15
   14 Q=D(I)-X1-B2(I)/Q
   15 CONTINUE
      IF(Q.LT.0)IA=IA+1
    8 CONTINUE
      IF(IA.GE.K)GO TO 9
      IF(IA.GE.1)GO TO 11
      XU=X1
```

```
      WU(I)=XI
      GO TO  1212
   11 WU(IA+1)=XI
      XU=WU(IA+1)
      IF(X(IA).GT.XI)X(IA)=XI
 1212 K=K
      GO TO 10
    9 XO=XI
   10 XX=XO-XU
      XXX=2*RELF*(DABS(XU)+DABS(XO))+EPSI
      XY=XX-XXX
C   *
      IF(XY)16,16,7
   16 X(K)=(XO+XU)/2
    3 CONTINUE
      DO 13 J=1,MM
      EV(J)=X(J)
   13 CONTINUE
      RETURN
      END
      SUBROUTINE REDUC(A,B,DL,N,NX)
C
C     REDUCE EL PROBLEMA A LA FORMA GENERAL.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(NX,NX),B(NX,NX),DL(NX)
      IF(N)1,2,2
    1 N=-N
      GO TO  3
    2 DO 8 I=1,N
      DO 8 J=I,N
      X=B(I,J)
      WRITE(6,*/) X
      JJ=I-1
      K=I
      IF(JJ.EQ.0)GO TO 16
      DO 6 KK=1,JJ
      K=K-1
    6 X=X-B(I,K)*B(J,K)
   16 K=K
      IF(I.NE.J)GO TO 7
      IF(X.LE.0.0)GO TO 999
      Y=DSQRT(X)
      DL(I)=DSQRT(X)
      GO TO 8
    7 B(J,I)=X/Y
    8 CONTINUE
    3 DO 9 I=1,N
      II=I-1
      Y=DL(I)
      DO 10 J=1,N
      X=A(I,J)
      K=I
      IF(II.EQ.0)GO TO 10
      DO 11 KK=1,II
      K=K-1
   11 X=X-B(I,K)*A(J,K)
   10 A(J,I)=X/Y
    9 CONTINUE
      DO  12 J=1,N
```

```
      DO  12 I=J,N
      II=I-1
      JJ=J-1
      K=I
      X=A(I,J)
      IF(II.LT.J)GO TO 17
      DO 13 KK=J,II
      K=K-1
   13 X=X-A(K,J)*B(I,K)
   17 K=J
      IF(JJ.EQ.0)GO TO 18
      DO 14 KK=1,JJ
      K=K-1
   14 X=X-A(J,K)*B(I,K)
   18 A(I,J)=X/DL(I)
   12 CONTINUE
      GO TO 888
  999 PRINT 15
   15 FORMAT(2X,"LA MATRIZ NO ES DEFINIDA POSITIVA")
  888 K=K
      RETURN
      END
      SUBROUTINE TRID(A,D,E,E2,TOL,INORM,N,NX)
C
C     TRIDIAGONALIZA LA MATRIZ.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(NX,NX),D(NX),E(NX),E2(NX)
      DO 1 I=1,N
    1 D(I)=A(I,I)
      I=N+1
      DO 2 II=1,N
      I=I-1
      L=I-1
      H=0
      IF(L.EQ.0)GO TO 14
      DO  1212 K=1,L
 1212 H=H+A(I,K)*A(I,K)
   14 IF(H.GT.TOL)GO TO 4
      E(I)=0
      E2(I)=0
      GO TO 5
    4 E2(I)=H
      F=A(I,I-1)
      IF(F.LT.0.0)GO TO 6
      E(I)=-DSQRT(H)
      G=-DSQRT(H)
      GO TO 7
    6 E(I)=DSQRT(H)
      G=DSQRT(H)
    7 H=H-F*G
      A(I,I-1)=F-G
      F=0
      IF(L.EQ.0)GO TO 15
      DO 8 J=1,L
      G=0
      DO 9 K=1,J
    9 G=G+A(J,K)*A(I,K)
      JJ=J+1
      IF(JJ.GT.L)GO TO 16
```

```fortran
      DO 10 K=J,L
   10 G=G+A(K,J)*A(I,K)
   16 E(J)=G/H
      G=E(J)
    8 F=F+G*A(I,J)
   15 H=F/(H+H)
      IF(L.EQ.0)GO TO 5
      DO 11 J=1,L
      F=A(I,J)
      E(J)=E(J)-H*F
      G=E(J)
      DO  12 K=1,J
   12  A(J,K)=A(J,K)-F*E(K)-G*A(I,K)
   11 CONTINUE
    5 H=D(I)
      D(I)=A(I,I)
      A(I,I)=H
    2 CONTINUE
      TNORM=DABS(D(1))+DABS(E(1))
      NI=N-1
      DO 102 I=1,NI
      AL=DABS(E(I))+DABS(E(I+1))+DABS(D(I))
  102 IF(AL.GT.TNORM) TNORM=AL
      RETURN
      END
      SUBROUTINE EIG(C,EE,W,Z,CHEPS,NORM,M1,N,NX)
C
C     CALCULA LOS VECTORES CARACTERISTICOS DE LA MATRIZ TRIDIAGONAL.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION C(NX),W(NX),EE(NX),Z(NX,NX)
      DIMENSION P(200),Q(200),R(200),X(200),INT(200)
      REAL*8 LAMDA,NORM,M(200)
      II=N-1
      DO 23 I=1,II
   23 EE(I)=EE(I+1)
      EE(N)=0.
      LAMDA=NORM
      EPS=CHEPS*NORM
      J=M1+1
      DO 1 JJ=1,M1
      J=J-1
      LAMDA=LAMDA-EPS
      IF(W(J).LT.LAMDA)LAMDA=W(J)
      U=C(1)-LAMDA
      V=EE(1)
      IF(V.EQ.0)V=EPS
      KK=N-1
      DO 2 I=1,KK
      II=I+1
      BI=EE(I)
      IF(BI.EQ.0)BI=EPS
      BI1=EE(I+1)
      IF(BI1.EQ.0)BI1=EPS
      IF(DABS(BI).LT.DABS(U))GO TO  1212
      M(I+1)=U/BI
      IF(M(I+1).NE.0)GO TO 4
      IF(BI.LE.EPS)M(I+1)=1.0
    4 P(I)=BI
      Q(I)=C(I+1)-LAMDA
```

```
      R(I)=BI1
      U=V-M(I+1)*Q(I)
      V=-M(I+1)*R(I)
      INT(I+1)=+1
      GO TO 2
 1212 M(I+1)=BI/U
      P(I)=U
      Q(I)=V
      R(I)=0
      U=C(I+1)-LAMDA-M(I+1)*V
      V=BI1
      INT(I+1)=-1
    2 CONTINUE
      P(N)=U
      Q(N)=0
      R(N)=0
      X(N+1)=0
      X(N+2)=0
      H=0
      ETA=1.0/N
      I=N+1
      DO 5 II=1,N
      I=I-1
      U=ETA-Q(I)*X(I+1)-R(I)*X(I+2)
      IF(P(I).EQ.0)X(I)=U/EPS
      IF(P(I).NE.0)X(I)=U/P(I)
    5 H=H+DABS(X(I))
      H=1/H
      DO 6 I=1,N
    6 X(I)=X(I)*H
      DO 7 I=2,N
      IF(INT(I).LE.0)GO TO 8
      U=X(I-1)
      X(I-1)=X(I)
      X(I)=U-M(I)*X(I-1)
      GO TO 7
    8 X(I)=X(I)-M(I)*X(I-1)
    7 CONTINUE
      H=0
      I=N+1
      DO 9 II=1,N
      I=I-1
      U=X(I)-Q(I)*X(I+1)-R(I)*X(I+2)
      IF(P(I).EQ.0)X(I)=U/EPS
      IF(P(I).NE.0)X(I)=U/P(I)
    9 H=H+X(I)*X(I)
      H=1.D0/DSQRT(H)
      DO 10 I=1,N
   10 Z(I,J)=X(I)*H
    1 CONTINUE
C
C     CAMBIA LOS INDICES DE LA SUBRUTINA NAC
C
      NN=N-1
      I=N
      DO 22 II=1,NN
      I=I-1
   22 EE(I+1)=EE(I)
      EE(1)=0.
      RETURN
```

```
      END
      SUBROUTINE BACK(A,EE,Z,M1,N,NX)
C
C     CALCULA LOS VECTORES CARACTERISTICOS DE LA MATRIZ REDUCIDA.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Z(NX,NX),EE(NX),A(NX,NX)
      DO 1 I=2,N
      IF(EE(I).EQ.0.0)GO TO 1
      L=I-1
      H=EE(I)*A(I,I-1)
      DO 2 J=1,M1
      S=0.0
      DO  1212 K=1,L
 1212 S=S+A(I,K)*Z(K,J)
      S=S/H
      DO 4 K=1,L
    4 Z(K,J)=Z(K,J)+S*A(I,L)
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
      SUBROUTINE TRANS(B,BL,Z,M1,N,NX)
C
C     CALCULA LOS VECTORES CARACT. DEL PROBLEMA ORIGINAL.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Z(NX,NX),BL(NX),B(NX,NX)
      DO 1 J=1,M1
      I=N+1
      DO 1 II=1,N
      I=I-1
      X=Z(I,J)
      III=I+1
      IF(III.GT.N)GO TO  1212
      DO 2 K=III,N
    2 X=X-B(K,I)*Z(K,J)
 1212 JJ=1
    1 Z(I,J)=X/BL(I)
      DO 5 J=1,M1
      H=DABS(Z(1,J))
      DO 28 I=2,N
   28 IF(DABS(Z(I,J)).GT.DABS(H))H=DABS(Z(I,J))
      DO 29 I=1,N
   29 Z(I,J)=Z(I,J)/H
    5 CONTINUE
      RETURN
      END
```

```
@@@@@@@@@@    @@    @@      @@
@@@@@@@@@@    @@    @@@     @@
@@            @@    @@@@    @@
@@            @@    @@@@    @@
@@@@@@        @@    @@ @@@  @@
@@@@@@        @@    @@  @@@ @@
@@            @@    @@    @@@@
@@            @@    @@    @@@@
@@            @@    @@     @@@
@@            @@    @@      @@
```

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

```
      SUBROUTINE RATOR(A,B,D,DL,E,EIGEN,EL,E2,Z,W,T,N,KI,M,M2,TOL
     *,DLAM,EPS,CHEPS)
C
C      PROGRAMA
C        RESUELVE EL PROBLEMA DE VALORES CARACTERISTICOS PARA MATRICES
C        SIMETRICAS, PARA EL NUMERO DE VALORES REQUERIDOS.
C        EN ESTE PROGRAMA SE REDUCE EL PROBLEMA DE LA FORMA GENERAL A
C        LA FORMA ESTANDAR. LA MATRIZ DE RIGIDEZ SE TRIDIAGONALIZA POR
C        EL METODO DE HOUSEHOLDER. LOS VALORES CARACTERISTICOS SE OBTIE-
C        NEN POR EL METODO QR. LOS VECTORES CARACTERISTICOS SE OBTIENEN
C        POR EL METODO DE ITERACION INVERSA.
C
C     REFERENCIA
C       1-G.M.L. GLADWELL AND U.C. TAHBILDAR
C         "THE ALGEBRAIC EIGENVALUE PROBLEM
C         -A SET OF FORTRAN SUBROUTINES",
C         SOLID MECHANICS DIVISION UNIVERSITY OF WATERLOO.
C         REPORT NO. 18 MARCH, 1972
C
C       2-G.M.L. GLADWELL
C         "THE ALGEBRAIC EIGENVALUE PROBLEM:
C         A COMMENTARY ON SOME COMPUTERS PROGRAMS",
C         SOLID MECHANICS DIVISION UNIVERSITY OF WATERLOO.
C         REPORT NO. 25 MAY, 1973
C
C
C      LA LLAMADA A LA SUBROUTINA  ES HECHA POR.
C      CALL RATOR(A,B,N,KI,M,M2,TOL,CHEPS,EPS)
C
C      - - ENTRADA DE VARIABLES -
C      A(N,N)  = MATRIZ DE RIGIDEZ DE LA ESTRUCTURA
C      B(N,N)  = MATRIZ DE MASAS DE LA ESTRUCTURA.
C      KI=0 SI A ES POSITIVA DEFINIDA, =1 SI NO LO ES
C      M       = NO. DE VALORES CARACTERISTICOS REQUERIDOS.
C      M2      = NO. DE VECT. CARACTERISTICOS REQUERIDOS.
C      TOL Y DLAM PUEDEN SER  CERO
C      CHEPS APARECE EN EIG.
C
C     - - SALIDA - -
C     Z(N,N)   = MATRIZ QUE CONTIENE LOS VECT.CARACTERISTICOS
C     EIGEN(N) = VECTOR QUE CONTIENE LOS VALORES CARACTERISTICOS
C
C
      IMPLICIT REAL*8(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)
C
      CALL REDUC(A,B,DL,N,NX)
C
      CALL TRID(A,D,E,E2,TOL,TNORM,N,NX)
C
      CALL BISE(D,E2,L,EIGEN,EPS,CHEPS,M,N)
C
      CALL EIG(D,E,EIGEN,Z,CHEPS,TNORM,M2,N,NX)
C
      CALL BACK (A,E,Z,M2,N,NX)
C
```

```
      CALL TRANS(B,DL,Z,M2 N,NX)
      RETURN
      END
      SUBROUTINE REDUC(A,B,DL,N,NX)
C
C     REDUCE EL PROBLEMA A LA FORMA GENERAL.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(NX,NX),B(NX,NX),DL(NX)
      IF(N)1,2,2
    1 N=-N
      GO TO 3
    2 DO 8 I=1,N
      DO 8 J=I,N
      X=B(I,J)
      JJ=I-1
      K=I
      IF(JJ.EQ.0)GO TO 16
      DO 6 KK=1,JJ
      K=K-1
    6 X=X-B(I,K)*B(J,K)
   16 K=K
      IF(I.NE.J)GO TO 7
      IF(X.LE.0.0)GO TO 999
      Y=DSQRT(X)
      DL(I)=DSQRT(X)
      GO TO 8
    7 B(J,I)=X/Y
    8 CONTINUE
    3 DO 9 I=1,N
      II=I-1
      Y=DL(I)
      DO 10 J=I,N
      X=A(I,J)
      K=I
      IF(II.EQ.0)GO TO 10
      DO 11 KK=1,II
      K=K-1
   11 X=X-B(I,K)*A(J,K)
   10 A(J,I)=X/Y
    9 CONTINUE
      DO 12 J=1,N
      DO 12 I=J,N
      II=I-1
      JJ=J-1
      K=I
      X=A(I,J)
      IF(II.LT.J)GO TO 17
      DO 13 KK=J,II
      K=K-1
   13 X=X-A(K,J)*B(I,K)
   17 K=J
      IF(JJ.EQ.0)GO TO 18
      DO 14 KK=1,JJ
      K=K-1
   14 X=X-A(J,K)*B(I,K)
   18 A(I,J)=X/DL(I)
   12 CONTINUE
      GO TO 888
  999 PRINT 15
```

```
   15 FORMAT(2X,'LA MATRIZ NO ES DEFINIDA POSITIVA')
  888 K=K
      RETURN
      END
      SUBROUTINE TRID(A,D,E,E2,TOL,TNORM,N,NX)
C
C     TRIDIAGONALIZA LA MATRIZ.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(NX,NX),D(NX),E(NX),E2(NX)
      DO 1 I=1,N
    1 D(I)=A(I,I)
      I=N+1
      DO 2 II=1,N
      I=I-1
      L=I-1
      H=0
      IF(L.EQ.0)GO TO 14
      DO 3 K=1,L
    3 H=H+A(I,K)*A(I,K)
   14 IF(H.GT.TOL)GO TO 4
      E(I)=0
      E2(I)=0
      GO TO 5
    4 E2(I)=H
      F=A(I,I-1)
      IF(F.LT.0.0)GO TO 6
      E(I)=-DSQRT(H)
      G=-DSQRT(H)
      GO TO 7
    6 E(I)=DSQRT(H)
      G=DSQRT(H)
    7 H=H-F*G
      A(I,I-1)=F-G
      F=0
      IF(L.EQ.0)GO TO 15
      DO 8 J=1,L
      G=0
      DO 9 K=1,J
    9 G=G+A(J,K)*A(I,K)
      JJ=J+1
      IF(JJ.GT.L)GO TO 16
      DO 10 K=JJ,L
   10 G=G+A(K,J)*A(I,K)
   16 E(J)=G/H
      G=E(J)
    8 F=F+G*A(I,J)
   15 H=F/(H+H)
      IF(L.EQ.0)GO TO 5
      DO 11 J=1,L
      F=A(I,J)
      E(J)=E(J)-H*F
      G=E(J)
      DO 12 K=1,J
   12 A(J,K)=A(J,K)-F*E(K)-G*A(I,K)
   11 CONTINUE
    5 H=D(I)
      D(I)=A(I,I)
      A(I,I)=H
    2 CONTINUE
```

```
      TNORM=DABS(D(1))+DABS(E(1))
      NI=N-1
      DO 102 I=1,NI
      AL =DABS(E(I))+DABS(E(I+1))+DABS(D(I))
  102 IF(AL.GT.TNORM) TNORM=AL
      RETURN
      END
      SUBROUTINE RATOR(D,B2,EIGEN,DLAM,EPS,LL,M,N)
C
C     CALCULA LOS VALORES CARACTERISTICOS.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION B2(N),D(N),EIGEN(N)
      DO 100 I=1,N
  100 EIGEN(I)=D(I)
      B2(1)=0
      ERR=0
      Q=0
      S=0
      TOT=D(1)
      I=N+1
      DO 1 II=1,N
      I=I-1
      P=0
      Q=DSQRT(B2(I))
      E=D(I)-P-Q
    1 IF(E.LT.TOT)TOT=E
      IF(KI.NE.0)GO TO 3
      IF(TOT)2,3,3
    2 TOT=0
      GO TO 4
    3 DO 5 I=1,N
    5 D(I)=D(I)-TOT
    4 DO 6 K=1,M
  777 TOT=TOT+S
      DELTA=D(N)-S
      I=N
      E=DABS(EPS*TOT)
      IF(DLAM.LT.E)DLAM=E
      IF(DELTA.LE.DLAM)GO TO 888
      E=B2(N)/DELTA
      QP=DELTA+E
      P=1.D0
      I=N
      LL=N-1
      IF(K.GT.LL) GO TO 999
      DO 7 II=K,LL
      I=I-1
      Q=D(I)-S-E
      R=Q/QP
      P=P*R+1
      EP=E*R
      D(I+1)=QP+EP
      DELTA=Q-EP
      LLJ=I+1
      IF(DELTA.LE.DLAM) GO TO 888
      IF(B2(I).LT.1.D-30)B2(I)=0.D0
      E=B2(I)/Q
      QP=DELTA+E
      B2(I+1)=QP*EP
```

```
    7 CONTINUE
  999 CONTINUE
      D(K)=QP
      S=QP/P
      ST=TOT+S
      IF(ST.GT.TOT)GO TO 777
      S=0
      I=K
      DELTA=QP
      KK=K+1
      IF(KK.GT.N) GO TO 888
      DO 8 J=KK,N
      IF(D(J).LT.DELTA)I=J
    8 IF(D(J).LT.DELTA)DELTA=D(J)
  888 IF(I.LT.N)B2(I+1)=B2(I)*E/QP
      II=I-1
      J=I
      IF(II.LT.K)GO TO 11
      DO 9 JJ=K,II
      J=J-1
      D(J+1)=D(J)-S
    9 B2(J+1)=B2(J)
   11 D(K)=TOT
      ERR=ERR+DABS(DELTA)
    6 B2(K)=ERR
C
C     INTERCAMBIA D Y EIGEN
C
      DO 101 I=1,N
      TEM=D(I)
      D(I)=EIGEN(I)
  101 EIGEN(I)= TEM
      RETURN
      END
      SUBROUTINE EIG(C,EE,W,Z,CHEPS,NORM,M1,N,NX)
C
C     CALCULA LOS VECTORES CARACTERISTICOS DE LA MATRIZ TRIDIAGONAL
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION C(NX),W(NX),EE(NX),Z(NX,NX)
      DIMENSION P(200),Q(200),R(200),X(200),INT(200)
      REAL*8 LAMDA,NORM,M(200)
      II=N-1
      DO 23 I=1,II
   23 EE(I)=EE(I+1)
      EE(N)=0.
      EPS=CHEPS*NORM
      J=M1+1
      DO 1 JJ=1,M1
      J=J-1
      LAMDA=LAMDA-EPS
      IF(W(J).LT.LAMDA)LAMDA=W(J)
      U=C(1)-LAMDA
      V=EE(1)
      IF(V.EQ.0)V=EPS
      KK=N-1
      DO 2 I=1,KK
      II=I+1
      B1=EE(I)
      IF(B1.EQ.0)B1=EPS
```

```
      BI1=EE(I+1)
      IF(BI1.EQ.0)BI1=EPS
      IF(DABS(BI1).LT.DABS(U))GO TO 3
      M(I+1)=U/BI
      IF(M(I+1).NE.0)GO TO 4
      IF(BI.LE.EPS)M(I+1)=1.0
    4 P(I)=BI
      Q(I)=C(I+1)-LAMDA
      R(I)=BI1
      U=V-M(I+1)*Q(I)
      V=-M(I+1)*R(I)
      INT(I+1)=+1
      GO TO 2
    3 M(I+1)=BI/U
      P(I)=U
      Q(I)=V
      R(I)=0
      U=C(I+1)-LAMDA-M(I+1)*V
      V=BI1
      INT(I+1)=-1
    2 CONTINUE
      P(N)=U
      Q(N)=0
      R(N)=0
      X(N+1)=0
      X(N+2)=0
      H=0
      ETA=1.0/N
      I=N+1
      DO 5 II=1,N
      I=I-1
      U=ETA-Q(I)*X(I+1)-R(I)*X(I+2)
      IF(P(I).EQ.0)X(I)=U/EPS
      IF(P(I).NE.0)X(I)=U/P(I)
    5 H=H+DABS(X(I))
      H=1/H
      DO 6 I=1,N
    6 X(I)=X(I)*H
      DO 7 I=2,N
      IF(INT(I).LE.0)GO TO 8
      U=X(I-1)
      X(I-1)=X(I)
      X(I)=U-M(I)*X(I-1)
      GO TO 7
    8 X(I)=X(I)-M(I)*X(I-1)
    7 CONTINUE
      H=0
      I=N+1
      DO 9 II=1,N
      I=I-1
      U=X(I)-Q(I)*X(I+1)-R(I)*X(I+2)
      IF(P(I).EQ.0)X(I)=U/EPS
      IF(P(I).NE.0)X(I)=U/P(I)
    9 H=H+X(I)*X(I)
      H=1.D0/DSQRT(H)
      DO 10 I=1,N
   10 Z(I,J)=X(I)*H
    1 CONTINUE
C
C     CAMBIA LOS INDICES DE LA SUBROUTINA DAUC
```

```
C
      NN=N-1
      I=N
      DO 22 II=1,NN
      I=I-1
   22 EE(I+1)=EE(I)
      EE(1)=0.
      RETURN
      END
      SUBROUTINE BACK(A,EE,Z,M1,N,NX)
C
C     CALCULA LOS VECTORES CARACTERISTICOS DE LA MATRIZ REDUCIDA.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Z(NX,NX),EE(NX),A(NX,NX)
      DO 1 I=2,N
      IF(EE(I).EQ.0.0)GO TO 1
      L=I-1
      H=EE(I)*A(I,I-1)
      DO 2 J=1,M1
      S=0.0
      DO 3 K=1,L
    3 S=S+A(I,K)*Z(K,J)
      S=S/H
      DO 4 K=1,L
    4 Z(K,J)=Z(K,J)+S*A(I,K)
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
      SUBROUTINE TRANS(B,DL,Z,M1,N,NX)
C
C     CALCULA LOS VECTORES CARACT. DEL PROBLEMA ORIGINAL.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Z(NX,NX),DL(NX),B(NX,NX)
      DO 1 J=1,M1
      I=N+1
      DO 1 II=1,N
      I=I-1
      X=Z(I,J)
      III=I+1
      IF(III.GT.N)GO TO 3
      DO 2 K=III,N
    2 X=X-B(K,I)*Z(K,J)
    3 JJ=1
    1 Z(I,J)=X/DL(I)
      DO 5 J=1,M1
      H=DABS(Z(1,J))
      DO 28 I=2,N
   28 IF(DABS(Z(I,J)).GT.DABS(H))H=DABS(Z(I,J))
      DO 29 I=1,N
   29 Z(I,J)=Z(I,J)/H
    5 CONTINUE
      RETURN
      END
```