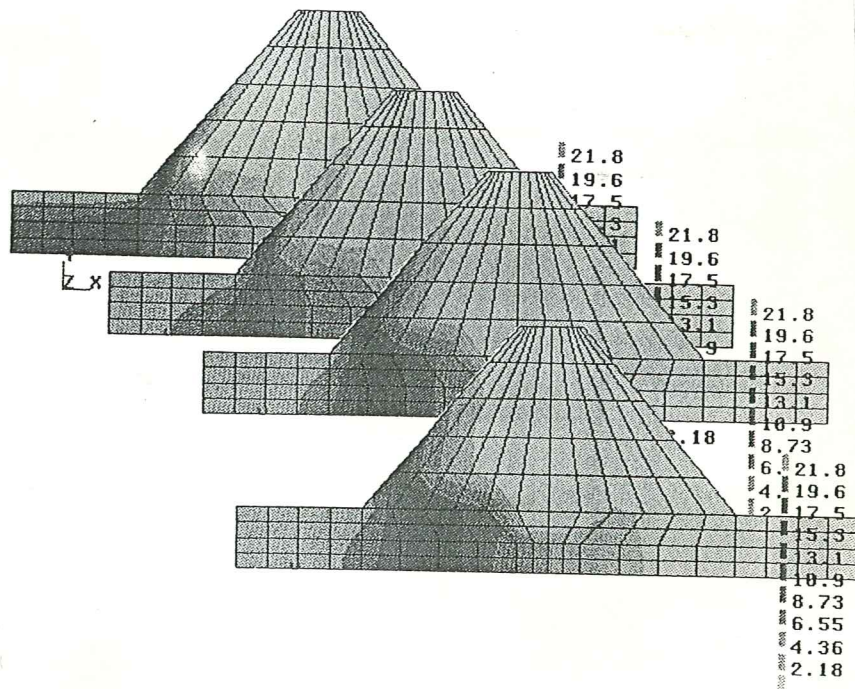


***CALTEP*: Programa para el cálculo transitorio de la ecuación de Poisson**

F. Zárate

E. Oñate



***CALTEP*: Programa para el cálculo transitorio de la ecuación de Poisson**

F. Zárate

E. Oñate

Publicación CIMNE Nº 27, Enero 1993

Centro Internacional de Métodos Numéricos en Ingeniería

Gran Capitán s/n, 08034 Barcelona, España

INDICE

CALTEP Programa para el cálculo transitorio de la ecuación de Poisson.

1	INTRODUCCION.....	1
2	ECUACIONES BASICAS.....	1
3	SOLUCION POR EL METODO DE LOS ELEMENTOS FINITOS	4
	3.1 FORMULACION BASICA	4
	3.2 DISCRETIZACION POR ELEMENTOS FINITOS.....	5
	3.3 ECUACIONES DE LA DISCRETIZACION.....	6
	3.3.1 CASO ESTACIONARIO	6
	3.3.2 CASO TRANSITORIO	6
	3.3.2.1 DIAGONALIZACION POR SUMA DE FILAS.....	7
	3.3.2.2 DIAGONALIZACION POR CONSERVACION DE LA MASA EQUIVALENTE	7
4	CARACTERISTICAS DEL PROGRAMA CALTEP	8
5	ORGANIZACION GENERAL CALTEP	9
	5.1 ETAPAS BASICAS. DIAGRAMA DE FLUJO PRINCIPAL	9
	5.2 SELECCION DE LOS NOMBRES DE LAS VARIABLES	12
	5.3 TRANSMISION DE INFORMACION ENTRE SUBRUTINAS	13
	5.4 LISTADO DE LA SUBROUTINA PRINCIPAL DE CALTEP.....	14
6	DESCRIPCION DE LA SUBROUTINA DATOS	16
	6.1 PARAMETROS DE CONTROL	16
	6.2 DATOS GEOMETRICOS	17
	6.3 CONDICIONES DE NODOS PRESCRITOS	17
	6.4 PROPIEDADES DEL MATERIAL	18
	6.5 CONDICIONES INICIALES DE TEMPERATURA	18
	6.6 CONDICIONES DE FLUJO DE CONVECCION / RADIACION EN LA FRONTERA	19
	6.7 SUBROUTINA GAUSS	19
	6.8 PREPARACION AUTOMATICA DE DATOS	19
	6.9 SUBROUTINA DE CONTROL DE DATOS	20
7	DESCRIPCION DE LAS SUBRUTINAS DE LA MATRIZ DE RIGIDEZ	20
	7.1 MATRIZ DE RIGIDEZ: SUBROUTINA RIGIMAT	20
	7.1.1 CASO ESTACIONARIO	20
	7.1.2 CASO TRANSITORIO	21
	7.2 MATRIZ DE RIGIDEZ: SUBROUTINA CONVECC	23
8	DESCRIPCION DE LA SUBROUTINA FUERZAS	25
	8.1 CONSIDERACIONES GENERALES	25
	8.2 FLUJOS PUNTUALES NODALES	26
	8.3 GENERACION INTERNA	26
	8.4 TEMPERATURA AMBIENTAL Y FLUJO EXTERIOR REPARTIDO SOBRE UN LADO	27
9	SOLUCION DEL SISTEMA DE ECUACIONES. SUBROUTINA SOLUCION	28
10	CALCULO DE LOS FLUJOS ELEMENTALES. SUBROUTINA TENSION	32
11	EJEMPLOS DE UTILIZACION DEL PROGRAMA CALTEP	33
	11.1 FLUJO A TRAVES DE UNA BARRA DELGADA	33
	11.2 GENERACION INTERNA DE CALOR EN UN DOMINIO CUADRADO	35
	11.2.1 CASO ESTACIONARIO	35
	11.2.2 CASO TRANSITORIO A=1.0	41
	11.2.3 CASO TRANSITORIO A=2/3	43
	REFERENCIAS	47
	APENDICE I	I.i
	I.I INSTRUCCIONES PARA ENTRADA DE DATOS	I.i
	I.II LISTADO DE VARIABLES DEL PROGRAMA CALTEP	I.ii
	APENDICE II	II.ii
	II.I LISTADO DEL PROGRAMA CALTEP	II.ii

1 INTRODUCCION

Esta publicación explica la utilización de un programa de elementos finitos que permita resolver la ecuación de Poisson transitoria, que rige una gran cantidad de problemas físicos como son la transmisión del calor a través de diversos medios, el flujo de un liquido a través de un medio permeable, problemas de magnetismo, etc.

En la publicación describe las etapas que intervienen en un programa de elementos finitos para cálculo de la ecuación de Poisson transitoria incidiendo principalmente en la metodología general de la programación de las diferentes subrutinas, así como en su aplicación a varios problemas. En la última parte de esta publicación se presentan diversos ejemplos de aplicación del programa, así como un listado completo del mismo, una descripción de sus variables más significativas y las instrucciones para la entrada de datos.

2 ECUACIONES BASICAS

La modelización del comportamiento de muchos fenómenos naturales puede describirse utilizando la formulación diferencial cuasi armónica conocida en el argot matemático como la ecuación de Poisson.

Dicha ecuación puede modelizar, entre otros problemas, el mecanismo de conducción de calor a través de un cuerpo, o bien, el flujo de un liquido en un medio permeable. En un siguiente apartado se dará una descripción de los diversos fenómenos naturales que rige esta ecuación, así como la equivalencia de las variables usadas. Sin embargo, para poder dar un significado tangible a la descripción de las ecuaciones que se utilizaran, se ha tomado como ejemplo concreto el problema de la conducción de calor, ya que se puede considerar como la aplicación mas sencilla y generalizada.

La ecuación de Poisson para un domino bidimensional puede expresarse en la forma siguiente:

$$\rho c \frac{\partial \phi}{\partial t} = \mathbf{D} \nabla^2 \phi + \rho r \quad \text{en } \Omega \quad (\text{i})$$

En donde ϕ representa la temperatura, t la variable tiempo, ρ la densidad del dominio, c el calor específico y ρr la densidad por fuente de calor interno; mientras que \mathbf{D} corresponde a la matriz constitutiva, formada por las conductividades térmicas, K , para las distintas dimensiones en que se describe el domino Ω ; de esta manera, para un domino bidimensional corresponde a:

$$\mathbf{D} = \begin{bmatrix} K_x & \mathbf{0} \\ \mathbf{0} & K_y \end{bmatrix} \quad (\text{ii})$$

Alternativamente \mathbf{D} queda descrita para el caso uni y tridimensional como:

$$\mathbf{D} = [K_x] \text{ caso unidimensional.} \quad (\text{iii})$$

$$\mathbf{D} = \begin{bmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & K_z \end{bmatrix} \text{ caso tridimensional.} \quad (\text{iv})$$

Las condiciones de contorno a las que se encuentra sujeta la formulación anterior se esquematizan en la Figura 1, y son conocidas en términos matemáticos como las condiciones de (a) *Dirichlet* que fija la temperatura ϕ , a un valor fijo y conocido de antemano sobre un contorno particular, y (b) *Cauchy* que fija el gradiente de la temperatura normal a la superficie.

$$(a) \phi - \bar{\phi} = 0 \text{ en } \Gamma_\phi \quad (\text{v})$$

$$(b) \mathbf{n}^T \mathbf{q} + \alpha(\phi - \phi_{ext}) + \bar{q} = 0 \text{ en } \Gamma_q \quad (\text{vi})$$

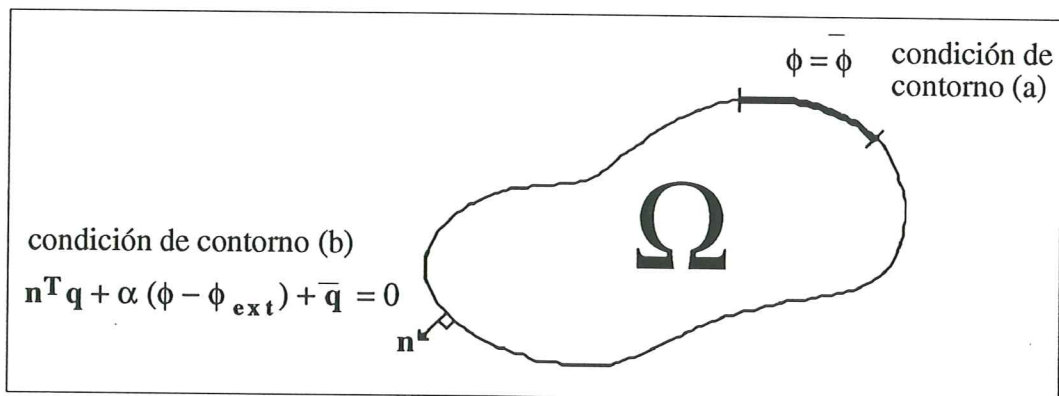


Figura 1 Región bidimensional con las condiciones de contorno permisibles.

En las ecuaciones anteriores $\bar{\phi}$ es la temperatura con valor conocido en la frontera, α representa al coeficiente de convección-radiación, \bar{q} es el flujo o gradiente de la temperatura, con valor conocido en la frontera, ϕ_{ext} es la temperatura en el exterior del dominio, mientras que el resto de las variables quedan definidas con las siguientes expresiones:

Vector de normales al contorno:

$$\mathbf{n} = [n_x, n_y]^T \quad (\text{vii})$$

Vector de gradientes de la temperatura:

$$\mathbf{q}_n = [q_x, q_y] = -\mathbf{D}\mathbf{g} = -\mathbf{D} \nabla\phi \quad (\text{viii})$$

En donde el gradiente ∇ corresponde a:

$$\nabla = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right]^T \quad (\text{ix})$$

Resulta inmediato la expresión de las ecuaciones (vii), (viii) y (ix) para los casos uni y tridimensionales, por lo que se omiten sus desarrollos.

Dependiendo de los valores escogidos para los parámetros de la ecuación (vi) es posible reproducir los siguientes casos en el contorno:

a) Contorno aislante. Matemáticamente conocido como la condición de *Newman* o condición de contorno natural que representa un flujo de temperatura cero en la interfase del dominio con el medio circundante, de manera que:

$$\mathbf{n}^T \mathbf{q} = 0 \text{ por lo que } \bar{q} = 0 \text{ y } \alpha = 0 \quad (\text{vi.a})$$

b) Contorno con pérdida o aporte de flujo externo. La condición señalada queda representada matemáticamente por:

$$\mathbf{n}^T \mathbf{q} = -\bar{q} \text{ por lo que } \alpha = 0 \quad (\text{vi.b})$$

c) Contorno con pérdida o aporte de calor por convección - radiación. En este caso existe una interacción del calor existente en el medio exterior ϕ_{ext} y la temperatura del dominio ϕ que de acuerdo con las leyes de la termodinámica se presentara un flujo en la interfase, representado por:

$$\mathbf{n}^T \mathbf{q} = -\alpha(\phi - \phi_{ext}) \text{ con lo que } \bar{q} = 0 \quad (\text{vi.c})$$

El factor α estrictamente representa el fenómeno de convección, mientras que el fenómeno de radiación sigue una ley mas compleja pero que a efectos de simplificación se puede expresar de la misma manera que la expresada en (vi.c), por lo que α representa la contribución de ambos

Como se ha mencionado en párrafos anteriores, los términos de las ecuaciones (i) a (ix) tienen diferentes significados físicos, dependiendo del problema que se modelice. A continuación se listan algunos de los casos que son mas comunes:

	Problema Térmico	Filtración en medios poroso	Electromagnetismo	Torsión de barras
ϕ	Temperatura	Altura Piezométrica	Potencial magnético	Esfuerzo torsor
D	Matriz de conductividad térmica	Matriz de permeabilidad	Matriz de reluctancia	Matriz de rigidez
r	Fuente de calor interno por unidad de masa	Caudal aportado en el medio	Fuente magnética interna	-
q	vector de flujo de calor	Flujo de agua en el contorno	Flujo magnético externo	Momento torsor
ρ	Densidad	Densidad	Densidad	Densidad
c	Calor específico por unidad de masa	-	Aportación magnética por unidad de masa	-
α	Coefficiente de convección-radiación	-	-	-

Figura 2 Reinterpretación física de las variables de la ecuación de Poisson.

3 SOLUCION POR EL METODO DE LOS ELEMENTOS FINITOS

3.1 Formulación Básica.

La aplicación del método de los elementos finitos exige como punto de partida la existencia de una forma integral expresando el mecanismo global del sistema. Dicha forma integral puede obtenerse aplicando el método de los residuos ponderados a la ecuación diferencial (i) y a la condición de contorno (vi) como:

$$\int_{\Omega} [\nabla^T \mathbf{D} \nabla \phi + \rho r] \partial \Omega - \int_{\Omega} \rho c \frac{\partial \phi}{\partial t} \partial \Omega + \oint_{\Gamma_q} [\mathbf{n}^T \mathbf{D} \nabla \phi + \alpha(\phi - \phi_{ext}) + \bar{q}] \partial \Gamma_q = 0 \quad (x)$$

La condición de contorno (v) no necesita incluirse en la expresión anterior puesto que se satisface imponiendo directamente el valor de ϕ al valor prescrito en los contornos adecuados (condición forzada).

Tras integrar por partes el termino $\nabla^T \mathbf{D} \nabla \phi$ y reagrupar la ecuación (x) se obtiene (después de hacer $\mathbf{W} = -\mathbf{W}$)

$$\int_{\Omega} \mathbf{W}^T \rho c \frac{\partial \phi}{\partial t} \partial \Omega + \int_{\Omega} \nabla^T \mathbf{W}^T \mathbf{D} \nabla \phi \partial \Omega + \oint_{\Gamma_q} \mathbf{W}^T \alpha \phi \partial \Gamma_q = \int_{\Omega} \mathbf{W}^T p r \partial \Omega - \oint_{\Gamma_q} \mathbf{W}^T \bar{\mathbf{q}}_n \partial \Gamma_q + \oint_{\Gamma_q} \mathbf{W}^T \alpha \phi_{ext} \partial \Gamma_q - \oint_{\Gamma_{\phi}} \mathbf{W}^T \mathbf{q}_n \partial \Gamma_{\phi} \tag{xi}$$

Observe que la ultima integral de (xi) incluye el flujo normal en los contornos donde ϕ esta prescrita. En la practica este flujo se calcula "a posteriori", una vez obtenidas las temperaturas en todos los nodos.

3.2 Discretización por Elementos Finitos

Después de discretizar el dominio en elementos en la forma clásica, (figura 3), las temperaturas se interpolan en el interior de cada elemento como:

$$\phi = \sum N_i \phi_i = \mathbf{N} \mathbf{a}^{(e)} \tag{xii}$$

donde N_i son las funciones de forma definidas en cada elemento y $\mathbf{a}^{(e)}$ contiene los valores de las temperaturas nodales del elemento (e).

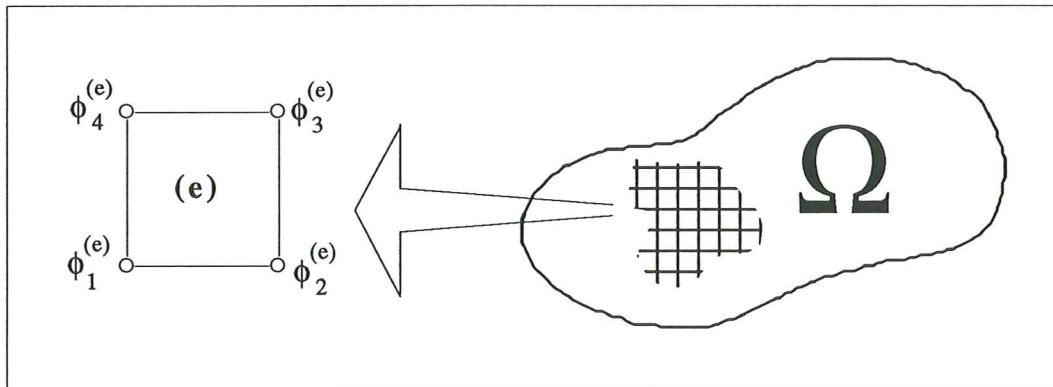


Figura 3 Discretización del dominio en Elementos Finitos.

El vector de gradientes en cada elemento se obtiene por:

$$\mathbf{g} = \nabla \phi = \nabla \mathbf{N} \mathbf{a}^{(e)} = \mathbf{B} \mathbf{a}^{(e)} \tag{xiii}$$

donde

$$\mathbf{B} = [\mathbf{B}_1, \dots, \mathbf{B}_n] \tag{xiv}$$

siendo

$$\mathbf{B}_i = \begin{bmatrix} \frac{\partial N_i}{\partial x} \end{bmatrix}, \quad \mathbf{B}_i = \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} \text{ y } \mathbf{B}_i = \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} \quad (\text{xv})$$

para problemas uni, bi y tridimensionales, respectivamente.

El vector de flujos puede calcularse en función de los valores nodales como:

$$\mathbf{q} = -\mathbf{DBa}^{(e)} \quad (\text{xvi})$$

3.3 Ecuaciones de la discretización

Sustituyendo la discretización (xii) y (xiii) en (xi) y haciendo $\mathbf{W}=\mathbf{N}$ (método de Galerkin) se obtiene un sistema matricial de ecuaciones que puede escribirse en la forma:

$$\mathbf{M} \frac{\partial \mathbf{a}}{\partial t} + \mathbf{Ka} = \mathbf{f} \quad (\text{xvii})$$

En (xvii), \mathbf{a} es el vector de incógnitas que contiene la temperatura de todos los nodos de la malla, y \mathbf{M} , \mathbf{K} , y \mathbf{f} son la matriz de masa, la matriz de rigidez y el vector de fuerzas nodales que pueden obtenerse ensamblando las contribuciones elementales definidas por

$$\mathbf{M}^{(e)} = \int_{\Omega^{(e)}} \rho c \mathbf{N}^T \mathbf{N} \, d\Omega^{(e)} \quad (\text{xviii})$$

$$\mathbf{K}^{(e)} = \int_{\Omega^{(e)}} \mathbf{B}^T \mathbf{DB} \, d\Omega^{(e)} + \alpha \oint_{\Gamma_q^{(e)}} \mathbf{N}^T \mathbf{N} \, d\Gamma_q^{(e)} \quad (\text{xix})$$

$$\mathbf{f}^{(e)} = \int_{\Omega^{(e)}} \mathbf{N}^T \rho r \, d\Omega^{(e)} - \oint_{\Gamma_q^{(e)}} \mathbf{N}^T \bar{q} \, d\Gamma_q^{(e)} + \oint_{\Gamma_q^{(e)}} \mathbf{N}^T \alpha \phi_{ext} \, d\Gamma_q^{(e)} - \oint_{\Gamma_\phi^{(e)}} \mathbf{n}^T \mathbf{N}^T \mathbf{q}_n \, d\Gamma_\phi^{(e)} \quad (\text{xx})$$

3.3.1 Caso Estacionario.

La solución estacionaria implica únicamente resolver el sistema de ecuaciones

$$\mathbf{Ka} = \mathbf{f} \quad (\text{xxi})$$

Obtenidas las temperaturas nodales pueden obtenerse los gradientes térmicos y los flujos de calor en cada punto mediante las ecuaciones (xiii) y (xvi), respectivamente. Asimismo, pueden obtenerse los flujos de calor a través de los contornos con temperaturas prescritas.

3.3.2 Caso Transitorio.

La solución del problema transitorio exige integrar en el tiempo la ecuación (xvii) . Utilizando un esquema de diferencias finitas trapezoidal generalizado se obtiene:

$$\left[\frac{\mathbf{M}}{\Delta t} + \alpha \mathbf{K} \right] \mathbf{a}^t = \alpha \mathbf{f}^t + (1 - \alpha) \mathbf{f}^{t-1} + \left[\frac{\mathbf{M}}{\Delta t} - (1 - \alpha) \mathbf{K} \right] \mathbf{a}^{t-1} \quad (\text{xxii})$$

donde α define el punto de "colocación" de la ecuación diferencial (xvii) , es decir:

$$\mathbf{a}^\alpha = \alpha \mathbf{a}^t + (1 - \alpha) \mathbf{a}^{t-1} \quad (\text{xxiii})$$

La ecuación (xxiii) permite obtener el valor de las temperaturas en el tiempo t en función de las temperaturas en el instante $t-1$ y de los valores de las "fuerzas" en t y $t-1$.

Puede demostrarse que el esquema trapezoidal de la ecuación (xxiii) es incondicionalmente estable para $\alpha \geq \frac{1}{2}$

En la practica se recomienda tomar $\alpha = \frac{2}{3}$ (Galerkin) ó $\alpha = 1$ (Fuertemente implícito) [21]. La integración de $\alpha = 0$ (método explícito de Euler)[21] es ventajosa si la matriz \mathbf{M} es diagonal, puesto que en este caso la solución es explícita y no precisa la inversión de ninguna matriz. La única dificultad en este caso se debe a que el sistema de integración es condicionalmente estable y exige la utilización de incrementos de tiempo pequeños que vienen gobernados por la condición:

$$\Delta t = \frac{L}{\omega} \quad (\text{xxiv})$$

siendo ω el mayor valor propio del sistema homogéneo:

$$[\mathbf{K} + \mathbf{M}\omega] \mathbf{a} = \mathbf{0} \quad (\text{xxv})$$

El programa *CALTEP* permite la integración explícita con dos posibilidades para diagonalizar la matriz \mathbf{M} :

3.3.2.1 Diagonalización por suma de filas.

Se expresa por:

$$\begin{aligned} M_{Dii} &= \sum_{j=1} M_{ij} \\ M_{Dij} &= 0 \end{aligned} \quad (\text{xxvi})$$

en donde la diagonal de la matriz de masa es la suma de de todos los elementos contenidos en la fila *i-esima*

3.3.2.2 Diagonalización por conservación de la masa:

Se expresa por:

$$M_{Dii} = p \int_{\Omega} N_i^T \rho c N_i \partial \Omega$$

$$\mathbf{M}_{Dij} = \mathbf{0} \quad (\text{xxvii})$$

con

$$\sum_i \mathbf{M}_{ii} = \int_{\Omega} \rho c \partial \Omega = p \quad (\text{xxviii})$$

de manera que la nueva matriz \mathbf{M} conserve la misma masa que la matriz original.

4 CARACTERISTICAS DEL PROGRAMA CALTEP

En los apartados siguientes se presenta la descripción del programa *CALTEP* para Cálculo Transitorio de la Ecuación de Poisson por el Método de Elementos Finitos con detalles de las subrutinas más relevantes del mismo. La versión de *CALTEP* que se presenta escrita en FORTRAN tiene las siguientes características generales:

Características del material

- Material lineal isótropo.

Elementos utilizables

Pueden utilizarse los elementos isoparamétricos siguientes:

Elemento

- Elemento lineal de dos nodos
- Elemento lineal de tres nodos
- Elemento triangular de tres nodos
- Elemento triangular de 6 nodos
- Elemento cuadrilátero lagrangianos de 4 nodos
- Elemento cuadrilátero serendípito de 8 nodos
- Elemento cuadrilátero lagrangiano de 9 nodos
- Elemento hexagonal serendípito de 20 nodos tridimensional

Condiciones de contorno

Para cada problema se admiten únicamente cargas *estáticas* de los tipos siguientes:

<u>Problema</u>	<u>Tipos de carga</u>
-----------------	-----------------------

Problema Unico	- Temperaturas puntuales nodales
----------------	----------------------------------

- Generación Interna de calor
- Flujo exterior uniformemente repartido
sobre los lados de los elementos
(condición de Convección/Radiación)
- Temperatura del medio externo cte.
- Temperatura puntual inicial
(para el problema transitorio)

Una vez familiarizado el lector con los detalles del programa podrá modificarlo fácilmente para incluir otros tipos de elementos y condiciones de contorno, así como para extender su rango de aplicación a otro tipo de problemas.

5 ORGANIZACION GENERAL CALTEP

5.1 Etapas básicas. Diagrama de flujo principal

En la Figura 4 se muestra el diagrama de flujo principal del programa *CALTEP*. Para centrar conceptos definiremos seguidamente las etapas fundamentales asociadas al análisis de un problema cualquiera mediante un programa de elementos finitos, así como la relación de cada etapa con las subrutinas del diagrama de la Figura 4.

Etapa 1: Selección del elemento

La elección del elemento es función de la tipología del problema y de la precisión buscada. Una vez escogido el elemento quedan definidas sus funciones de forma. En el apartado anterior se han definido los tipos de elementos incluidos en la versión del programa que aquí se presenta.

Etapa 2: Discretización de la geometría en elementos finitos

Esta etapa puede representar un porcentaje alto del esfuerzo total de cálculo si la geometría del problema a modelizar es compleja. En ella hay que definir perfectamente la topología de la malla (que de nuevo depende de la geometría y la precisión buscada), las coordenadas de los nodos, las propiedades del material de cada elemento y las condiciones de contorno. Esta etapa se denomina generalmente *preproceso* y puede automatizarse en gran manera si se dispone de los programas de generación de malla adecuados [5], [6], [7]. Esta automatización es mucho más esencial si se utilizan técnicas de solución adaptables.

Es importante destacar que el coste de la solución del sistema de ecuaciones global depende en gran manera de: a) la numeración de los nodos de la malla (ejemplo: si se utiliza el método de eliminación de Gauss [2-4]), o b) la numeración de los elementos (ejemplo: si se utiliza el método frontal [2]). Conviene, por tanto, cuidar la topología de la malla y adecuarla en lo posible al método de solución de ecuaciones utilizado. Para ello puede hacerse uso de técnicas especiales de optimización de la numeración de nodos y/o elementos [2], [4]. En el Apartado 9 se volverá a tratar este tema.

Etapa 3: Entrada de datos (Subrutina DATOS)

Esta etapa consiste en la lectura por el ordenador de los datos generados en la discretización. Dicha lectura se efectúa en la subrutina **DATOS** detallada en el Apartado 6.

Etapa 4: Cálculo de la matriz de rigidez de los elementos (Subrutina RIGIMAT)

En la etapa siguiente se calculan las matrices de rigidez $K^{(e)}$ de cada uno de los elemento de la malla. Dicho cálculo se efectúa en la subrutina **RIGIMAT** y su mayor o menor complejidad depende del tipo de elemento utilizado. En el caso de tener un problema transitorio se calculará también la correspondiente matriz de masa, así como la aportación a la matriz de rigidez por condiciones de contorno generadas por convección-radiación, en la subrutina **CONVECC**.

Etapa 5: Cálculo del vector de temperaturas nodales (Subrutina FUERZAS)

La siguiente etapa es el cálculo del vector de temperaturas nodales equivalentes $f^{(e)}$ para cada elemento y se efectúa en la subrutina **FUERZAS**. De nuevo su mayor o menor complejidad depende del elemento utilizado y también de las cargas exteriores consideradas. La descripción de **FUERZAS** se incluyen en el Apartado 8. En caso de solucionar un problema transitorio, se considera las aportaciones necesarias de las condiciones fijadas inicialmente, y de la matriz de rigidez por el esquema de integración en el tiempo, que en cada incremento de tiempo Δt se ira actualizando.

Etapa 6: Solución del sistema de ecuaciones global (Subrutina SOLUCION)

Conocidas las matrices de rigidez y los vectores de temperaturas nodales de cada elemento la etapa siguiente es el ensamblaje de dichas matrices y vectores en la ecuación

de equilibrio global $Ka = f$, y su solución para obtener las temperaturas nodales a . Este proceso se efectúa en la subrutina **SOLUCION** y para el mismo puede utilizarse toda una

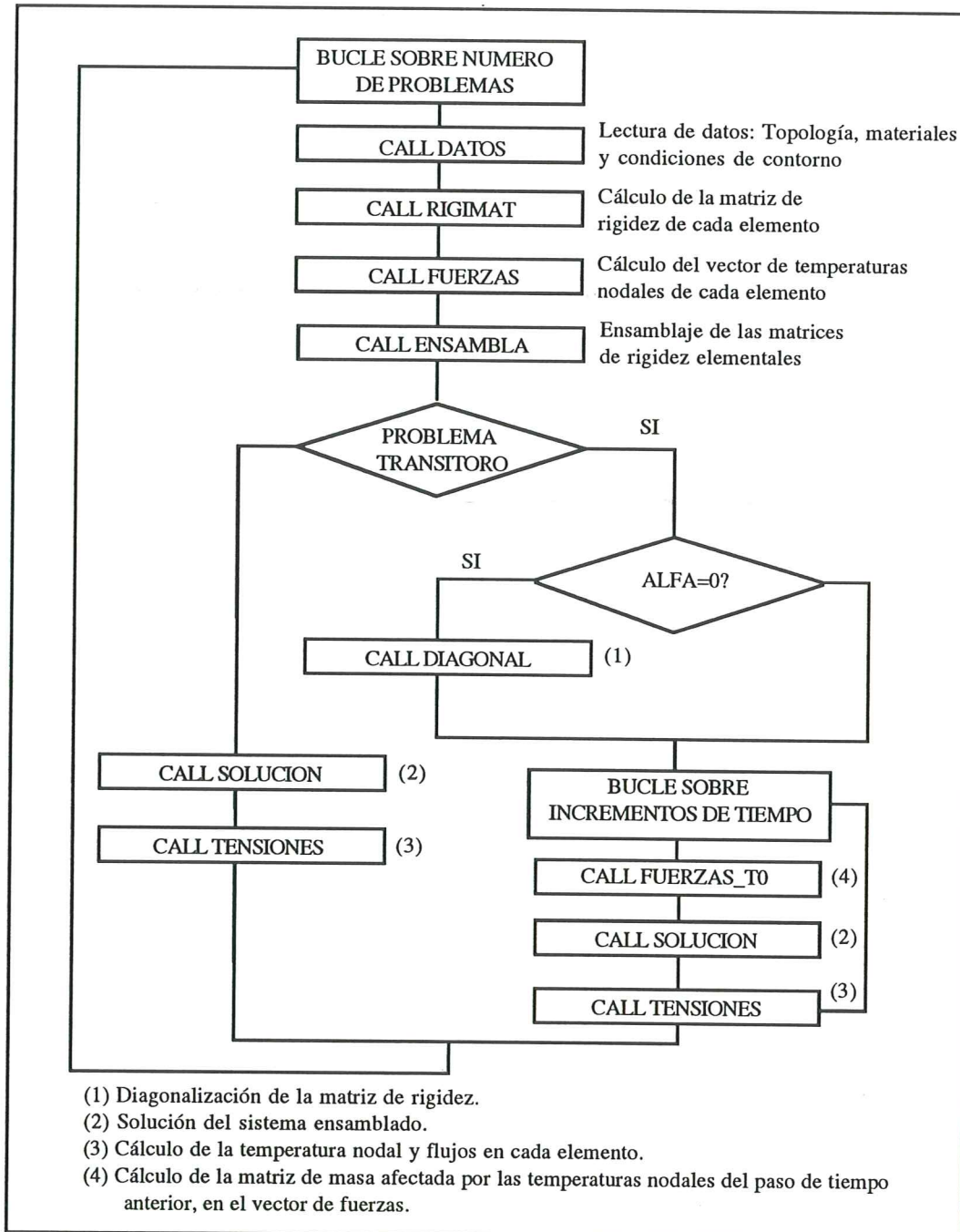


Figura 4 Diagrama de flujo del programa *CALTEP*.

variedad de técnicas de cálculo numérico perfectamente desarrolladas. En el Apartado 9 se hace referencia a algunas de ellas al tratar con más detalle el contenido de esta subrutina.

Etapa 7: Cálculo de gradientes (Subrutina TENSIONES)

La etapa final consiste en calcular los flujos o gradientes de la temperatura en los diferentes elementos a partir de los valores nodales. Dicho cálculo se lleva a cabo en la subrutina **TENSIONES** que se describe en el Apartado 10. Esta etapa de *postproceso* va también asociada en la práctica a la representación gráfica de los resultados del cálculo (temperaturas y flujos). En caso de tener un problema transitorio, se harán los cambios necesarios sobre los vectores de temperaturas y se cerrará un ciclo de tiempo para continuar iterativamente, hasta llegar a la convergencia, o bien, al número de iteraciones previstas. La programación del postproceso no es un problema trivial, fundamentalmente en problemas tridimensionales, siendo necesario un conocimiento profundo de técnicas de dibujo por ordenador. No obstante, generalmente puede hacerse uso de programas comerciales "*ad hoc*" y el problema se reduce a compatibilizar los formatos de resultados del cálculo con los necesarios para su representación gráfica [5-9].

La versión 1/92 de **CALTEP** para micro ordenadores incluye software para representación gráfica de datos y resultados compatibles con las subrutinas gráficas de MICROSOFT 5.0.y Finder 6.0.2 o menores para Macintosh.

5.2 Selección de los nombres de las variables

Una norma elemental de buena programación es mantener un criterio uniforme para escoger el nombre de las variables del programa [2]. Salvo contadas excepciones, en **CALTEP** todas las variables tienen cinco letras y además se ha tratado de que el nombre de cada una esté lo más relacionado posible con su función en el programa. Por otro lado, todas las variables que empiezan con la letra N indican "número de", si empiezan con M indican "número máximo de" y si lo hacen con I, J, K, indican un valor determinado de la variable. Así, por ejemplo, NELEM es el número de elementos de la malla, MELEM es el número máximo de elementos que puede analizar el programa, IELEM es un elemento determinado de los NELEM, etc.

Estas normas pueden resultar algo tediosas a los que se inician en la programación del método de los elementos finitos. No obstante, su estricto cumplimiento es de gran utilidad, tanto para el estudio y utilización de un programa determinado, como para su posterior modificación por personas ajenas a su desarrollo inicial (e incluso para el propio autor).

En el Apéndice I se presenta una relación de las principales variables del programa juntamente con la explicación de su significado.

5.3 Transmisión de información entre subrutinas

En *CALTEP* se utilizan bloques COMMON para transmitir toda la información necesaria (escalar y vectorial) entre subrutinas. Dichos bloques son iguales en *todas* las subrutinas, y sus vectores se han dimensionado en función del problema de mayor tamaño que puede analizarse con el programa. Se ha escogido esta alternativa frente a otras opciones posibles (dimensionamiento dinámico, transmisión por argumentos, etc.) por su valor didáctico para los no iniciados en programación.

CALTEP utiliza fundamentalmente cuatro bloques COMMON. El primero denominado DTSGRAL almacena los parámetros de control definidos en la subrutina DATOS y que son necesarios en todo el programa. El bloque COMMON DTSGRAL tiene la forma siguiente:

```
COMMON /DTSGRAL/   NTRAN, NDIME, NELEM, NEVAB, NGAUS, NGDLN,
                   NMATS, NNODE, NPNOD, NPRES, NPROB, NPROP,
                   NTENS, NTIPO, NFRON, NNFRO, NTIME, ITIME,
                   ALFAT, XLUMP
```

El segundo bloque COMMON se denomina DATA y almacena un conjunto de matrices y vectores relacionados con la geometría de la malla, las propiedades de los materiales, las condiciones de contorno y las temperaturas nodales. La forma de dicho bloque es

```
COMMON /DATA/      COORD (MPNOD, MDIME), CARGA (MELEM, NEVAB),
                   INPRE (MPRES, MGDLN), NODPR (MPRES),
                   LNODS (MELEM, MNODE), MATNU (MELEM),
                   PROPS (MMATS, MPROP), PRESC (MPRES, MGDLN),
                   CONVE (MPRES, MNODE), XTIME (MPNOD)
```

El tercer bloque COMMON se denomina CALCULO y almacena un conjunto de matrices y vectores que están específicamente relacionados con el cálculo de la matriz de rigidez y el vector de temperaturas nodales. Dicho bloque tiene la forma general siguiente:

```
COMMON/ CALCULO/  BMATZ (MTENS, MEVAB), COREL (MDIME, MNODE),
                   CORPG (MDIME, MGASP), DERIV (MDIME, MNODE),
                   DBMAT (MTENS, MEVAB), DCART (MDIME, MNODE),
                   DMATZ (MTENS, MTENS), FFORM (MNODE),
                   TENSZ (MTENS, MEVAB, MGASP)
```

El cuarto bloque COMMON se denomina GAUSSDAT y almacena los datos generales asociados a los puntos de gauss relacionados con el cálculo. Dicho bloque tiene la forma general siguiente:

COMMON/GAUSSDAT/POSGT(MGAUS*MDIME), POSPG(MGAUS),
PESGT(MGAUS), PESPG(MGAUS)

Los argumentos de las matrices y vectores de los bloques COMMON se han dimensionado en el programa de manera que sirvan para todos los problemas resolubles. Dichos valores se muestran en la Figura 5.

MDIME = 3	MNODE = 27
MELEM = 200	MPNOD = 200
MEVAB = 81	MPRES = 200
MGAUS = 14	MPROP = 10
MGDLN = 3	MTENS = 3
MMATS = 10	MTOTV = 600

Figura 5 Dimensiones de las matrices y vectores de los bloques COMMON DATA, CALCULO y GAUSSDAT.

En el Apéndice I puede encontrarse el significado de las variables de los bloques COMMON anteriores.

5.4 Listado de la subrutina principal de CALTEP

A continuación se muestra un listado de la subrutina principal de CALTEP incluyendo los bloques COMMON.

```

1.      PROGRAM CALTEP
2.      C*****
3.      C
4.      C*** SUBROUTINA PRINCIPAL
5.      C
6.      C*****
7.      C*****COMMON STATEMENTS
8.          IMPLICIT NONE
9.          INCLUDE 'dtsgral.f'
10.         INCLUDE 'data.f'
11.         INCLUDE 'gaussdat.f'
12.         INCLUDE 'calculo.f'
13.         INCLUDE 'solu.f'
14.         INCLUDE 'solucuas.f'
15.         INTEGER*2 MPROB, IPROB, IVARI, ITIME
16.         CHARACTER*80 TITULO
17.         CHARACTER*10 INPUT, OUTPUT
18.         CHARACTER*13 AUX3
19.         DATA AUX3 /' /aux3/zarate/' /
20.      C*****COMMON STATEMENTS
21.         OPEN (2, FILE='COMAN.DAT', FORM='FORMATTED')
22.         READ(2, 800) INPUT
23.         READ(2, 800) OUTPUT
24.         CLOSE(2, STATUS='KEEP')
25.         800 FORMAT(A10)

```

```

26.          OPEN (5,FILE=INPUT,FORM='FORMATTED')
27.          OPEN (6,FILE=OUTPUT,FORM='FORMATTED')
28.          OPEN (1,FILE=AUX3//'TEMP1.TMP',FORM='UNFORMATTED')
29.          OPEN (3,FILE=AUX3//'TEMP3.TMP',FORM='UNFORMATTED')
30.          OPEN (4,FILE=AUX3//'TEMP4.TMP',FORM='UNFORMATTED')
31.          OPEN (8,FILE=AUX3//'TEMP8.TMP',FORM='UNFORMATTED')
32.          OPEN(10,FILE=AUX3//'TEMP10.TMP',FORM='UNFORMATTED')
33.          OPEN (29,FILE=AUX3//'CALSEF.POS', FORM='UNFORMATTED')
34.          C
35.          C*** LEE NUMERO DE PROBLEMAS A ANALIZAR
36.          C
37.          READ(5,900) NPROB
38.          900 FORMAT(I5)
39.          WRITE(6,905) NPROB
40.          905 FORMAT(/10X,'NUMERO DE PROBLEMAS= ',I5)
41.          C
42.          C*** BUCLE SOBRE NUMERO DE PROBLEMAS
43.          C
44.          MPROB=INT2(NPROB)
45.          ITIME=0
46.          WRITE(29) MPROB
47.          DO IPROB=1,NPROB
48.              REWIND 1
49.              REWIND 3
50.              REWIND 4
51.              WRITE(6,910) IPROB
52.          910  FORMAT(////,6X,'PROBLEMA NO.',I3,////)
53.              READ(5,915) TITULO
54.          915  FORMAT(A80)
55.              WRITE(6,920) TITULO
56.          920  FORMAT(A80,////)
57.              CALL DATOS
58.              CALL GAUSSQ(NDIME,NNODE,NGAUS)
59.              CALL RIGIMAT
60.              CALL FUERZAS
61.              CALL ENSAMBLA
62.              IF (NTRAN.EQ.0) THEN
63.                  CALL REDUCE
64.                  CALL SUSTITUIR
65.                  CALL TENSIONES
66.                  CALL SUAV1
67.                  CALL FEMV(ITIME,TITULO)
68.              ELSE
69.                  IF (ALFAT.EQ.0) CALL DIAGONAL
70.                  DO ITIME=1,NTIME
71.          C
72.          C***          APLICA LAS CONDICIONES DE TIEMPO T-1
73.          C
74.              CALL FUERZAS_T0
75.          C
76.          C***          RESUELVE POR ELIMINACION GAUSSIANA
77.          C
78.              IF (ITIME.EQ.1) THEN
79.                  CALL REDUCE
80.              ELSE
81.                  CALL REDUCE1
82.              ENDIF
83.              CALL SUSTITUIR
84.          C
85.          C***          CALCULA LAS TENSIONES EN LOS ELEMENTOS
86.          C
87.              CALL TENSIONES
88.          C
89.          C***          SUAVIZADO DE TENSIONES
90.          C
91.              CALL SUAV1

```



```

92.          CALL FEMV(ITIME,TITULO)
93.          C
94.          C***          EL TIEMPO T SE CONVIERTE EN T-1
95.          C
96.          DO IVARI=1,NGDLN*NPNOD
97.             XTIME(IVARI)=DESPL(IVARI)
98.          ENDDO
99.          C
100.         ENDDO
101.        ENDIF
102.       ENDDO
103.      CLOSE(1,STATUS='DELETE')
104.      CLOSE(3,STATUS='DELETE')
105.      CLOSE(4,STATUS='DELETE')
106.      CLOSE(8,STATUS='DELETE')
107.      CLOSE(10,STATUS='DELETE')
108.      CLOSE(29,STATUS='KEEP')
109.      CLOSE(5,STATUS='KEEP')
110.      CLOSE(6,STATUS='KEEP')
111.      STOP
112.      END

```

6 DESCRIPCION DE LA SUBROUTINA DATOS

Recordemos que en la subrutina **DATOS** se lee toda la información relacionada con la geometría de la malla y las propiedades de los materiales. **DATOS** está organizada de manera que sirva para cualquier problema térmico analizado con elementos uni, bi o tridimensionales. Seguidamente se describen las partes fundamentales de esta subrutina.

6.1 Parámetros de control

Dado que **CALTEP** se ha escrito para que pueda ser utilizable en más de un tipo de problemas, es esencial definir al principio una serie de parámetros de control que establezcan las características propias de cada problema. Dichos parámetros son los siguientes:

- * Número total de puntos nodales en la malla (NPNOD)
- * Número total de elementos en la malla (NELEM)
- * Número total de nodos con movimientos prescritos (NPRES)
- * Indicador de tipo de problema (NTRAN=0 estacionario, =1 transitorio.sin diagonalización,=2 transitorio con diagonalización por suma de filas, =3 transitorio.con diagonalizacion por masa equivalente)
- * Número de nodos por elemento (NNODE)
- * Número de materiales diferentes en la estructura (NMATS)
- * Orden de la cuadratura de Gauss utilizada (NGAUS)
- * Número de coordenadas necesarias para definir un nodo (NDIME)

- * Indicador de escritura (IWRITE= 0 deshabilitada, =1 habilitada.)

Recordemos que cuando uno de dichos parámetros comienza por la letra M su valor es el *máximo* que admite el programa.

6.2 Datos geométricos

La geometría de la malla puede definirse a partir de las conexiones nodales de cada elemento y las coordenadas de los nodos.

a) *Definición de las conexiones nodales de los elementos*

La geometría de cada elemento se define listando de manera sistemática sus nodos. Como en principio cada elemento puede tener diferentes propiedades del material, es conveniente asignar un número de identificación del material a cada elemento. Por consiguiente, los datos a leer para cada elemento son:

- * Número del elemento (NUMEL)
- * Número del tipo de material del elemento (MATNU (NUMEL))
- * Lista de los nodos del elemento (LNODS (NUMEL,INODE),
INODE =1, NNODE)

La definición de los nodos del elemento debe seguir siempre un mismo orden, siguiendo una secuencia antihoraria que comience por un nodo esquina cualquiera.

b) *Definición de las coordenadas de los nodos*

Las coordenadas de los nodos se definen siempre con relación a un sistema cartesiano global. La información a leer para cada nodo es la siguiente:

- * Número del nodo (IPNOD)
- * Coordenadas cartesianas del nodo (COORD (IPNOD,IDIME),
IDIME =1, NDIME)

6.3 Condiciones de nodos prescritos

Las condiciones de contorno en los NPRES nodos con movimientos prescritos se leen de acuerdo con la secuencia siguiente:

- * Número del nodo prescrito (NODPR (IPRES))
- * Indicador de los movimientos nodales prescritos (INPRE
(IPRES,IGDLN), IGDLN=1, NGDLN)

- * Valores de los movimientos nodales prescritos (PRESC (IPRES,IGDLN), IGDLN=1, NGDLN)
- * Si INPRE (IPRES,IGDLN)=1 indica que el grado de libertad IGDLN del nodo prescrito IPRES está coaccionado. Por el contrario INPRE=0 denota que dicho grado de libertad está libre. Esto permite coaccionar de manera selectiva los movimientos de cada nodo.

6.4 Propiedades del material

Deben leerse los parámetros del material necesarios para formar la matriz constitutiva **D**. Así, para cada material diferente la información a suministrar es la siguiente:

- * Número del tipo de material (NUMAT)
- * Propiedades del tipo de material en cuestión (PROPS (NUMAT,IPROP), IPROP=1, NPROP)

El orden de lectura de las propiedades del material es el siguiente: Conductividad térmica (tantos valores como dimensiones cartesianas tenga el problema), coeficiente de convección/radiación en fronteras del elemento, valor del calor de generación interna del elemento y en el caso de resolver problemas temporales, se deberá incluir el valor de la densidad del material.

6.5 Condiciones Iniciales de temperatura

En el caso de realizar un análisis transitorio es necesario proporcionar el numero de iteraciones que se deseen, el incremento de tiempo a utilizar y el parámetro α de integración en el tiempo; debido al esquema de iteración temporal implementado se recomienda utilizar incrementos de tiempo adecuados, para no provocar oscilaciones en la integración temporal. La información a suministrar es la siguiente:

- * Número del iteraciones (NTIME)
- * Incremento de tiempo (DTIME)
- * Punto de integración en el intervalo Δt (ALFAT)

A continuación se darán el número de nodos con valores prescritos de las temperaturas iniciales nodales, pudiendo simplificar su captura si todos los nodos de la malla tienen un mismo valor, al introducir (NPNOD) en (NNCIT) y a continuación el valor inicial en (CITIM). Si existen nodos con temperaturas iniciales variables se dará en (NNCIT) el numero de estos y en (CITIM) el valor correspondiente, por lo que el programa asume que los nodos restantes inician con una temperatura de cero.

6.6 Condiciones de flujo de Convección/Radiación en la frontera

Cuando se tenga el caso de elementos que presenten radiación o convección en alguno de sus lados, se deberá proporcionar el numero de elementos existentes en (NNFRON) y el numero de nodos que tienen por lados en (NNFRO) (En el caso de no existir deberá darse el valor de cero a ambas variables). Si fue considerada la situación de tener convección o radiación, la información a suministrar es la siguiente:

Número del elemento con la condición establecida (CONVE(IFRON, 1))

Nodos del lado considerado en numeración global

(CONVE(IFRON,1+INFRO))

6.7 Subrutina GAUSS

La función de esta subrutina es definir las coordenadas y los pesos de la cuadratura de Gauss seleccionada para las integrales del elemento. En elementos unidimensionales, cuadriláteros y hexagonales el orden de la cuadratura utilizada lo define la variable NGAUS, siendo entonces la cuadratura de orden: NGAUS en una dimensión, NGAUS X NGAUS en dos dimensiones y NGAUS X NGAUS X NGAUS en tres dimensiones. Las coordenadas y los pesos de los puntos de integración se almacenan en las variables POSGP(•) y PESPG(•).

En elementos triangulares NGAUS define asimismo el orden de la cuadratura, estando en este caso las coordenadas y los pesos de los puntos de integración asignados a las variables POSGT(•) y PESGT(•).

La subrutina Gauss de la versión de *CALTEP* que se lista en el Apéndice II permite la utilización de cuadraturas con NGAUS=1, 2 y 3 para elementos unidimensionales y cuadriláteros y hexagonales, y NGAUS=1, 3 y 7 para elementos triangulares.

6.8 Preparación automática de datos

En muchos casos el mayor esfuerzo en el análisis de una estructura por elementos finitos se invierte en la preparación de los datos. Por este motivo, es conveniente disponer de medios informáticos auxiliares que permitan automatizar gran parte de las operaciones de la entrada de datos. Ejemplos de esto son las pantallas gráficas, plotters, digitalizadores y copadoras gráficas. Asimismo, es muy útil disponer de programas para generación automática de mallas en una, dos y tres dimensiones y para su representación gráfica [5-7].

Con estos medios se pueden conseguir importantes ahorros en el tiempo de preparación de datos, así como en el de detección de los errores que inevitablemente se producen.

Los programas de generación y dibujo de la malla se incluirían en una hipotética subrutina **GENER** que podría ser llamada por **DATOS**. La descripción de dicha subrutina cae fuera del alcance de este capítulo. Los lectores interesados en este tema pueden encontrar abundante información en las referencias [2], [5-14].

6.9 Subrutina de control de datos

Una vez leídos los datos geométricos y del material de la malla es importante realizar una serie de comprobaciones básicas que garanticen mínimamente la ausencia de errores antes de comenzar el cálculo de las matrices y vectores de cada elemento. Dichas comprobaciones pueden ser tan simples como comprobar que no se han asignado valores absurdos a los parámetros de control; que dichos valores son compatibles con las dimensiones de los vectores de los bloques **COMMON**; que todos los nodos aparecen alguna vez en algún elemento; que no hay dos nodos con el mismo número o mismas coordenadas, etc. Asimismo, puede comprobarse que las dimensiones características de la matriz de rigidez global (ancho de banda [2], ancho de frente [2], perfil [15, 16], etc.) no son superiores a los límites establecidos de acuerdo con el método de solución del sistema de ecuaciones utilizado. Dichas comprobaciones podrían efectuarse en una subrutina auxiliar **COMPROB** que sería llamada por **DATOS** una vez leídos los datos fundamentales de la geometría y propiedades mecánicas de la malla.

Por razones de brevedad no se incluyen aquí subrutinas de control de datos. Dichas subrutinas son fáciles de escribir, aunque siempre deben adecuarse al sistema de generación de datos disponible (manual, automático, etc.). En [2] se pueden encontrar ejemplos de subrutinas de control de datos compatibles con la organización del programa aquí presentado.

7.1 MATRIZ DE RIGIDEZ: SUBROUTINA RIGIMAT

Describiremos brevemente las operaciones para el cálculo de la matriz de rigidez para el caso estacionario y transitorio, incluidos en **RIGIMAT**.

7.1.1 Caso Estacionario

En todos los casos utilizaremos una formulación *isoparamétrica* [4]. Recordemos la expresión general de la matriz de rigidez de un elemento plano [4]:

$$\mathbf{K}_{ij}^{(e)} = \int_{-1}^1 \int_{-1}^1 \mathbf{B}_i^T \mathbf{D} \mathbf{B}_j \left| \mathbf{J}^{(e)} \right| \delta\xi \delta\eta \quad (\text{xxix})$$

En donde $\mathbf{J}^{(e)}$ es la clásica matriz Jacobiano de la transformación de coordenadas cartesianas (x,y) a naturales (ξ,η) .

Los elementos de $\mathbf{K}_{ij}^{(e)}$ se calculan numéricamente. Así, denominando

$$\mathbf{T}_{ij}^{(e)}(\xi, \eta) = \mathbf{B}_i^T \mathbf{D} \mathbf{B}_j \left| \mathbf{J}^{(e)} \right| \quad (\text{xxx})$$

la integración numérica para un *elemento cuadrilátero* con una cuadratura de Gauss de orden $m \times m$ se escribe como

$$\mathbf{K}_{ij}^{(e)} = \sum_{p=1}^m \sum_{q=1}^m \mathbf{T}_{ij}^{(e)}(\xi_p, \eta_q) W_p W_q \quad (\text{xxxii})$$

donde ξ_p y η_q son las coordenadas naturales de los puntos de integración y W_p y W_q los correspondientes pesos [4].

Para un elemento *triangular* con una cuadratura de orden m se tiene

$$\mathbf{K}_{ij}^{(e)} = \sum_{p=1}^m \mathbf{T}_{ij}^{(e)}(\alpha_p, \beta_q) W_p \quad (\text{xxxiii})$$

En el programa se incluyen los elementos lineales isoparamétricos de dos y tres nodos, los elementos cuadriláteros isoparamétricos serendípticos de cuatro y ocho nodos y el lagrangiano de nueve nodos, y los triangulares de tres y seis nodos así como el elemento hexagonal isoparamétrico serendípito de veinte nodos (Figura 6).

En todos los elementos cuadriláteros se recomienda una cuadratura 2×2 . Por otra parte, los elementos triangulares lineal y cuadrático precisan cuadraturas de uno y tres puntos, respectivamente [4]. Mientras que para el elemento hexagonal isoparamétrico serendípito de veinte nodos cuya matriz de rigidez se presenta a continuación, se recomienda evaluar con una cuadratura de orden $2 \times 2 \times 2$ por

$$\mathbf{K}_{ij}^{(e)} = \sum_{p=1}^2 \sum_{q=1}^2 \sum_{r=1}^2 \mathbf{T}_{ij}^{(e)}(\xi_p, \eta_q, \zeta_r) W_p W_q W_r \quad (\text{xxxiiii})$$

En el caso de los elementos lineales, una cuadratura de orden 2 es suficiente.

7.1.2 Caso Transitorio

Al realizar la integración en el tiempo es necesario incluir en la matriz de rigidez elemental, la aportación de una matriz de masa elemental que se evalúa de acuerdo con:

$$\mathbf{M}_{ij}^{(e)} = \int_{-1}^1 \int_{-1}^1 \rho c \mathbf{N}_i^T \mathbf{N}_j \left| \mathbf{J}^{(e)} \right| \delta\xi \delta\eta \quad (\text{xxxv})$$

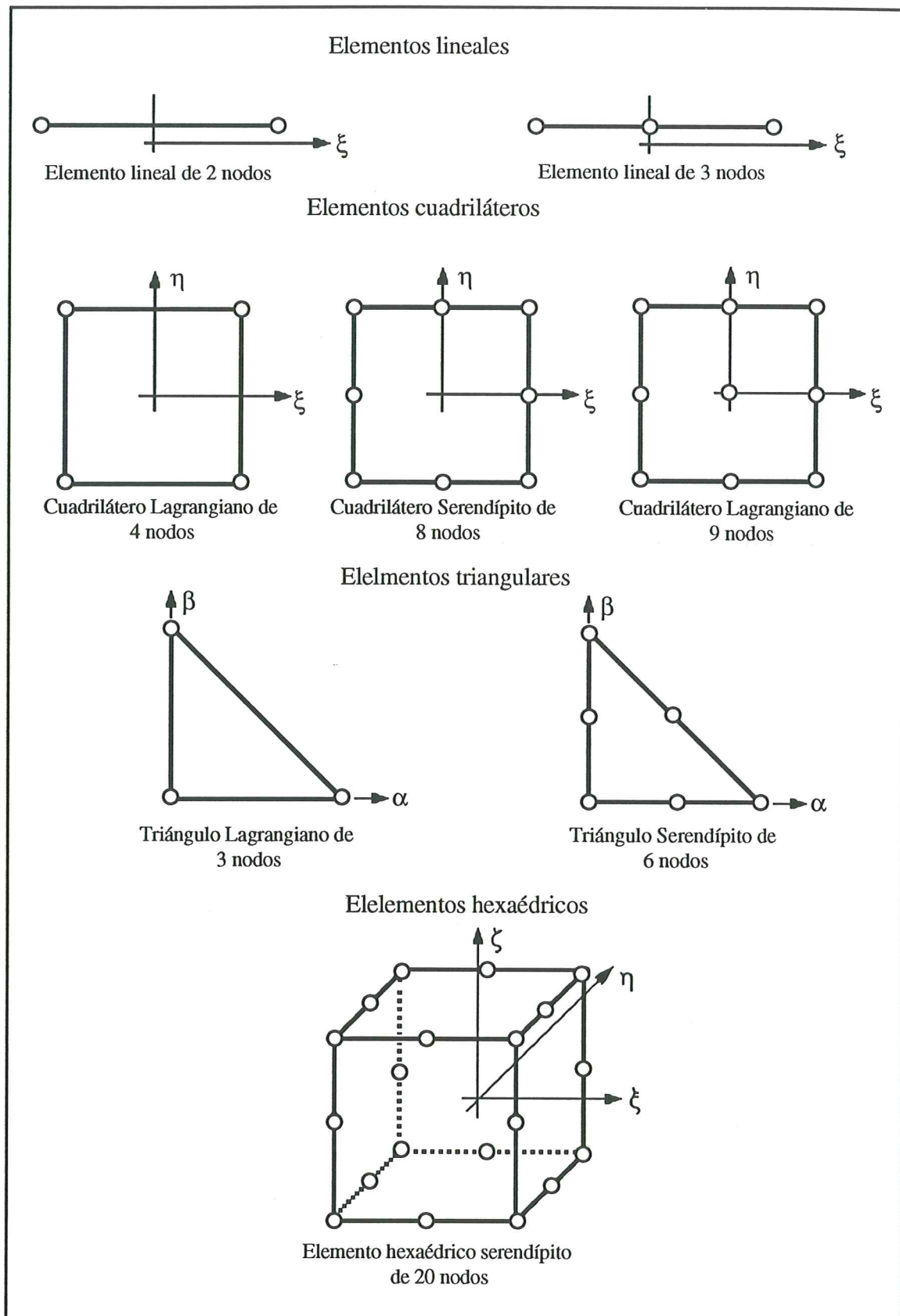


Figura 6 Elementos incluidos en el programa CALTEP

como se encontró en (xviii) y de ahí se concluye que el sistema a resolver para un paso de tiempo es el propuesto en (xxii), por lo que la matriz de rigidez encontrada en el apartado 7.1.1, se incrementa termino a termino con la encontrada en la ecuación (xxxiv), siendo afectada la matriz $\mathbf{K}^{(e)}$ por el coeficiente α de integración en el tiempo.. Las ventajas de este esquema de integración temporal, además de su sencillez de implementación es que presenta una estabilidad incondicional ante los valores de α mayores a 0.5 y para α igual a cero no es necesario invertir la matriz de rigidez.

Es obvio que en el caso transitorio, también se verán afectados los términos de carga según se muestra en la ecuación (xxii).

En la Figura 7 se presenta el diagrama de flujo de **RIGIMAT**. Obsérvese que se han supuesto propiedades del material constantes sobre el elemento y el cálculo de la matriz constitutiva **D** se efectúa antes del bucle sobre los puntos de integración. En el caso de propiedades variables bastaría con incluir la evaluación de **D** dentro de dicho bucle.

El contenido de las diferentes subrutinas que intervienen en **RIGIMAT** puede estudiarse con detalle en el listado que se presenta en el Apéndice II.

7.2 MATRIZ DE RIGIDEZ: SUBROUTINA CONVECC

La aportación a la matriz de rigidez por efectos de convección y radiación, expresado por el tercer elemento del primer término de la ecuación (xi), debe de ser considerada en todos los elementos de contorno donde esté presente dicho fenómeno.

En el caso de un elemento bidimensional isoparamétrico, la expresión general es:

$$\mathbf{K}_{ij}^{(\alpha e)} = \int_{-1}^1 \mathbf{N}_i^T \mathbf{N}_j \alpha |\mathbf{J}^{(e)}| d\xi \quad (\text{xxxv})$$

Debemos observar que la integración es unidimensional, ya que la pérdida de calor por convección y radiación se produce sobre los lados del elemento y no sobre todo el elemento en si, por lo que la matriz jacobiano de transformación de coordenadas deberá corresponder a la del lado en cuestión, siendo la integración sobre esa línea; por lo tanto es importante resaltar que la matriz $\mathbf{K}^{(\alpha e)}$ contendrá elementos nulos con excepción de aquellos $\mathbf{K}_{ij}^{(\alpha e)}$ que correspondan a los del lado con la condición de contorno impuesta. Para mayor claridad del concepto, se puede observar la figura 8.

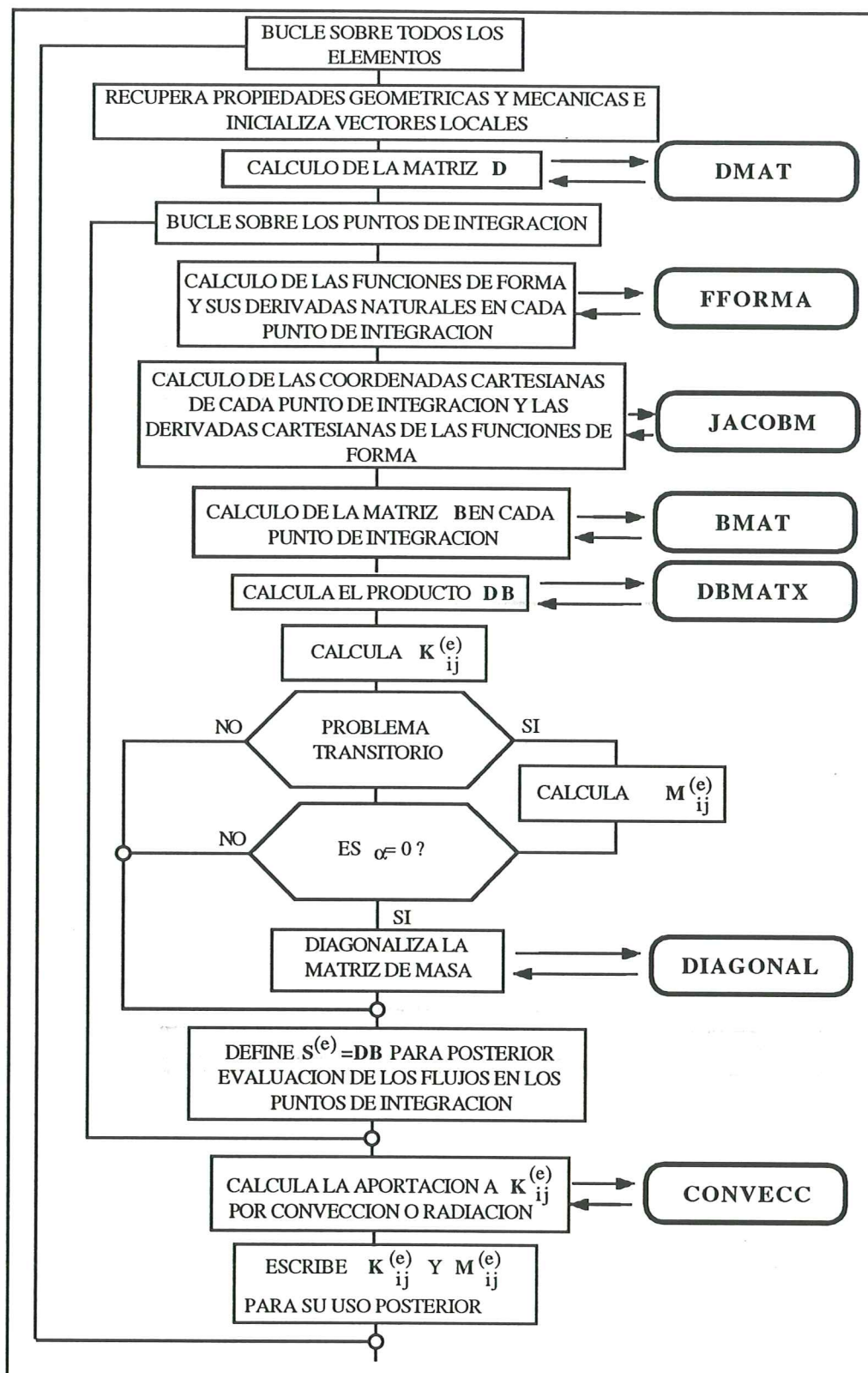


Figura 7 Diagrama de flujo de la subrutina RIGIMAT

Análogamente, para el caso tridimensional, la frontera de convección y radiación corresponderá a una superficie, por lo que la expresión es:

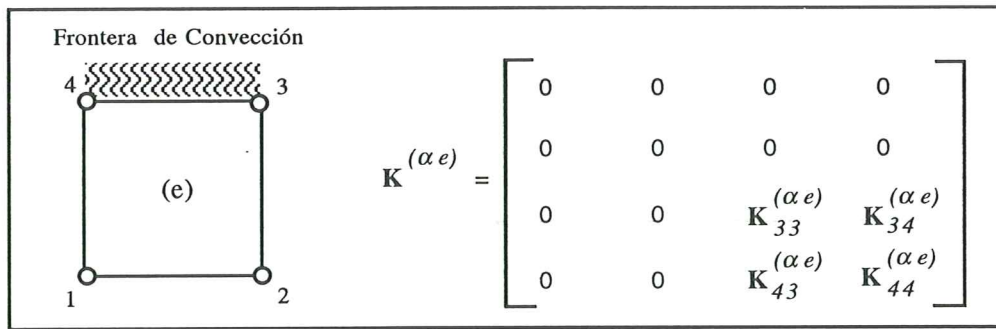


Figura 8 Aportación correspondiente por Convección y Radiación

$$K_{ij}^{(\alpha e)} = \int_{-1}^1 \int_{-1}^1 N_i^T N_j \alpha |J^{(e)}| \partial \xi \partial \eta \tag{xxxvi}$$

correspondiendo a una integral de superficie.

En el caso unidimensional, se considera que existe el fenómeno sobre la superficie perimetral del elemento de línea y no en la sección transversal de este, siendo la expresión idéntica a la ecuación (xxxv) y en este caso la matriz $K^{(\alpha e)}$ será llena.

Al tener de esta manera, en todos los casos vistos, una matriz $K^{(\alpha e)}$ de igual dimensión que la matriz de rigidez, la matriz de rigidez del elemento será la adición de ambas.

$$K_{ij}^{(e)} = K_{ij}^{(e)} + K_{ij}^{(\alpha e)} \tag{xxxvii}$$

8 SUBROUTINA FUERZAS

8.1 Consideraciones generales.

Presentaremos seguidamente la subrutina de cálculo de temperaturas nodales equivalentes para los diferentes elementos considerados. Se tendrán en cuenta los siguientes tipos de acciones exteriores:

- * *Flujos puntuales nodales* (IPUNT \neq 0).
- * *Generación Interna* (IPESO \neq 0).
- * *Flujo y Temperatura Externa sobre el elemento* (IDIST \neq 0).

La actuación de los distintos tipos de temperaturas se controla asignando un valor diferente de cero a los parámetros de control IPUNT, IPESO e ILADO.

En la Figura 9 se presenta el diagrama de flujo de la subrutina FUERZAS. En los apartados siguientes se detallan los módulos de temperaturas puntuales, Generación Interna y Flujo y Temperatura Externa sobre el elemento.

8.2 flujos puntuales nodales

Para mayor sencillez consideraremos que los flujos puntuales actúan directamente sobre un nodo. El vector de flujos nodales es en este caso simplemente

$$\mathbf{f}_i = \mathbf{p}_i \quad (\text{xxxviii})$$

siendo \mathbf{p}_i el vector de flujos puntuales sobre el nodo de numeración global i . Aunque este termino no se incluye explícitamente en la ecuación (xx) el vector \mathbf{f} corresponde directamente al flujo nodal equivalente por lo que la aplicación de un flujo sobre un nodo en particular queda representada directamente por un vector \mathbf{f} , particularizado en la ecuación (xvii).

8.3 Generación interna

La generación interna de calor equivale a una temperatura "máscica" actuando por unidad de superficie/volumen. Así pues, las temperaturas por generación interna para los diferentes elementos de CALTEP se obtienen por [4], que de acuerdo con la formulación descrita, y aplicando las transformaciones isoparamétricas correspondientes, se tiene que el primer termino del segundo miembro de la ecuación (xx) se expresa como:

Elementos unidimensionales

$$\mathbf{f}_i^{(e)} = \int_{l^{(e)}} \mathbf{N}_i^T \rho r \partial l^{(e)} = \sum_{p=1}^{n_p} \left(\mathbf{N}_i^T \rho r \left| \mathbf{J}^{(e)} \right| \right)_p W_p \quad (\text{xxxix})$$

Elementos bidimensionales

$$\mathbf{f}_i^{(e)} = \int_{A^{(e)}} \mathbf{N}_i^T \rho r \partial A^{(e)} = \sum_{p=1}^{n_p} \sum_{q=1}^{n_q} \left(\mathbf{N}_i^T \rho r \left| \mathbf{J}^{(e)} \right| \right)_{p,q} W_p W_q \quad (\text{xl})$$

Elementos de sólido tridimensional

$$\mathbf{f}_i^{(e)} = \int \int \int_{V^{(e)}} \mathbf{N}_i^T \rho r \partial V^{(e)} = \sum_{p=1}^{n_p} \sum_{q=1}^{n_q} \sum_{r=1}^{n_r} \left(\mathbf{N}_i^T \rho r \left| \mathbf{J}^{(e)} \right| \right)_{p,q,r} W_p W_q W_r \quad (\text{xli})$$

La cuadratura para el calculo de estos términos, suele ser la misma que para el de la matriz de rigidez, aún a costa de introducir un cierto error en el cálculo. No obstante, este

error es de poca importancia y suele compensarse con los errores en la evaluación de la matriz de rigidez.

8.4 Temperatura ambiental y flujo Exterior repartido sobre un lado

Sobre los lados de los elementos limítrofes de la geometría, es común que exista una temperatura externa diferente que coaccione el comportamiento del flujo de temperaturas en el interior de la malla, o bien que exista un flujo de calor entrante o saliente, normal a la superficie del lado del elemento; en análisis bidimensional pueden actuar un flujo repartido por *unidad de longitud* en direcciones normal y tangencial, (aunque en realidad el flujo tangencial no afecta la solución de la ecuación diferencial y se desprecia). Estas cargas de calor no tienen porqué ser uniformes y su intensidad puede variar a lo largo del lado. Dicha variación se define por los valores de las temperaturas en los nodos del lado cargado.

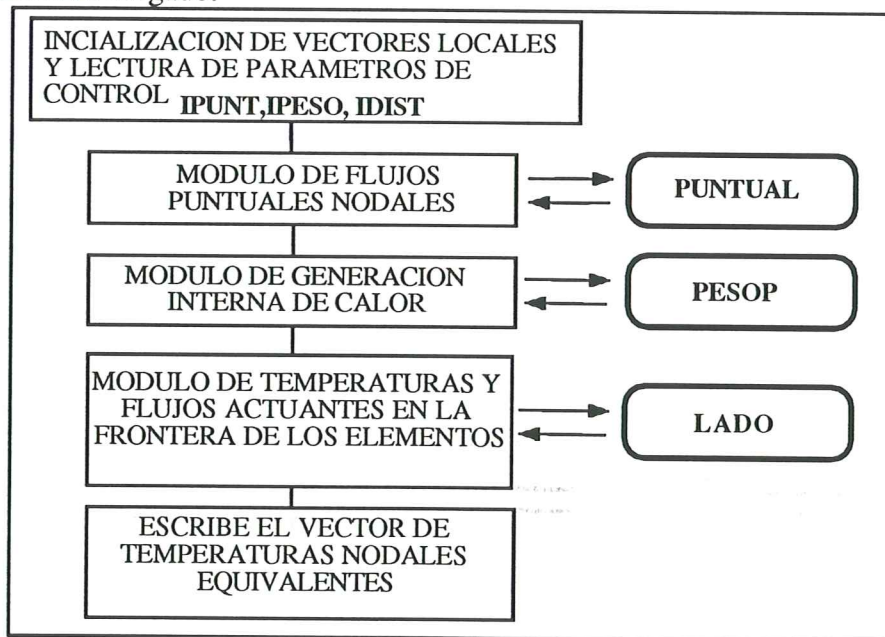


Figura 9 Diagrama de flujo de la subrutina FUERZAS para el cálculo de las temperaturas nodales equivalentes

Para ser coherente con el orden de numeración de las conexiones nodales, los nodos del lado cargado deben listarse en una secuencia antihoraria.

Un flujo normal a un lado se considerará positiva si va dirigida *hacia el interior* del elemento. Esta definición es necesaria para evitar confusiones cuando los flujos distribuidos actúan sobre lados comunes a dos elementos, siendo importante advertir que en dicho caso la carga generada sólo debe asignarse a un solo elemento.

El vector de las temperaturas equivalentes nodales se obtiene como se explica con detalle en la referencia [4]. La expresión final de dicho vector es la combinación de los elementos segundo y tercero de la ecuación (xx) que expresado para el caso unidimensional con la formulación isoparamétrica y la integración numérica se describe como:

$$\mathbf{f}_i^{(e)} = \oint_{\Gamma_q^{(e)}} \mathbf{N}_i^T [\alpha \phi_{ext} - \bar{q}] \partial \Gamma_q^{(e)} = \sum_{p=1}^{n_p} \left(\mathbf{N}_i^T [\alpha \phi_{ext} - \bar{q}] \big| \mathbf{J}^{(e)} \big| \right)_p W_p \quad (\text{xlii})$$

que de manera similar puede ser expresada para los dominios bi y tridimensionales.

9 SOLUCION DEL SISTEMA DE ECUACIONES. SUBRUTINA SOLUCION

Existen muchas técnicas para resolver el sistema de ecuaciones $\mathbf{Ka} = \mathbf{f}$ resultante del ensamblaje de las ecuaciones de equilibrio de los diferentes elementos. Entre los procedimientos más populares podemos citar los:

Métodos directos, como

- * Eliminación Gaussiana [2], [3], [11], [17]
- * Método frontal [2], [18]
- * Reducción de Choleski (o Choleski modificado) [3, 17]
- * Reducción de Crout [3, 17]
- * Método del perfil [15, 16], etc.

Métodos iterativos, como

- * Método iterativo de Gauss-Seidel [3, 17]
- * Métodos de gradiente conjugado [3, 17]
- * Métodos de relajación [3, 17]

La descripción detallada de estos métodos puede encontrarse en la mayoría de libros de cálculo numérico, y en particular en las referencias citadas. Aquí emplearemos el método de eliminación Gaussiana, por ser quizás el más sencillo y fácil de implementar en un programa de elementos finitos. El método de eliminación Gaussiana se basa en la reducción del sistema de ecuaciones $\mathbf{Ka} = \mathbf{f}$ a la siguiente forma triangular

$$\begin{aligned} K_{1,1}a_1 + K_{1,2}a_2 + K_{1,3}a_3 + \dots + K_{1,n-1}a_{n-1} + K_{1,n}a_n &= f_1 \\ K'_{2,2}a_2 + K'_{2,3}a_3 + \dots + K'_{2,n-1}a_{n-1} + K'_{2,n}a_n &= f'_2 \\ K'_{3,3}a_3 + \dots + K'_{3,n-1}a_{n-1} + K'_{3,n}a_n &= f'_3 \\ &\dots \\ &\dots \end{aligned}$$

$$\begin{aligned} K'_{n-1,n-1}a_{n-1} + K'_{n-1,n}a_n &= f'_{n-1} \\ K'_{n,n}a_n &= f'_n \end{aligned} \quad (\text{xliii})$$

donde las primas indican que los coeficientes de la matriz de rigidez global y los términos de temperatura se han modificado durante la etapa de reducción.

El sistema (xliv) permite calcular sistemáticamente todas las incógnitas resolviendo las ecuaciones *reducidas* en orden inverso desde la última hasta la primera, ya que cada nueva ecuación sólo introduce una nueva incógnita. Así, la última ecuación proporciona directamente el valor de a_n ; sustituyendo dicho valor en la ecuación anterior puede obtenerse el valor de a_{n-1} y así sucesivamente. Esta etapa de la solución se denomina *sustitución hacia atrás*.

Por consiguiente, las tres etapas características del método de eliminación Gaussiana son las siguientes:

1) *Etapa de ensamblaje*. Consiste en ensamblar la matriz de rigidez global K y el vector de temperaturas nodales f a partir de las contribuciones de los diferentes elementos de la malla.

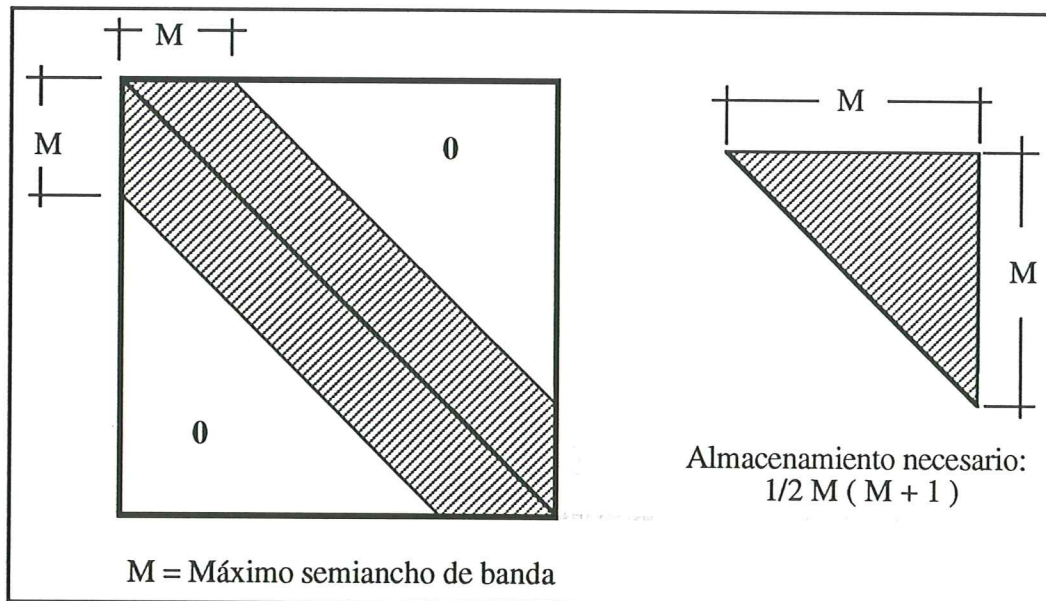


Figura 10 Matriz de rigidez en banda.

2) *Etapa de reducción*. Consiste en reducir el sistema de ecuaciones original $Ka = f$ a la forma (xlv). Esto se efectúa empleando la ecuación i -ésima para eliminar la variable a_j de todas las ecuaciones inferiores, es decir, de la ecuación $i + 1$ a la n . Formalmente esto puede efectuarse sustrayendo de la ecuación r -ésima ($i < r \leq n$) la ecuación i -ésima multiplicada por $(i)K_{ri} / (i)K_{ii}$, donde el índice i indica que dichos coeficientes se han modificado $i - 1$ veces antes de la eliminación de a_j . Por ejemplo, la primera ecuación se utiliza para eliminar a_1 de las ecuaciones 2 a n , como :

$$K_{1,1}a_1 + K_{1,2}a_2 + K_{1,3}a_3 + \dots + K_{1,n}a_n = f_1$$

$$0a_1 + \left(K_{2,2} - K_{1,2} \frac{K_{2,1}}{K_{1,1}} \right) a_2 + \left(K_{2,3} - K_{1,3} \frac{K_{2,1}}{K_{1,1}} \right) a_3 + \dots +$$

$$\left(K_{2,n} - K_{1,n} \frac{K_{2,1}}{K_{1,1}} \right) a_n = f_2 - f_1 \frac{K_{2,1}}{K_{1,1}} = f_2'$$

.....

$$0a_1 + \left(K_{n,2} - K_{1,2} \frac{K_{n,1}}{K_{1,1}} \right) a_2 + \left(K_{n,3} - K_{1,3} \frac{K_{n,1}}{K_{1,1}} \right) a_3 + \dots +$$

$$\left(K_{n,n} - K_{1,n} \frac{K_{n,1}}{K_{1,1}} \right) a_n = f_n - f_1 \frac{K_{n,1}}{K_{1,1}} = f_n'$$

(xlv)

A continuación se utiliza la ecuación 2 para eliminar a_2 de todas las ecuaciones y así sucesivamente. Adviértase que se mantiene la simetría en el sistema de ecuaciones modificado.

Si un movimiento está prescrito, por ejemplo $a_2 = \bar{a}_2$, la incógnita pasa a ser la reacción correspondiente t_2 . En este caso, la eliminación de a_2 es trivial y todo lo que hay que hacer es sustituir $a_2 = \bar{a}_2$ en todas las ecuaciones y transferir la cantidad conocida $K'_{r,2} \bar{a}_2$ ($3 \leq r \leq n$) al segundo miembro de cada ecuación, como se ilustra seguidamente

$$K_{1,1}a_1 + K_{1,2} \bar{a}_2 + K_{1,3}a_3 + \dots + K_{1,n}a_n = f_1$$

$$0a_1 + K'_{2,2} \bar{a}_2 + K'_{2,3}a_3 + \dots + K'_{2,n}a_n = f_2'$$

$$0a_1 + 0a_2 + K'_{3,3}a_3 + \dots + K'_{3,n}a_n = f_3' - K'_{3,2} \bar{a}_2$$

.....

$$0a_1 + 0a_2 + K'_{n,3}a_3 + \dots + K'_{n,n}a_n = f_n' - K'_{n,2} \bar{a}_2$$

(xlv)

Si el movimiento está prescrito a un valor nulo es más sencillo prescindir directamente de la fila y columna correspondientes a dicho movimiento y calcular la reacción a posteriori una vez resuelto el sistema.

En el proceso de reducción puede hacerse uso de las ventajas de la *simetría* de la matriz de rigidez global. Ello permite reducir sólo los términos sobre y por arriba de la diagonal principal de \mathbf{K} , obteniéndose el resto por simetría, con el consiguiente ahorro de cálculo.

En la mayor parte de los problemas estructurales \mathbf{K} tiene forma de banda, con ceros fuera de una banda simétrica con respecto a la diagonal principal (Figura 10). Dicha forma en banda permite introducir considerables economías en el proceso de solución del sistema, ya que en cada instante sólo es necesario almacenar la matriz triangular superior de una submatriz de dimensiones $M \times M$, siendo M el semi-ancho de banda. Por consiguiente, los requisitos de almacenaje son de $1/2M(M+1)$ números. El valor de M depende de cómo se numeran los nodos en la malla. Puede demostrarse que $M=(D+1)*NDGDL$ siendo D la máxima diferencia entre los números de dos nodos pertenecientes a cualquiera de los elementos de la malla [2].

Lo anterior evidencia la importancia de escoger una numeración nodal tal que el valor de M sea mínimo. Esto en general no es sencillo, sobre todo para mallas grandes, y en estos casos suele acudir a programas que automáticamente proporcionan una numeración nodal óptima. Una alternativa es utilizar otros métodos de solución en los que el orden de numeración nodal sea irrelevante. Entre éstos se encuentra el método **FRONTAL** en el que la capacidad de almacenamiento necesaria viene condicionada por la numeración de los elementos y no de los nodos, lo que facilita la etapa de preproceso. El método **FRONTAL** es una modificación del de eliminación Gaussiana y se basa en realizar el ensamblaje de las ecuaciones y la eliminación de las variables de manera simultánea. Así, una vez ensamblada la matriz de rigidez de un nuevo elemento, se eliminan las variables nodales que no se verán afectadas por posteriores ensamblajes, con lo que el espacio asignado a los coeficientes de rigidez de dichas variables puede ser ocupado por el de nuevas variables al ensamblar las ecuaciones, de otro elemento, y así sucesivamente. Esto reduce considerablemente los requisitos de almacenaje y el número de operaciones aritméticas, a costa, sin embargo, de incrementar sensiblemente la complejidad del algoritmo de solución. Una descripción detallada del método **FRONTAL** puede encontrarse en la referencia [2].

En el Apéndice II se presenta el listado de las subrutinas **SOLUCION**, **ENSAMBLA**, **REDUCE** y **SUSTITUIR** para resolver el sistema de ecuaciones utilizando el método de eliminación Gaussiana explicado. Como nota característica se destaca que en dichas subrutinas los términos de la matriz de rigidez se almacenan en un vector siguiendo una secuencia a lo largo de la banda, prescindiéndose de los coeficientes nulos situados fuera de la misma, con la consiguiente economía de memoria.

El método de solución seguido para el problema de integración en tiempo, es esencialmente el mismo, con la salvedad de que en la parte triangular inferior de la matriz de rigidez, que el sistema de reducción gaussiana convierte en ceros, se almacenan los valores de los elementos K'_{ij}/K'_{ii} de manera que para resolver el sistema en todas las

iteraciones temporales, no será necesario resolver nuevamente la matriz de rigidez, ya que esta es constante en todo el proceso, y esta información almacenada servirá para reducir el vector de temperaturas nodales, con el consecuente ahorro de tiempo de computo.

10 CALCULO DE LOS FLUJOS ELEMENTALES. SUBROUTINA TENSION

Como es usual los flujos se calculan inicialmente en los puntos de integración y a partir de dichos valores se procede a su extrapolación a los nodos y al subsecuente alisado nodal si se desea [4]. Así, las tensiones en el punto de Gauss p se obtienen por

$$\mathbf{q}_p = \sum_{i=1}^n (\mathbf{DB}_i)_p \mathbf{a}_i \quad (\text{xlvi})$$

Recordemos que durante el cálculo de la matriz de rigidez en **RIGIMAT** se evalúa el producto **DB** en cada punto de Gauss, almacenándose la matriz resultante en un fichero, juntamente con las coordenadas cartesianas de los puntos de Gauss. Por consiguiente, para efectuar los productos de (xlvi) basta con leer para cada elemento la información almacenada en dicho fichero, evitándose así repetir el cálculo de las matrices **B** y **D** en cada punto de Gauss.

En el Apéndice II se presenta el listado de la subrutina **TENSIONES** donde se efectúan esas operaciones.

Se incluye en **CALTEP**, una subrutina de extrapolación de los valores de los flujos de los valores de los flujos en los puntos de Gauss a los nodos siguiendo la técnica de extrapolación y alisado global descritas en la referencia [4].

11 EJEMPLOS DE UTILIZACION DEL PROGRAMA CALTEP

A efectos meramente ilustrativos se presentan dos sencillos ejemplos de aplicación del programa *CALTEP* a los problemas siguientes [11]:

- Flujo de calor a través de una barra delgada con frontera de convección y radiación, analizada con dos elementos unidimensionales.
- Generación de calor en un dominio cuadrado, caso estacionario, transitorio con $\alpha = 1$ y transitorio con $\alpha=2/3$; analizado con una malla de 5 X 5 elementos triangulares de 3 nodos.

Para cada problema analizado se presentan los listados de los datos y de la salida de resultados. Los flujos nodales que se listan se han obtenido mediante la técnica de extrapolación y alisado global, utilizando un campo *lineal* de flujos. Esto proporciona directamente los valores alisados de las tensiones en los nodos esquina de cada elemento.

11.1 Flujo a través de una barra delgada.

Dada la geometría del problema y características del problema, su análisis es muy sencillo como se observa en la figura 11, por lo que pasaremos directamente a la codificación del fichero de entrada:

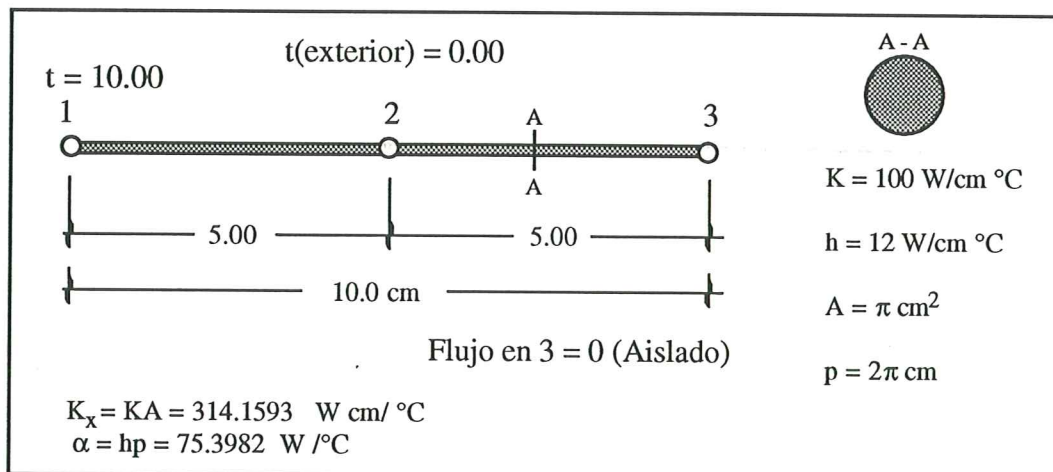


Figura 11 Geometría y discretización del problema del flujo de calor a través de una barra delgada.

LISTADO DE DATOS (BARRA DELGADA)

```

FLUJO DE CALOR EN BARRA DELGADA (LINEAL 2 NODOS)
3  2  1  0  2  1  2  1  1
1  1  1  2
2  1  2  3
1  0.00000
2  5.00000
3  10.00000
1  1  10.0
1  314.1593  75.3982  0.0
2  2
1  1  2
2  2  3
sin condiciones de carga
0  0  0

```

LISTADO DE RESULTADOS (BARRA DELGADA)

NUMERO DE PROBLEMAS= 1

PROBLEMA NO. 1

FLUJO DE CALOR EN BARRA DELGADA (LINEAL 2 NODOS)

```

NPNOD = 3  NELEM = 2  NPRES = 1  NTRAN = 0  *NTIPO = 1
NNODE = 2  *NGDLN = 1  NMATS = 1  *NPROP = 3  NGAUS = 2
NDIME = 1  *NTENS = 1  *NEVAB = 2  IWRTIT = 1

```

```

ELEMENTO  PROPIEDAD  NUMERO DE NODOS
1  1  1  2
2  1  2  3

```

COORDENADAS DE PUNTOS NODALES

```

NODO  X  Y  Z
1  0.000
2  5.000
3  10.000

```

NODOS RESTRINGIDOS Y PRESCRIPCIONES

```

NODO CODIGO  VALORES PRESCRITOS
1  1  10.00000

```

PROPIEDADES DE LOS MATERIALES

```

NUMERO  PROPIEDADES
1  K= 0.31E+03
  ALPHA= 0.75E+02  ROr= 0.00E+00  ROc=
FRONTERAS CON FLUJO DE CONVECCION/RADIACION = 2

```

NUMERO DE NODOS POR FRONTERA = 2

```

ELEMENTO  N O D O S
1  1  2
2  2  3

```

sin condiciones de carga

```

COND. DE CARGA PUNTUAL:..... 0
COND. DE CARGA DE GENER. INTERNA:.... 0
COND. DE CALOR DISTRIBUIDO POR LADO.. 0

```

FUERZAS TOTALES NODALES EQUIVALENTES ELEMENTALES

```
1      0.0000E+00  0.0000E+00
2      0.0000E+00  0.0000E+00
```

TEMPERATURAS

```
NODO  TEMPERATURA
  1    0.100000E+02
  2    0.607128E-06
  3    0.737209E-13
```

REACCIONES

```
NODO      REACCION
  1    0.188496E+04
```

TENSIONES

G.P. X-COORD X-FLUJ.

```
ELEMENTO NO.= 1
1      1.0566 -628.3185
2      3.9434 -628.3185
```

```
ELEMENTO NO.= 2
1      6.0566  0.0000
2      8.9434  0.0000
```

FLUJOS ALISADAS NODALES

```
NODO X-FLUJ.      Y-FLUJ.      Z-FLUJ.
1-0.62832E+03
2-0.31416E+03
3-0.38147E-04
```

11.2 Generación interna de calor en un dominio cuadrado.

11.2.1 Caso estacionario

La geometría del problema en este caso nos permite analizar un cuarto del dominio, debido a la simetría que guarda con los ejes XX e YY aplicando las características mostradas en la figura 12, tanto en las condiciones de contorno como en la malla utilizada. El caso que se resuelve es el estacionario; se muestra a continuación los datos del programa

11.2.1.1 LISTADO DE DATOS (GENERACION DE CALOR EN DOMINIO CUADRADO) CASO ESTACIONARIO.

GENERACION DE CALOR EN DOM. CUADRADO (TRIANGULAR 3 NODOS)

```
36  50  11  0  3  1  3  2  1
1   1   1   2   8
2   1   2   3   9
3   1   3   4  10
4   1   4   5  11
5   1   5   6  12
6   1   8   7   1
7   1   9   8   2
```

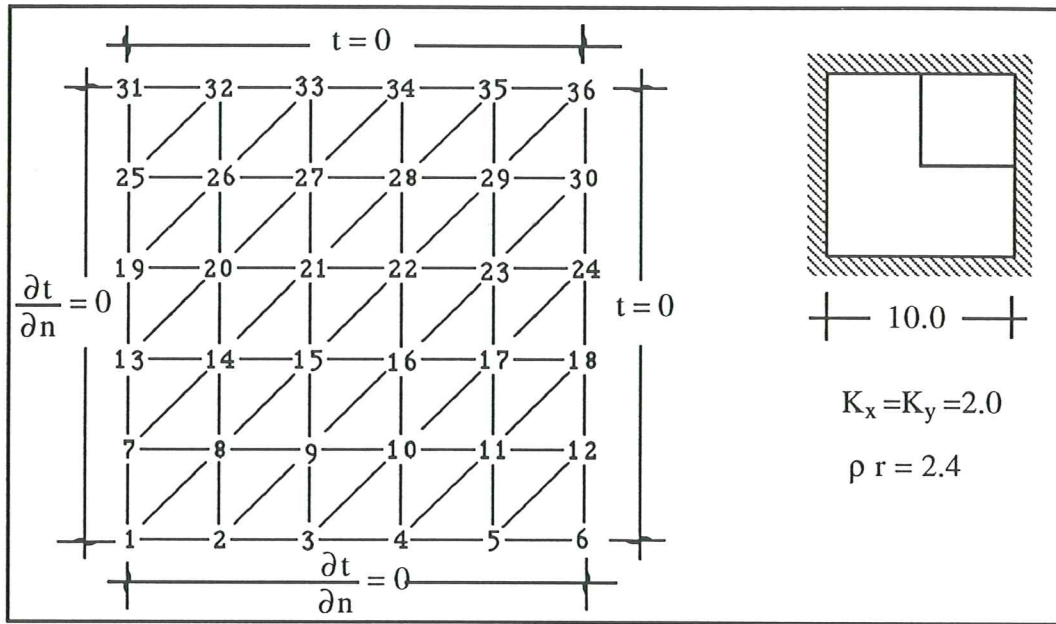


Figura 12 Geometría y discretización del problema de generación interna de calor en un dominio cuadrado.

8	1	10	9	3
9	1	11	10	4
10	1	12	11	5
11	1	7	8	14
12	1	8	9	15
13	1	9	10	16
14	1	10	11	17
15	1	11	12	18
16	1	14	13	7
17	1	15	14	8
18	1	16	15	9
19	1	17	16	10
20	1	18	17	11
21	1	13	14	20
22	1	14	15	21
23	1	15	16	22
24	1	16	17	23
25	1	17	18	24
26	1	20	19	13
27	1	21	20	14
28	1	22	21	15
29	1	23	22	16
30	1	24	23	17
31	1	19	20	26
32	1	20	21	27
33	1	21	22	28
34	1	22	23	29
35	1	23	24	30
36	1	26	25	19
37	1	27	26	20
38	1	28	27	21
39	1	29	28	22
40	1	30	29	23
41	1	25	26	32
42	1	26	27	33
43	1	27	28	34
44	1	28	29	35
45	1	29	30	36
46	1	32	31	25

```

47  1  33  32  26
48  1  34  33  27
49  1  35  34  28
50  1  36  35  29
  1  0.00000  0.00000
  2  1.00000  0.00000
  3  2.00000  0.00000
  4  3.00000  0.00000
  5  4.00000  0.00000
  6  5.00000  0.00000
  7  0.00000  1.00000
  8  1.00000  1.00000
  9  2.00000  1.00000
10  3.00000  1.00000
11  4.00000  1.00000
12  5.00000  1.00000
13  0.00000  2.00000
14  1.00000  2.00000
15  2.00000  2.00000
16  3.00000  2.00000
17  4.00000  2.00000
18  5.00000  2.00000
19  0.00000  3.00000
20  1.00000  3.00000
21  2.00000  3.00000
22  3.00000  3.00000
23  4.00000  3.00000
24  5.00000  3.00000
25  0.00000  4.00000
26  1.00000  4.00000
27  2.00000  4.00000
28  3.00000  4.00000
29  4.00000  4.00000
30  5.00000  4.00000
31  0.00000  5.00000
32  1.00000  5.00000
33  2.00000  5.00000
34  3.00000  5.00000
35  4.00000  5.00000
36  5.00000  5.00000
  6  1
12  1
18  1
24  1
30  1
31  1
32  1
33  1
34  1
35  1
36  1
  1  2.0e+00  2.0  1.0  2.4
  0  0
  generación interna de calor
  0  1  0

```

11.2.1.2 LISTADO DE RESULTADOS (GENERACION DE CALOR EN DOMINIO CUADRADO) CASO ESTACIONARIO.

PROBLEMA NO. 1

HEAT GENERATION IN A SQUARE DOMAIN (TRIANGULAR 3 NODOS)

NPNOD = 36 NELEM = 50 NPRES = 11 NTRAN = 0 *NTIPO = 1
 NNODE = 3 *NGDLN = 1 NMATS = 1 *NPROP = 4 NGAUS = 3
 NDIME = 2 *NTENS = 2 *NEVAB = 3 IWRIT = 1

ELEMENTO	PROPIEDAD	NUMERO DE NODOS		
1	1	1	2	8
2	1	2	3	9
3	1	3	4	10
.
48	1	34	33	27
49	1	35	34	28
50	1	36	35	29

COORDENADAS DE PUNTOS NODALES

NODO	X	Y	Z
1	0.000	0.000	
2	1.000	0.000	
3	2.000	0.000	
.	.	.	.
34	3.000	5.000	
35	4.000	5.000	
36	5.000	5.000	

NODOS RESTRINGIDOS Y PRESCRIPCIONES

NODO	CODIGO	VALORES PRESCRITOS
6	1	0.00000
12	1	0.00000
18	1	0.00000
24	1	0.00000
30	1	0.00000
31	1	0.00000
32	1	0.00000
33	1	0.00000
34	1	0.00000
35	1	0.00000
36	1	0.00000

PROPIEDADES DE LOS MATERIALES

NUMERO	PROPIEDADES			
1	K=	0.20E+01	K=	0.20E+01
	ALPHA=	0.10E+01	ROR=	0.24E+01 ROc=
FRONTERAS CON FLUJO DE CONVECCION/RADIACION = 0				
NUMERO DE NODOS POR FRONTERA = 0				

generación interna

COND. DE CARGA PUNTUAL:.....	0
COND. DE CARGA DE GENER. INTERNA:....	1
COND. DE CALOR DISTRIBUIDO POR LADO..	0

FUERZAS TOTALES NODALES EQUIVALENTES ELEMENTALES

1	0.4000E+00	0.4000E+00	0.4000E+00
2	0.4000E+00	0.4000E+00	0.4000E+00
3	0.4000E+00	0.4000E+00	0.4000E+00

.

.

.

48	0.4000E+00	0.4000E+00	0.4000E+00
49	0.4000E+00	0.4000E+00	0.4000E+00
50	0.4000E+00	0.4000E+00	0.4000E+00

TEMPERATURAS

NODO	TEMPERATURA
1	0.898176E+01
2	0.858176E+01
3	0.761063E+01
4	0.595499E+01
5	0.347666E+01
6	0.000000E+00
7	0.858176E+01
8	0.826732E+01
9	0.735289E+01
10	0.576634E+01
11	0.337582E+01
12	0.000000E+00
13	0.761063E+01
14	0.735289E+01
15	0.656727E+01
16	0.518164E+01
17	0.306029E+01
18	0.000000E+00
19	0.595499E+01
20	0.576634E+01
21	0.518164E+01
22	0.413267E+01
23	0.248370E+01
24	0.000000E+00
25	0.347666E+01
26	0.337582E+01
27	0.306029E+01
28	0.248370E+01
29	0.154185E+01
30	0.000000E+00
31	0.000000E+00
32	0.000000E+00
33	0.000000E+00
34	0.000000E+00
35	0.000000E+00
36	0.000000E+00

0 REACCIONES

0	NODO	REACCION
	6	-0.387666E+01
	12	-0.795164E+01
	18	-0.732058E+01
	24	-0.616741E+01
	30	-0.428370E+01
	31	-0.387666E+01
	32	-0.795164E+01
	33	-0.732058E+01
	34	-0.616741E+01
	35	-0.428370E+01
	36	-0.800000E+00

TENSIONES

G.P. X-COORD Y-COORD X-FLUJ. Y-FLUJ.

ELEMENTO NO.=	1
1	0.3333 0.1667-0.8000E+00-0.62887E+00
2	0.8333 0.1667-0.8000E+00-0.62887E+00
3	0.8333 0.6667-0.8000E+00-0.62887E+00

```

ELEMENTO NO.=      2
1   1.3333      0.1667-0.19423E+01-0.51549E+00
2   1.8333      0.1667-0.19423E+01-0.51549E+00
3   1.8333      0.6667-0.19423E+01-0.51549E+00

```

```

.
.
.

```

```

ELEMENTO NO.=      49
1   3.6667      4.8333 0.00000E+00-0.49674E+01
2   3.1667      4.8333 0.00000E+00-0.49674E+01
3   3.1667      4.3333 0.00000E+00-0.49674E+01

```

```

ELEMENTO NO.=      50
1   4.6667      4.8333 0.00000E+00-0.30837E+01
2   4.1667      4.8333 0.00000E+00-0.30837E+01
3   4.1667      4.3333 0.00000E+00-0.30837E+01

```

FLUJOS ALISADAS NODALES

NODO	X-FLUJ.	Y-FLUJ.	Z-FLUJ.
1	-0.71444E+00	-0.71444E+00	
2	-0.15237E+01	-0.59108E+00	
3	-0.28089E+01	-0.46943E+00	
4	-0.43497E+01	-0.31876E+00	
5	-0.62205E+01	-0.13445E+00	
6	-0.69533E+01	0.00000E+00	
7	-0.59108E+00	-0.15237E+01	
8	-0.12145E+01	-0.12145E+01	
9	-0.24529E+01	-0.99529E+00	
10	-0.39104E+01	-0.70666E+00	
11	-0.56904E+01	-0.34046E+00	
12	-0.68189E+01	-0.67225E-01	
13	-0.46943E+00	-0.28089E+01	
14	-0.99529E+00	-0.24529E+01	
15	-0.21020E+01	-0.21020E+01	
16	-0.34165E+01	-0.15432E+01	
17	-0.50792E+01	-0.78964E+00	
18	-0.63309E+01	-0.21035E+00	
19	-0.31876E+00	-0.43497E+01	
20	-0.70666E+00	-0.39104E+01	
21	-0.15432E+01	-0.34165E+01	
22	-0.25745E+01	-0.25745E+01	
23	-0.39762E+01	-0.13619E+01	
24	-0.53518E+01	-0.38439E+00	
25	-0.13445E+00	-0.62205E+01	
26	-0.34046E+00	-0.56904E+01	
27	-0.78964E+00	-0.50792E+01	
28	-0.13619E+01	-0.39762E+01	
29	-0.22055E+01	-0.22055E+01	
30	-0.37116E+01	-0.62790E+00	
31	0.00000E+00	-0.69533E+01	
32	-0.67224E-01	-0.68189E+01	
33	-0.21035E+00	-0.63309E+01	
34	-0.38439E+00	-0.53518E+01	
35	-0.62790E+00	-0.37116E+01	
36	-0.15419E+01	-0.15419E+01	

En la siguiente gráfica se muestra la distribución de temperaturas en el dominio mediante isothermas, así como las distribuciones de los flujos calóricos en X e Y.

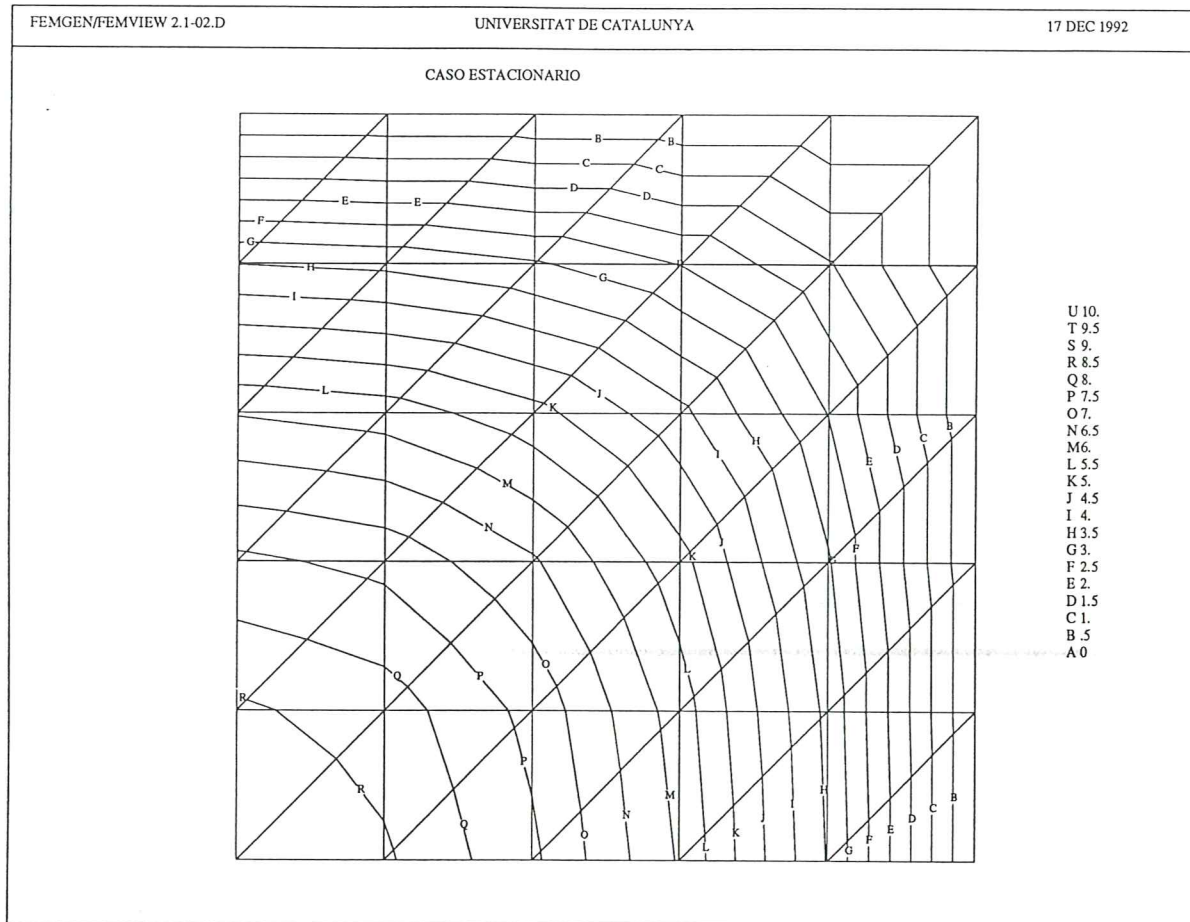


Figura 13 Distribución de Isotermas en caso estacionario.

11.2.2 Caso transitorio $\alpha = 1.0$

La geometría del problema, en este caso es la correspondiente a la figura 12, considerando un valor de para rr. y de 1.0 para a, por lo que se utiliza un esquema de integración temporal fuertemente implícito [21], utilizando 25 iteraciones en el tiempo, partiendo de una temperatura inicial de todos los nodos igual a cero.

De acuerdo con la metodología establecida para la lectura de datos (apéndice I), a las tarjetas del apartado 11.2.1 se adicionan las que se describen en el siguiente párrafo.

11.2.2.1 LISTADO DE DATOS CASO TRANSITORIO $\alpha = 1.0$.(TARJETAS ADICIONALES)

La primer tarjeta corresponde a los datos generales, mientras que la siguiente tarjeta que se modifica, se refiere a las propiedades del material tipo 1, precediendo las correspondientes a la integración temporal.

```

36   50   11    1    3    1    3    2    1
.
.
.
1      2.0e+00          2.0          1.0          2.4          1.0
25      1.0000          1.00
36      0.0000
0      0
      generación interna de calor
.
.
.

```

11.2.2.2 LISTADO DE RESULTADOS CASO TRANSITORIO $\alpha = 1.0$

Dado que los resultados numéricos son muy extensos, solo se muestra los valores de la temperatura de dos nodos, durante toda la evolución del problema; así mismo se gráfica las isotermas sobre el dominio para los pasos de tiempo 1, 9 17 y 25

TEMPERATURAS PASO 1	TEMPERATURAS PASO 2	TEMPERATURAS PASO 3
NODO TEMPERATURA	NODO TEMPERATURA	NODO TEMPERATURA
1 0.219147E+01	1 0.399268E+01	1 0.537256E+01
2 0.215197E+01	2 0.388487E+01	2 0.519438E+01
TEMPERATURAS PASO 4	TEMPERATURAS PASO 5	TEMPERATURAS PASO 6
NODO TEMPERATURA	NODO TEMPERATURA	NODO TEMPERATURA
1 0.639050E+01	1 0.712784E+01	1 0.765746E+01
2 0.615336E+01	2 0.684554E+01	2 0.734191E+01
TEMPERATURAS PASO 7	TEMPERATURAS PASO 8	TEMPERATURAS PASO 9
NODO TEMPERATURA	NODO TEMPERATURA	NODO TEMPERATURA
1 0.803644E+01	1 0.830718E+01	1 0.850044E+01
2 0.769684E+01	2 0.795032E+01	2 0.813123E+01
TEMPERATURAS PASO 10	TEMPERATURAS PASO 11	TEMPERATURAS PASO 12
NODO TEMPERATURA	NODO TEMPERATURA	NODO TEMPERATURA
1 0.863836E+01	1 0.873676E+01	1 0.880697E+01
2 0.826033E+01	2 0.835244E+01	2 0.841815E+01
TEMPERATURAS PASO 13	TEMPERATURAS PASO 14	TEMPERATURAS PASO 15
NODO TEMPERATURA	NODO TEMPERATURA	NODO TEMPERATURA
1 0.885705E+01	1 0.889279E+01	1 0.891829E+01
2 0.846504E+01	2 0.849849E+01	2 0.852235E+01
TEMPERATURAS PASO 16	TEMPERATURAS PASO 17	TEMPERATURAS PASO 18
NODO TEMPERATURA	NODO TEMPERATURA	NODO TEMPERATURA
1 0.893647E+01	1 0.894945E+01	1 0.895871E+01
2 0.853937E+01	2 0.855152E+01	2 0.856018E+01
TEMPERATURAS PASO 19	TEMPERATURAS PASO 20	TEMPERATURAS PASO 21
NODO TEMPERATURA	NODO TEMPERATURA	NODO TEMPERATURA
1 0.896531E+01	1 0.897003E+01	1 0.897339E+01
2 0.856637E+01	2 0.857078E+01	2 0.857392E+01
TEMPERATURAS PASO 22	TEMPERATURAS PASO 23	TEMPERATURAS PASO 24

NODO	TEMPERATURA	NODO	TEMPERATURA	NODO	TEMPERATURA
1	0.897579E+01	1	0.897750E+01	1	0.897872E+01
2	0.857617E+01	2	0.857777E+01	2	0.857891E+01

TEMPERATURAS PASO 25

NODO	TEMPERATURA
1	0.897959E+01
2	0.857973E+01

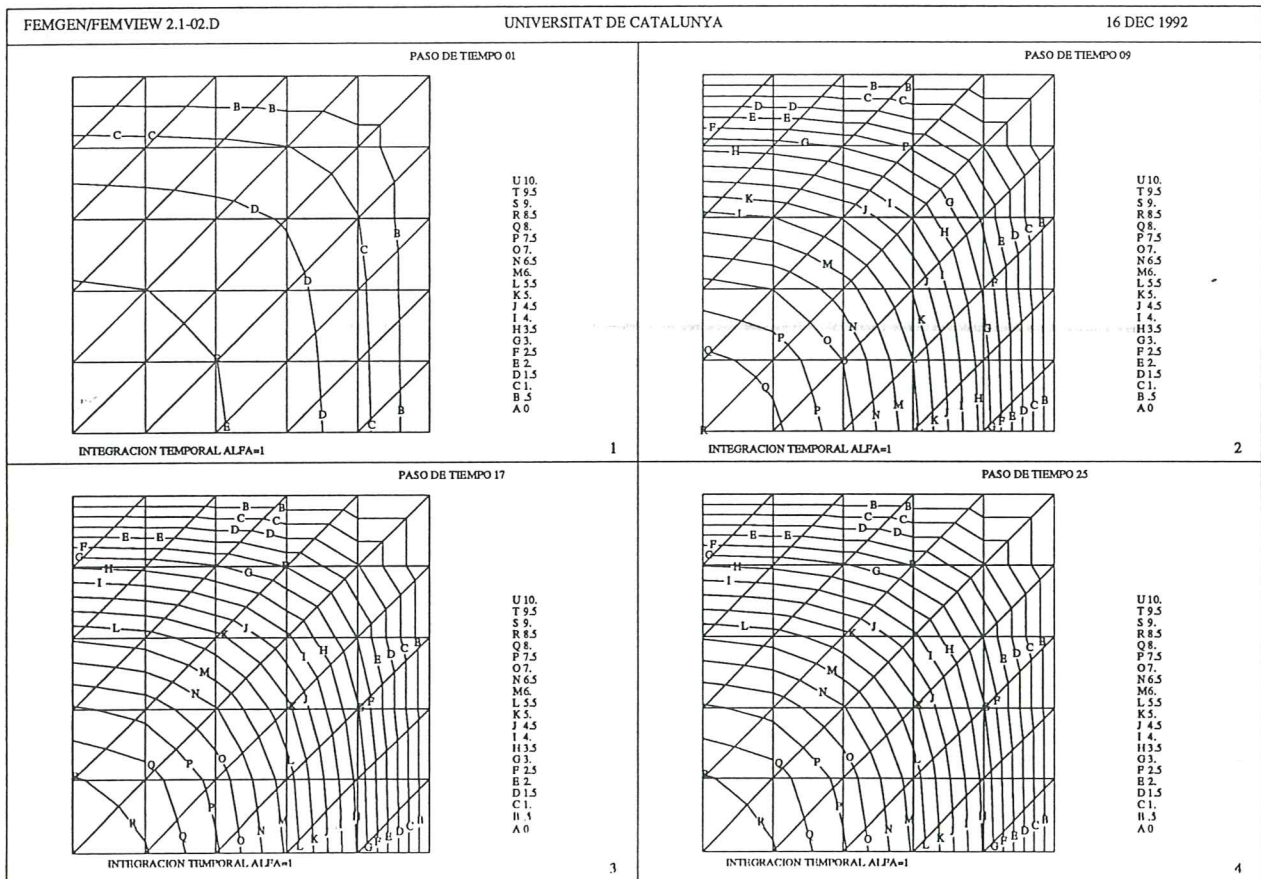


Figura 14 Distribución de isotermas

11.2.3 Caso transitorio $\alpha = 2/3$

La geometría del problema corresponde íntegramente a los datos del apartado 11.2.2, con excepción del valor de a que, usando la integración de Galerkin corresponde a $2/3$.

11.2.3.1 LISTADO DE DATOS CASO TRANSITORIO $\alpha = 2/3$ (TARJETAS ADICIONALES)

La tarjeta que se modifica respecto al problema anterior corresponde a la referente a la integración temporal.

25 1.0000 0.66666

11.2.3.2 LISTADO DE RESULTADOS CASO TRANSITORIO $\alpha = 2/3$

Dado que los resultados numéricos son muy extensos, solo se muestra los valores de la temperatura de dos nodos, durante toda la evolución del problema; así mismo se gráfica las isotermas sobre el dominio para los pasos de tiempo 1, 9 17 y 25

TEMPERATURAS PASO 1	TEMPERATURAS PASO 2	TEMPERATURAS PASO 3
NODO TEMPERATURA	NODO TEMPERATURA	NODO TEMPERATURA
1 0.231373E+01	1 0.430091E+01	1 0.577800E+01
2 0.228730E+01	2 0.419724E+01	2 0.558338E+01
TEMPERATURAS PASO 4	TEMPERATURAS PASO 5	TEMPERATURAS PASO 6
NODO TEMPERATURA	NODO TEMPERATURA	NODO TEMPERATURA
1 0.679258E+01	1 0.748559E+01	1 0.795982E+01
2 0.653223E+01	2 0.718147E+01	2 0.762518E+01
TEMPERATURAS PASO 7	TEMPERATURAS PASO 8	TEMPERATURAS PASO 9
NODO TEMPERATURA	NODO TEMPERATURA	NODO TEMPERATURA
1 0.828354E+01	1 0.850478E+01	1 0.865590E+01
2 0.792823E+01	2 0.813530E+01	2 0.827675E+01
TEMPERATURAS PASO 10	TEMPERATURAS PASO 11	TEMPERATURAS PASO 12
NODO TEMPERATURA	NODO TEMPERATURA	NODO TEMPERATURA
1 0.875915E+01	1 0.882968E+01	1 0.887786E+01
2 0.837339E+01	2 0.843941E+01	2 0.848451E+01
TEMPERATURAS PASO 13	TEMPERATURAS PASO 14	TEMPERATURAS PASO 15
NODO TEMPERATURA	NODO TEMPERATURA	NODO TEMPERATURA
1 0.891078E+01	1 0.893327E+01	1 0.894863E+01
2 0.851532E+01	2 0.853637E+01	2 0.855075E+01
TEMPERATURAS PASO 16	TEMPERATURAS PASO 17	TEMPERATURAS PASO 18
NODO TEMPERATURA	NODO TEMPERATURA	NODO TEMPERATURA
1 0.895913E+01	1 0.896630E+01	1 0.897120E+01
2 0.856058E+01	2 0.856729E+01	2 0.857187E+01
TEMPERATURAS PASO 19	TEMPERATURAS PASO 20	TEMPERATURAS PASO 21
NODO TEMPERATURA	NODO TEMPERATURA	NODO TEMPERATURA
1 0.897454E+01	1 0.897683E+01	1 0.897839E+01
2 0.857501E+01	2 0.857714E+01	2 0.857861E+01
TEMPERATURAS PASO 22	TEMPERATURAS PASO 23	TEMPERATURAS PASO 24
NODO TEMPERATURA	NODO TEMPERATURA	NODO TEMPERATURA
1 0.897946E+01	1 0.898019E+01	1 0.898069E+01
2 0.857961E+01	2 0.858029E+01	2 0.858075E+01
TEMPERATURAS PASO 25		
NODO TEMPERATURA		
1 0.898103E+01		
2 0.858107E+01		

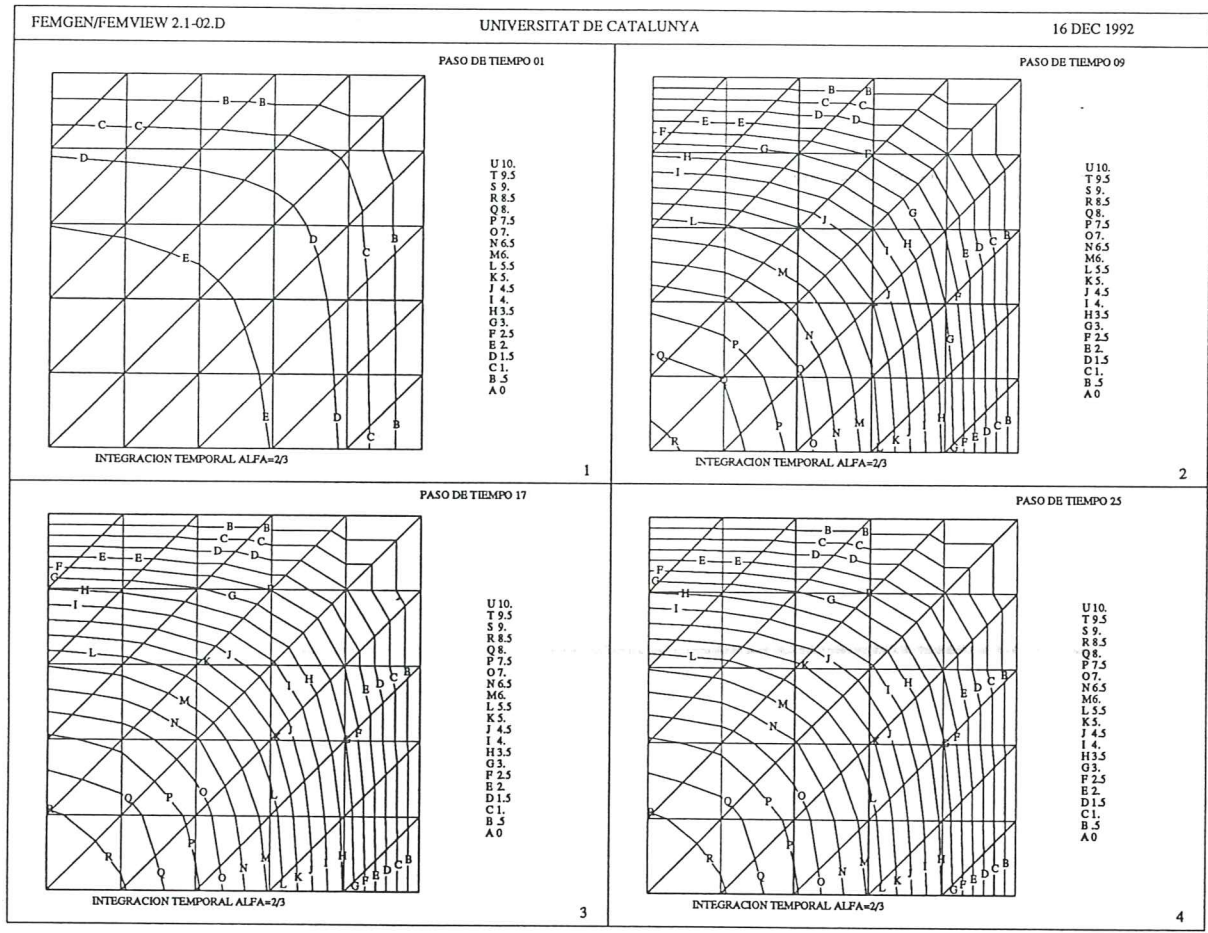


Figura 15 Distribución de isotermas

Con objeto de comparar ambos esquemas de integración, en la figura 16 se presenta la convergencia al valor estacionario de la temperatura del nodo 1

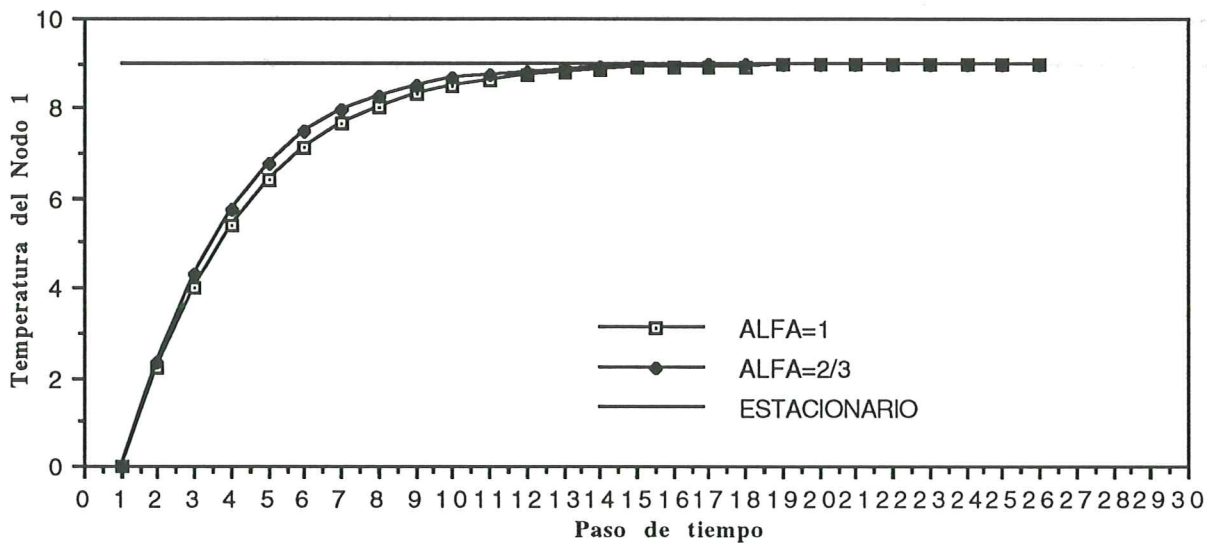


Figura 16 Convergencia de temperatura con distintos esquemas de integración

REFERENCIAS

1. Mackerle, J. y Fredriksson B. *Handbook of Finite Element Software*, Chartwell-Bratt Ltd., Bromley, U.K., 1990.
2. Hinton, E. y Owen, D.R.J., *Finite element programming*, Academic Press, 1979.
3. Pres, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., *Numerical Recipes. The art of Scientific Computing*, Cambridge Univ. Press, 1986.
4. Oñate, E. "Cálculo de estructuras por el método de los elementos finitos", CIMNE, Barcelona, 1992.
5. FEMGEN/FEMVIEW, The Finite Element Pre and Post-processor, User Manual, Femview Limited, Leicester, U.K., 1991.
6. I-DEAS, General capabilities, P-0100, Structural Dynamics Research Corporation, Milford, Ohio, USA, 1991.
7. PATRAN Plus User Manual, PDA Engineering, PATRAN Division, Costa Mesa, California, USA, 1991.
8. Bugada, G., "FLAVIA. Programa para visualización gráfica en 2 y 3 dimensiones", Publicación CIMNE, Barcelona, 1992.
9. Fong, H. M., "Interactive graphics and commercial finite element code, Mechanical" Engng., 106(6), pp. 18-27, 1989.
10. Bonet, J. y Peraire, J., "An alternate digital tree algorithm for geometric searching and intersection problems", Int. J. Num. Meth. Engng. Vol. 31, pp 1-17, 1991.
11. Hinton, E. y Owen, D.R.J., *Introduction to finite element computations*, Pineridge Press, 1980.
12. Nguye, V. P., "Automatic mesh generation with tetraedon elements", Int. J. Num. Meth. Engng. Vol. 18, pp 273-289, 1982.
13. Peiró, J. *A finite element procedure for the solution of the Euler equations on unstructured meshes*, Ph.D. Thesis, Civil Eng. Dpt., Univ. College of Swansea, U.K., 1989.
14. Pissanetzky, S., "Kubik: An automatic three dimensional mesh generator", Int. J. Num. Meth. Engng. Vol. 17, pp 255-269, 1987.
15. Zienkiewickz, O.C., *El Método de los Elementos Finitos*, 3ra. Edición, Ed. Reverté, Barcelona, 1979.
16. Zienkiewickz, O.C., *The Finite Element Method*, 4ra. Edición, Ed. Mc. Graw Hill, Vol. I, 1989, Vol II, 1991.
17. Ralston, A., *Introducción al análisis numérico*, Limusa-Wiley, 1970.
18. Irons, B.M. y Ahmad, S., *Techniques of finite elements*, Ellis Harwood, Chichester, 1980.
19. Timoshenko, S.P. y Goodier, J.N., *Teoría de la elasticidad*, Edic. Urmo, 1968.
20. Timoshenko, S.P. y Woinowsky-Krieger S., *Teoría de Placas y Láminas*, Edic. Urmo, Bilbao, 1990.
21. Marshall, Guillermo, *Solución numérica de ecuaciones diferenciales*, Tomo I, d. Reverté Argentina S.A. Argentina 1985.

APENDICE I

INSTRUCCIONES PARA ENTRADA DE DATOS Y LISTADO DE VARIABLES DEL PROGRAMA CALTEP

I.I INSTRUCCIONES PARA ENTRADA DE DATOS

Las instrucciones se han agrupado en "tarjetas" formateadas. Cada "grupo de tarjetas" incluye un conjunto de datos similares; ejemplo: coordenadas nodales, cargas puntuales, etc.

GRUPO DE TARJETAS 1. NUMERO DE PROBLEMAS Y TITULO DE CADA PROBLEMA.

TARJETA 1.1	I5	NPROB	Número de problemas a analizar.
TARJETA 1.2	A80	TITULO	Título de cada problema.

NOTA: Tantas tarjetas 1.2 como número de problemas a analizar.

GRUPO DE TARJETAS 2. PARAMETROS DE CONTROL.

TARJETA 2.1	9I5	NPNOD	Número total de nodos.
		NELEM	Número de elementos.
		NPRES	Número de nodos con temp. prescritas.
		NTRAN	Indicador de tipo de problema.
			=0 Caso estacionario
			=1 Caso transitorio, si $\alpha=0$, no diagonaliza.
			=2 Caso transitorio, si $\alpha=0$, diagonalización por suma de filas
			=3 Caso transitorio, si $\alpha=0$, diagonalización conservación de masa.
		NNODE	Número de nodos por elemento.
		NMATS	Número de materiales.
		NGAUS	Número de puntos de Gauss.
		NDIME	Número de dimensiones.
		IWRIT	Indicador para escritura de datos.
			=1 Escribe fichero de resultados.

GRUPO DE TARJETAS 3. CONECTIVIDADES NODALES DE CADA ELEMENTO

TARJETA 3.1	15I5	NUMEL	Número de elemento.
		MATNU(NUMEL)	Tipo de material en elemento.
		LNODS(NUMEL,INODE)	Conectividades (bucle sobre NNODE)
TARJETA 3.2	15I5		Continúan conectividades en elem. 3D.

NOTAS:

Tantas tarjetas 3.1 como número de elementos (NELEM)
La tarjeta 3.2 se utilizar solamente para NNODE=20

GRUPO DE TARJETAS 4. COORDENADAS NODALES

TARJETA 4.1	I5,3F10.5		
	IPNOD		Número de nodo.
	COORD(IPNOD,1)		Coordenada X del nodo.
	COORD(IPNOD,2)		Coordenada Y del nodo (en 2D)
	COORD(IPNOD,3)		Coordenada Z del nodo (en 3D)

NOTA:

Tantas tarjetas 4.1 como número de nodos (NPNO)

GRUPO DE TARJETAS 5. MOVIMIENTOS PRESCRITOS

TARJETA 5.1	1X,I4,3X,<NGDLN>I1,<NGDLN>F10.5		
	NODPR(IPRES)		Nodo.
	INPRE(IPRES,NGLN)		Código de temperatura prescrito
		=0	Temperatura libre.
		=1	Temperatura prescrita.
	PRESC(IPRES,NGDLN)		Valor de la Temperatura prescrita.

NOTA:

Tantas tarjetas 5.1 como número de nodos prescritos (NPRES)

GRUPO DE TARJETAS 6. PROPIEDADES DE LOS MATERIALES

TARJETA 6.1	1X,I4,5E15.5		
	NUMAT		Número de material.
	PROPS(NUMAT,1..NDIME)		Conductividad Térmica en X, Y Z.
	PROPS(NUMAT,NDIME+1)		Alpha.
	PROPS(NUMAT,NDIME+2)		roR.
	PROPS(NUMAT,NDIME+3)		roC.

NOTA:

Tantas tarjetas 6.1 como número de propiedades (NDIME+2) si NTRAN = 0 ó (NDIME+3) si NTRAN > 0

GRUPO DE TARJETAS 7. INTEGRACION EN EL TIEMPO

TARJETA 7.1	I5,2E15.5		
	NTIME		Número de iteraciones en el tiempo.
	DTIME		Incremento de tiempo en cada iteración.
	ALFAT		Esquema de Intergación temporal.
TARJETA 7.2	I5,2E15.5		
	NNCIT		Número de nodos con condición de temperatura inicial distinta de cero.
	CITIM		Temperatura inicial generalizada.
TARJETA 7.3	I5,2E15.5		
	IPNOD		Nodo con condición de temperatura inicial distinta de cero.
	XTIME(IPNOD)		Temperatura inicial del nodo (bucle sobre NNCIT).

NOTAS:

Solo sera necesario introducir el grupo 7 cuando NTRAN>0

En caso de tener una temperatura inicial distinta de cero igual en todos los nodos solo basta introducir NNCIT=NPNO y dar el valor en CITIM, si este no es el caso CITIM carece de significado.

El grupo de tarjetas 7.3 habrá tantas como NNCIT; si NNCIT=NPNO no serán necesarias.

GRUPO DE TARJETAS 8. CONDICIONES DE FRONTERA DE FLUJO POR RADICACION / CONVECCION

TARJETA 8.1	1X,2I4 NFRON NNFRO	Número de elementos de frontera Número de nodos en cada lado.
TARJETA 8.2	1X,<1+NNFRO>I4 CONVE(IFRON,1) CONVE(IFRON,1+NNFRO)	Número de elemento de frontera. Número de nodos de frontera (numeración global)

NOTA:

Solo sera necesario introducir el grupo 8.2 cuando NFRON <> 0

GRUPO DE TARJETAS 9. PARAMETROS DE LAS CONDICIONES DE FRONTERA

TARJETA 9.1	A20 TITULO	Titulo del estado de carga.
TARJETA 9.2	4I5 IPUNT IPESO IDIST	Indicador de flujo puntual. =0 No existe el caso. =1 Existe el caso. Indicador de generación interna de calor. =0 No existe el caso. =1 Existe el caso. Indicador de flujo y temperatura externa aplicada por lado. =0 No existe el caso. =1 Existe el caso.

GRUPO DE TARJETAS 10. FLUJOS PUNTUALES

TARJETA 10.1	I5,F10.3 LODPT PNODT(1)	Nodo cargado. Flujo puntual nodal.
---------------------	-------------------------------	---------------------------------------

NOTAS:

Se repite la tarjeta 9.1 para cada nodo cargado. La última tarjeta 9.1 debe corresponder al nodo de mayor numeración, con independencia de que esté o no cargado (en este último caso PNODT=0.0).
Si IPUNT=0 se omite este grupo de tarjetas.

GRUPO DE TARJETAS 11. FLUJO Y TEMPERATURA EXTERNA ACTUANTES SOBRE UN LADO DEL ELEMENTO.

TARJETA 11.1	2I5 NEDGE NODEG	Número de lados expuestos. Número de nodos por lado expuesto.
TARJETA 11.2	I5,<NODEG>I5 NEASS NOPRS(1NODEG)	Número del elemento cargado. Nodo cargado en numeración global.
TARJETA 11.3	2F10.3 FLUJO TEMPE	Flujo actuante sobre el lado. Temperatura externa sobre el lado.

NOTAS:

El flujo normal es positivo si va dirigido hacia el interior del elemento.
Si IDIST=0 se omite el grupo de tarjetas 11.
Tantas tarjetas 11.3 como nodos por lado y tantos grupos de tarjetas 11.2 y 11.3 como número de lados

I.II LISTADO DE VARIABLES DEL PROGRAMA CALTEP.

VARIABLES ESCALARES

ALFAT	Valor α del esquema de integración.
ALFAU	$1-\alpha$.
AUX3	Variable auxiliar de direccionamiento de ficheros.
B	Variable auxiliar operacional.
CERO	Variable Dummy (siempre igual a 0).
CITIM	Temperatura inicial generalizada.
DAREA	Diferencial de área del elemento.
DJACB	Determinante del Jacobiano.
DLONG	Diferencial de longitud.
DVOLU	Diferencial de volumen.
EGASP	Variable Dummy sobre EGISP (siempre igual a cero).
EGISP	Coordenada ζ del punto de Gauss.
ERROR	Variable de condición de error.
ETASP	Coordenada ξ del punto de Gauss.
EXISP	Coordenada η del punto de Gauss.
FACTR	Factor de multiplicación de la ecuación a reducir.
FLUJO	Flujo actuante sobre el lado.
ICOLS	Contador sobre el número de columnas de la matriz global.
IDIME	Contador sobre el número de dimensiones.
IDIST	Indicador de flujo y temperatura externa.
IELEM	Contador sobre el número de elementos.
IEON1	Contador sobre el número de ecuaciones.
IEONS	Contador sobre el número de ecuaciones.
IEVAB	Contador sobre el número de variables nodales.
IFRON	Contador sobre nodos de frontera NFRON .
IGASH	Contador sobre cuadraturas no triangulares de Gauss.
IGAUS	Contador sobre los puntos de Gauss.
IGDLN	Contador sobre los grados de libertad nodales.
IMATS	Contador sobre número de materiales.
INCIT	Contador sobre NNCIT .
INFRO	Contador sobre los nodos de frontera.
INODE	Contador sobre los nodos del elemento.
INPUT	Nombre del fichero de datos.
IODEG	Contador sobre los nodos por lado expuesto.
IPESO	indicador de generación interna de calor.
IPNOD	Contador sobre los nodos de la malla.
IPRES	Contador sobre los nodos prescritos.
IPROB	Contador sobre el Número de problemas.
IPROP	Contador sobre las propiedades de los materiales.
IPUNT	Indicador de temperatura puntual.
IROWS	Contador sobre el número de filas de la matriz global.
ITENS	Contador Dummy sobre NTENS .
ITIME	Contador sobre las iteraciones en tiempo.
ITOTV	Contador sobre el número de temperaturas nodales.
IVARI	Contador sobre el vector de temperaturas nodales.
IWRIT	Indicador para escritura de datos.
JDIME	Contador sobre el número de dimensiones.
JELEM	Contador sobre el número de elementos.

JEVAB	Contador sobre el número de variables nodales.
JGASH	Contador sobre cuadraturas no triangulares de Gauss.
JGAUS	Contador sobre los puntos de Gauss.
JGDLN	Contador sobre los grados de libertad nodales.
JNFRO	Contador sobre los nodos de frontera.
JNODE	Contador sobre los nodos del elemento.
JPNOD	Contador sobre los nodos de la malla.
JTENS	Contador Dummy sobre NTENS .
JTOTV	Contador sobre el número de temperaturas nodales.
KCONT	Contador sobre grados de libertad.
KGAUS	Contador sobre los puntos de Gauss.
KNODE	Contador sobre los nodos del elemento.
KPGAU	Contador sobre puntos de integración numérica.
LNODE	Variable auxiliar operacional.
LODPT	Nodo con temperatura puntual.
LPROP	Tipo de material para cada elemento.
NBAC1	Contador decreciente del número de ecuaciones.
NBACK	Contador decreciente del número de ecuaciones.
NCOLE	Columna de la matriz de rigidez elemental del nodo JNODE .
NCOLS	Columna de la matriz de rigidez global del nodo JNODE .
NCONT	Apuntador sobre los grados de libertad.
NDIME	Número de dimensiones en el problema.
NEASS	Número del elemento cargado.
NEDGE	Número de lados expuestos.
NELEM	Número de elementos total.
NEONS	Número de ecuaciones a resolver.
NEVAB	Número de variables por elemento.
NFRON	Número de fronteras de convección/radiación.
NGAUS	Número de puntos de integración de Gauss.
NGDLN	Grados de libertad nodales (siempre igual a 1).
NGISH	Número de grados de libertad prescritos.
NGUSH	Contador sobre los grados de libertad prescritos.
NLOCA	Localizador del nodo con temperatura puntual.
NMATS	Número de materiales distintos en la geometría.
NNCIT	Número de nodos con condición de temp. inicial distinta de cero.
NNFRO	Número de nodos de frontera convección/radiación.
NNOD1	Variable auxiliar para la lectura de la tarjeta 3.2.
NNOD2	Variable auxiliar para la lectura de la tarjeta 3.2.
NNODE	Número de nodos por elemento.
NODEG	Número de nodos por lado expuesto.
NODEI	INODE en numeración global.
NODEJ	JNODE en numeración global.
NPNOD	Número de nodos en toda la malla.
NPOSN	Localizador de nodo donde aplicar la temperatura puntual.
NPRES	Número de prescripciones nodales.
NPROB	Número de problemas a resolver.
NPROP	Número de propiedades por cada material.
NROWE	Fila de la matriz de rigidez elemental del nodo INODE .
NROWS	Fila de la matriz de rigidez global del nodo INODE .
NSTR1	Numero de flujos finales.
NTENS	Variable Dummy (siempre igual a NDIME).
NTIME	Número de iteraciones sobre el tiempo.
NTIPO	Variable Dummy (siempre igual a 1).
NTOTV	Número total de temperaturas nodales.
NTRAN	Indicador de problema transitorio o estacionario.

NUMAT	Número de materiales.
NUMEL	Número de elementos.
OUTPUT	Nombre del Fichero de resultados.
PATOP	Variable auxiliar operacional.
PESPP	Fuente de calor interna por densidad.
PIVOT	Elemento pivote de la matriz de rigidez, en la reducción.
Q	Coordenada ζ del punto de Guass.
Q2	Valor de $2*\zeta$.
QQ	Valor $\zeta*\zeta$.
RESID	Reacción de la ecuación a reducir.
ROCEE	Densidad por calor por densidad entre incremento de tiempo.
S	Coordenada ξ del punto de Guass.
S 1	Valor $\xi+1$.
S 2	Valor $2*\xi$.
S 9	Valor $\xi-1$.
SS	Valor $\xi*\xi$.
SST	Valor $\xi*\xi*\eta$.
ST	Valor $\xi*\eta$.
ST2	Valor $2*\xi*\eta$.
STT	Valor $\xi*\eta*\eta$.
T	Coordenada η del punto de Guass.
T1	Valor $\eta+1$.
T2	Valor $2*\eta$.
T9	Valor $\eta-1$.
TEMPE	Temperatura externa sobre el lado.
TITULO	Título del problema y/o estado de cargas.
TT	Valor $\eta*\eta$.
UNOVA	Variable Dummy (siempre igual a 1).

VARIABLES VECTORIALES

ASLOD(200)	Temperaturas nodales equivalentes globales.
BMATZ (3,81)	Matriz B
BSLOD(200)	Temperaturas nodales equivalentes globales.
CARGA (200,20)	Matriz para almacenar temperaturas nodales
CONVE (200,27)	Elementos y nodos de frontera por radiación convección.
COORD (200,3)	Coordenadas nodales
COREL (3,27)	Coordenadas nodales locales
CORPG (3,27)	Coordenadas en los puntos de gauss.
D(2)	Diferencial de x e y.
DBMAT (3,81)	Matriz del producto DB.
DCART (3,27)	Derivadas cartesianas de las funciones de forma.
DERIV(3,20)	Derivadas naturales de las funciones de forma.
DMATZ (3,3)	Matriz constitutiva D.
FFORM(20)	Funciones de forma.
FTFOR(20,20)	Matriz de rigidez por aportación de condiciones de frontera.
IFFIX(200,3)	Indicador de temperaturas prescritas globales.
INPRE (200,3)	Indicador de temperaturas prescritas elementales.
JFFIX(200)	Indicador de temperaturas prescritas globales.
LNODS (200,27)	Conectividades nodales.
LOCAL(8)	Numeración local de nodos de frontera.
MATNU (200)	Tipo de material del elemento.

NODPR (200)	Numero de nodos prescritos.
PESPG(14)	Pesos de los puntos de integración en cuadriláteros.
PNODT(3)	Conectividades para flujos repartidos.
POSGT(14)	Coordenadas de los pts. de integración de triángulos.
POSPG(14)	Coordenadas de pts. de integración en cuadriláteros.
PRESC (200,3)	Valores de las temperaturas prescritas.
PRESS(20,2)	Valor del flujo y temp. externa por lado.
PROPS (10,200)	Propiedades del material.
PXCOM(20)	Temperatura nodal equivalente por elemento.
RIGID(20,20)	Matriz de rigidez elemental.
TENSG(3)	Flujos en cada punto de integración.
TENSZ (3,81,27)	Flujos elementales en cada punto de integración.
XFORM(20)	Vector de funciones de forma de lado con convecc.Rad.
XJACI(3,3)	Matriz Jacobiana inversa.
XJACM(3,3)	Matriz Jacobiana.
XTIME (200)	Temperatura en el tiempo n-1.
ZJACC(3,3)	Matriz Jacobiana.
ZMASA(20,20)	Matriz de masa elemental.

APENDICE II

LISTADO DEL PROGRAMA CALTEP

```

1.      PROGRAM CALTEP
2.      C*****
3.      C
4.      C*** SUBROUTINA PRINCIPAL
5.      C
6.      C*****
7.      C*****COMMON STATEMENTS
8.          IMPLICIT NONE
9.          INCLUDE 'dtsgral.f'
10.         INCLUDE 'data.f'
11.         INCLUDE 'gaussdat.f'
12.         INCLUDE 'calculo.f'
13.         INCLUDE 'solu.f'
14.         INCLUDE 'solucuas.f'
15.         INTEGER*2 MPROB, IPROB, IVARI, ITIME
16.         CHARACTER*80 TITULO
17.         CHARACTER*10 INPUT, OUTPUT
18.         CHARACTER*13 AUX3
19.         DATA AUX3 /'/aux3/zarate/'/
20.      C*****COMMON STATEMENTS
21.         OPEN (2, FILE='COMAN.DAT', FORM='FORMATTED')
22.         READ(2, 800) INPUT
23.         READ(2, 800) OUTPUT
24.         CLOSE(2, STATUS='KEEP')
25.         800 FORMAT(A10)
26.         OPEN (5, FILE=INPUT, FORM='FORMATTED')
27.         OPEN (6, FILE=OUTPUT, FORM='FORMATTED')
28.         OPEN (1, FILE=AUX3// 'TEMP1.TMP', FORM='UNFORMATTED')
29.         OPEN (3, FILE=AUX3// 'TEMP3.TMP', FORM='UNFORMATTED')
30.         OPEN (4, FILE=AUX3// 'TEMP4.TMP', FORM='UNFORMATTED')
31.         OPEN (8, FILE=AUX3// 'TEMP8.TMP', FORM='UNFORMATTED')
32.         OPEN (10, FILE=AUX3// 'TEMP10.TMP', FORM='UNFORMATTED')
33.         OPEN (29, FILE=AUX3// 'CALSEF.POS', FORM='UNFORMATTED')
34.      C
35.      C*** LEE NUMERO DE PROBLEMAS A ANALIZAR
36.      C
37.         READ(5, 900) NPROB
38.         900 FORMAT(I5)
39.         WRITE(6, 905) NPROB
40.         905 FORMAT(//10X, 'NUMERO DE PROBLEMAS= ', I5)
41.      C
42.      C*** BUCLE SOBRE NUMERO DE PROBLEMAS
43.      C
44.         MPROB=INT2(NPROB)
45.         ITIME=0
46.         WRITE(29) MPROB
47.         DO IPROB=1, NPROB
48.             REWIND 1
49.             REWIND 3
50.             REWIND 4
51.             WRITE(6, 910) IPROB
52.             910 FORMAT(///// , 6X, 'PROBLEMA NO. ', I3, ////)
53.             READ(5, 915) TITULO
54.             915 FORMAT(A80)

```

```

55.          WRITE(6,920) TITULO
56.      920  FORMAT(A80,////)
57.          CALL DATOS
58.          CALL GAUSSQ(NDIME,NNODE,NGAUS)
59.          CALL RIGIMAT
60.          CALL FUERZAS
61.          CALL ENSAMBLA
62.          IF (NTRAN.EQ.0) THEN
63.              CALL REDUCE
64.              CALL SUSTITUIR
65.              CALL TENSIONES
66.              CALL SUAV1
67.              CALL FEMV(ITIME,TITULO)
68.          ELSE
69.              IF (ALFAT.EQ.0) CALL DIAGONAL
70.              DO ITIME=1,NTIME
71.      C
72.      C***      APLICA LAS CONDICIONES DE TIEMPO T-1
73.      C
74.              CALL FUERZAS_T0
75.      C
76.      C***      RESUELVE POR ELIMINACION GAUSSIANA
77.      C
78.              IF (ITIME.EQ.1) THEN
79.                  CALL REDUCE
80.              ELSE
81.                  CALL REDUCE1
82.              ENDIF
83.              CALL SUSTITUIR
84.      C
85.      C***      CALCULA LAS TENSIONES EN LOS ELEMENTOS
86.      C
87.              CALL TENSIONES
88.      C
89.      C***      SUAVIZADO DE TENSIONES
90.      C
91.              CALL SUAV1
92.      C
93.      C***      EL TIEMPO T SE CONVIERTE EN T-1
94.      C
95.              DO IVARI=1,NGDLN*NPNOD
96.                  XTIME(IVARI)=DESPL(IVARI)
97.              ENDDO
98.          ENDDO
99.          ENDIF
100.      ENDDO
101.      CLOSE(1,STATUS='DELETE')
102.      CLOSE(3,STATUS='DELETE')
103.      CLOSE(4,STATUS='DELETE')
104.      CLOSE(8,STATUS='DELETE')
105.      CLOSE(10,STATUS='DELETE')
106.      CLOSE(29,STATUS='KEEP')
107.      CLOSE(5,STATUS='KEEP')
108.      CLOSE(6,STATUS='KEEP')
109.      STOP
110.      END

1.          SUBROUTINE BMAT(IELEM,KPGAU)
2.      C*****
3.      C
4.      C***      MATRIZ DE DEFORMACION B
5.      C

```



```

6.      C*****
7.      IMPLICIT NONE
8.      INTEGER*2      IELEM,KPGAU
9.      INTEGER*2      KCONT, ITENS, IEVAB, INODE, IDIME
10.     INCLUDE 'dtsgral.f'
11.     INCLUDE 'calcul0.f'
12.     C*****
13.     KCONT=0
14.     DO ITENS=1,NTENS
15.       DO IEVAB=1,NEVAB
16.         BMATZ (ITENS, IEVAB)=0.0
17.       ENDDO
18.     ENDDO
19.     IF (NTIPO.EQ.1) THEN
20.       DO INODE=1,NNODE
21.     C
22.     C***      TRANSMISION DE CALOR
23.     C
24.       DO IDIME=1,NDIME
25.         BMATZ (IDIME, INODE)=DCART (IDIME, INODE)
26.       ENDDO
27.     ENDDO
28.     ELSE
29.       WRITE(6,900)
30.     900   FORMAT(5X, 'ERROR EN SELECCION DE TIPOLOGIA DEL PROBLEMA')
31.       STOP
32.     ENDIF
33.     RETURN
34.     END

```

```

1.      SUBROUTINE CONVECC (IELEM, FTFOR, LPROP)
2.     C*****
3.     C
4.     C***      CALCULA MATRIZ DE RIGIDEZ DE CADA ELEMENTO CUANDO HAY
5.     C***      CONVECCION Y RADIACION
6.     C
7.     C*****
8.     IMPLICIT NONE
9.     INCLUDE 'dtsgral.f'
10.    INCLUDE 'data.f'
11.    INCLUDE 'gaussdat.f'
12.    INCLUDE 'calcul0.f'
13.    INTEGER*2      IELEM, IGAUS, JNFRO, IFRON, IEVAB, JEVAB,
14.    .              INFRO, LNODE, IDIME, LPROP, LOCAL, INODE,
15.    .              UNOVA, KPGAU, JGAUS, JDIME
16.    REAL*4        XLONG, XFORM, ALPHA, EXISP, DLONG, FTFOR,
17.    .              D, DAREA, ZJACC, ETASP
18.    DIMENSION     XFORM(20), FTFOR(60,60), LOCAL(8), D(2),
19.    .              ZJACC(3,3)
20.    C*****
21.    C
22.    C*** CALCULO DE LA APORTACION DE RIGIDEZ POR CONVECCION/RADIACION
23.    C
24.    JNFRO=0
25.    DO IFRON=1,NFRON
26.      IF (CONVE(IFRON,1).EQ.IELEM) JNFRO=IFRON
27.    ENDDO
28.    DO IEVAB=1,NEVAB
29.      XFORM(IEVAB)=0.0
30.      DO JEVAB=1,NEVAB
31.        FTFOR(IEVAB, JEVAB)=0.0
32.      ENDDO

```

```

33.      ENDDO
34.      IF (JNFRO.NE.0) THEN
35.          DO INFRO=1,NNFRO
36.              LNODE=CONVE(JNFRO,INFRO+1)
37.              DO IDIME=1,NDIME
38.                  COREL(IDIME,INFRO)=COORD(LNODE,IDIME)
39.              ENDDO
40.          ENDDO
41.      C
42.      C*** CAMBIA DE NUMERACION GLOBAL A LOCAL
43.      C
44.          DO INFRO=1,NNFRO
45.              DO INODE=1,NNODE
46.                  IF (CONVE(JNFRO,INFRO+1).EQ.
47.                      LNODS(IELEM,INODE)) LOCAL(INFRO)=INODE
48.              ENDDO
49.          ENDDO
50.          ALPHA=PROPS(LPROP,NDIME+1)
51.          IF (NDIME.EQ.1) THEN
52.      C
53.      C*** BUCLE DE INTEGRACION NUMERICA
54.      C
55.          KPGAU=0
56.          DO IGAUS=1,NGAUS
57.              EXISP=POSPG(IGAUS)
58.              CALL FFORMA(EXISP,0,0,NDIME,NNODE,FFORM,DERIV)
59.              CALL JACOBM(IELEM,DLONG,KPGAU)
60.              DLONG=DLONG*PESPG(IGAUS)
61.              DO IEVAB=1,NEVAB
62.                  DO JEVAB=IEVAB,NEVAB
63.                      FTFOR(IEVAB,JEVAB)=FTFOR(IEVAB,JEVAB)+
64.                          FFORM(IEVAB)*FFORM(JEVAB)*DLONG*ALPHA
65.                  ENDDO
66.              ENDDO
67.          ENDDO
68.          ENDIF
69.          IF (NDIME.EQ.2) THEN
70.              UNOVA=1
71.      C
72.      C*** BUCLE DE INTEGRACION NUMERICA
73.      C
74.          DO IGAUS=1,NGAUS
75.              EXISP=POSPG(IGAUS)
76.      C
77.      C*** CALCULA FUNCIONES DE FORMA EN PUNTOS DE INTEGRACION
78.      C
79.          CALL FFORMA(EXISP,0,0,UNOVA,NNFRO,FFORM,DERIV)
80.      C
81.      C*** CALCULA EL JACOBIANO
82.      C
83.          D(1)=0.0
84.          D(2)=0.0
85.          DO INFRO=1,NNFRO
86.      C*** ARMA VECTOR DE LAS FUNCIONES DE FORMA CON CEROS EN LOS NODOS
87.      C*** NO INCLUIDOS EN EL LADO
88.              XFORM(LOCAL(INFRO))=FFORM(INFRO)
89.              DO IDIME=1,NDIME
90.                  D(IDIME)=D(IDIME)+
91.                      DERIV(1,INFRO)*COREL(IDIME,LOCAL(INFRO))
92.              ENDDO
93.          ENDDO
94.          DLONG=SQRT(1+D(2)*D(2)/D(1)/D(1))*D(1)*PESPG(IGAUS)*ALPHA
95.      C

```

```

96.  C*** ARMA LA MATRIZ DE APORTACION
97.  C
98.          DO IEVAB=1,NEVAB
99.          DO JEVAB=IEVAB,NEVAB
100.             FTFOR (IEVAB,JEVAB) =FTFOR (IEVAB,JEVAB) +
101.             .             XFORM (IEVAB) *XFORM (JEVAB) *DLONG
102.             ENDDO
103.          ENDDO
104.          ENDDO
105.          ENDIF
106.          IF (NDIME.EQ.3) THEN
107.             UNOVA=2
108.             KPGAU=0
109.             DO IGAUS=1,NGAUS
110.                DO JGAUS=1,NGAUS
111.                   KPGAU=KPGAU+1
112.                   EXISP=POSPG (IG AUS)
113.                   ETASP=POSPG (JGAUS)
114.             C
115.             C*** CALCULA FUNCIONES DE FORMA EN PUNTOS DE INTEGRACION
116.             C
117.                CALL FFORMA (EXISP,ETASP,0,UNOVA,NNFRO,FFORM,DERIV)
118.             C
119.             C*** CALCULA EL JACOBIANO
120.             C
121.                DO IDIME=1,3
122.                   DO JDIME=1,3
123.                      ZJACC (IDIME,JDIME) =0.0
124.                      DO INFRO=1,NNFRO
125.                         ZJACC (IDIME,JDIME) =ZJACC (IDIME,JDIME) +
126.                         .                         DERIV (IDIME,INFRO) *COREL (JDIME,LOCAL (INFRO))
127.                         ENDDO
128.                      ENDDO
129.                   ENDDO
130.                   DAREA=SQRT (
131.                      .                   (ZJACC (1,2) *ZJACC (2,3) -ZJACC (1,3) *ZJACC (2,2) ) **2+
132.                      .                   (ZJACC (1,3) *ZJACC (2,1) -ZJACC (1,1) *ZJACC (2,3) ) **2+
133.                      .                   (ZJACC (1,1) *ZJACC (2,2) -ZJACC (1,2) *ZJACC (2,1) ) **2)
134.                      .                   *ALPHA
135.             C*** ARMA VECTOR DE LAS FUNCIONES DE FORMA CON CEROS EN LOS NODOS
136.             C*** NO INCLUIDOS EN EL LADO
137.                DO INFRO=1,NNFRO
138.                   XFORM (LOCAL (INFRO) ) =FFORM (INFRO)
139.                ENDDO
140.             C
141.             C*** ARMA LA MATRIZ DE APORTACION
142.             C
143.                DO IEVAB=1,NEVAB
144.                   DO JEVAB=IEVAB,NEVAB
145.                      FTFOR (IEVAB,JEVAB) =FTFOR (IEVAB,JEVAB) +
146.                      .                      XFORM (IEVAB) *XFORM (JEVAB) *DAREA
147.                      ENDDO
148.                   ENDDO
149.                ENDDO
150.                ENDDO
151.                ENDDO
152.                ENDDO
153.                RETURN
154.                END

```

```

1.      SUBROUTINE DATOS
2.      C*****
3.      C
4.      C***      LEE DATOS DE LA TIPOLOGIA DE LA MALLA, CONDICIONES DE
5.      C          CONTORNO Y PROPIEDADES DEL MATERIAL
6.      C
7.      C*****
8.      C*****data definition
9.      IMPLICIT NONE
10.     INTEGER*2      IWRTIT, IELEM, NUMEL, INODE, NNOD1, NNOD2, IPNOD,
11.     .              IDIME, JPNOD, IPRES, IGDLN, IMATS, NUMAT, IPROP,
12.     .              NNCIT, INCIT, IFRON, INFRO
13.     REAL*4        CITIM, X
14.     INTEGER*2     NN, MNODE, IIODR, NNON, ICONT, IIODE, IILEM, IX,
15.     .              IINOD, IIIME
16.     INCLUDE 'dtsgral.f'
17.     INCLUDE 'data.f'
18.     DIMENSION     X(3,500), IX(24,200), NN(12)
19.     STRING FILENAME5, FILENAME
20.     INTEGER*2     ERROR
21.     C*****data definition
22.     C
23.     C
24.     C          TIPO DE PROBLEMA
25.     C
26.     C          1  TRANSMISION DE CALOR
27.     C
28.     C          TIPOS DE ELEMENTOS
29.     C
30.     C          +  ELEMENTO TRIANGULAR DE 3 NODOS
31.     C          +  ELEMENTO TRIANGULAR DE 6 NODOS
32.     C          +  ELEMENTO CUADRILATERO LAGRANGIANO DE 4 NODOS
33.     C          +  ELEMENTO CUADRILATERO LAGRANGIANO DE 9 NODOS
34.     C          +  ELEMENTO CUADRILATERO SERENDIPITO DE 8 NODOS
35.     C          +  ELEMENTO HEXAGONAL SERENDIPITO DE 20 NODOS
36.     C
37.     C          NTRAN=0 ESTACIONARIO
38.     C          NTRAN=1 TRANSITORIO (SI ALFAT=0 NO DIAGONALIZA)
39.     C          NTRAN=2 TRANSITORIO (SI ALFAT=0 DIAGONALIZA SUMA RENGLONES)
40.     C          NTRAN=3 TRANSITORIO (SI ALFAT=0 DIAGONALIZA MASA EQUIV.)
41.     C
42.     C
43.     C***  LEE PARAMETROS DE CONTROL
44.     C
45.     READ(5,900,ERR=4,IOSTAT=ERROR)
46.     .NPNOD, NELEM, NPRES, NTRAN, MNODE, NMATS, NGAUS, NDIME, IWRTIT
47.     C  ASIGNA VALORES DE DEFAULT
48.     NTIPO=1
49.     NGDLN=1
50.     NTENS=NDIME
51.     IF (NTRAN.GT.0) THEN
52.       NPROP=NDIME+3
53.     ELSE
54.       NPROP=NDIME+2
55.     ENDIF
56.     900 FORMAT(13I5)
57.     NEVAB=NGDLN*NNODE
58.     WRITE(6,905) NPNOD, NELEM, NPRES, NTRAN, NTIPO, MNODE, NGDLN, NMATS,
59.     .           NPROP, NGAUS, NDIME, NTENS, NEVAB, IWRTIT
60.     905 FORMAT(2X, ' NPNOD =', I4, 4X, ' NELEM =', I4, 4X, ' NPRES =', I4,
61.     .           4X, ' NTRAN =', I4, 4X, '*NTIPO =', I4, //,
62.     .           2X, ' MNODE =', I4, 4X, '*NGDLN =', I4, 4X, ' NMATS =', I4,
63.     .           4X, '*NPROP =', I4, 4X, ' NGAUS =', I4, //,

```



```

64.      .      2X, ' NDIME =', I4, 4X, '*NTENS =', I4, 4X, '*NEVAB =', I4,
65.      .      4X, ' IWRIT =', I4)
66.      C
67.      C*** LEE Y ESCRIBE CONEX. NODALES Y NUM DE PROPIEDADES DE MATERIAL
68.      C
69.      WRITE(6,910)
70.      910 FORMAT(// 'ELEMENTO', 2X, 'PROPIEDAD', 4X, 'NUMERO DE NODOS')
71.      IF(NNODE.LE.13) THEN
72.          DO IELEM=1,NELEM
73.              READ(5,903,ERR=4,IOSTAT=ERROR)
74.              NUMEL,MATNU(NUMEL), (LNODS(NUMEL, INODE), INODE=1, NNODE)
75.      903   FORMAT(15I5)
76.          ENDDO
77.      ELSE
78.          NNOD1=13
79.          NNOD2=14
80.          DO IELEM=1,NELEM
81.              READ(5,903,ERR=4,IOSTAT=ERROR)
82.              NUMEL,MATNU(NUMEL), (LNODS(NUMEL, INODE), INODE=1, NNOD1)
83.              READ(5,904,ERR=4,IOSTAT=ERROR)
84.              (LNODS(NUMEL, INODE), INODE=NNOD2, NNODE)
85.      904   FORMAT(10X,15I5)
86.          ENDDO
87.      ENDIF
88.      IF(IWRIT.EQ.1) THEN
89.          IF(NNODE.LE.13) THEN
90.              DO IELEM=1,NELEM
91.              WRITE(6,915) IELEM,MATNU(IELEM), (LNODS(IELEM, INODE), INODE=1,
92.              NNODE)
93.      915   FORMAT(1X,I5,I9,5X,10I5)
94.          ENDDO
95.      ELSE
96.          NNOD1=10
97.          NNOD2=11
98.          DO IELEM=1,NELEM
99.          WRITE(6,915) IELEM,MATNU(IELEM), (LNODS(IELEM, INODE), INODE=1,
100.          NNOD1)
101.          WRITE(6,916) (LNODS(IELEM, INODE), INODE=NNOD2, NNODE)
102.      916   FORMAT(20X,10I5)
103.          ENDDO
104.      ENDIF
105.      ENDIF
106.      DO IPNOD=1,NPNOD
107.          DO IDIME=1,NDIME
108.              COORD(IPNOD, IDIME)=0.0
109.          ENDDO
110.      ENDDO
111.      C
112.      C*** LEE Y ESCRIBE COORDENADAS NODALES
113.      C
114.      DO JPNOD=1,NPNOD
115.          READ(5,930,ERR=4,IOSTAT=ERROR)
116.          IPNOD, (COORD(IPNOD, IDIME), IDIME=1, NDIME)
117.      ENDDO
118.      930 FORMAT(I5,5F10.5)
119.      IF(IWRIT.EQ.1) THEN
120.          WRITE(6,920)
121.      920   FORMAT(// 'COORDENADAS DE PUNTOS NODALES')
122.          WRITE(6,925)
123.      925   FORMAT('  NODO', 7X, 'X', 9X, 'Y', 9X, 'Z')
124.          DO IPNOD=1,NPNOD
125.              WRITE(6,935) IPNOD, (COORD(IPNOD, IDIME), IDIME=1, NDIME)
126.          ENDDO

```

```

127.    935    FORMAT(1X,I5,3F10.3)
128.    ENDIF
129.    C
130.    C*** LEE Y ESCRIBE MOVIMIENTOS PRESCRITOS
131.    C
132.    DO IPRES=1,NPRES
133.    READ(5,950)
134.    .    NODPR(IPRES), (INPRE(IPRES,IGDLN),IGDLN=1,NGDLN),
135.    .    (PRESC(IPRES,IGDLN),IGDLN=1,NGDLN)
136.    ENDDO
137.    950    FORMAT(1X,I4,3X,<ngdln>I1,<ngdln>F10.5)
138.    IF(IWRIT.EQ.1) THEN
139.    WRITE(6,940)
140.    940    FORMAT(//,'NODOS RESTRINGIDOS Y PRESCRIPCIONES')
141.    WRITE(6,945)
142.    945    FORMAT('NODO',1X,'CODIGO',3X,'VALORES PRESCRITOS')
143.    DO IPRES=1,NPRES
144.    WRITE(6,950) NODPR(IPRES), (INPRE(IPRES,IGDLN),IGDLN=1,NGDLN),
145.    .    (PRESC(IPRES,IGDLN),IGDLN=1,NGDLN)
146.    ENDDO
147.    ENDIF
148.    C
149.    C*** LEE Y ESCRIBE PROPIEDADES DE LOS MATERIALES
150.    C
151.    DO IMATS=1,NMATS
152.    READ(5,990,ERR=4,IOSTAT=ERROR)
153.    .    NUMAT, (PROPS(NUMAT,IPROP),IPROP=1,NPROP)
154.    ENDDO
155.    990    FORMAT(1X,I4,5E15.5)
156.    IF(IWRIT.EQ.1) THEN
157.    WRITE(6,980)
158.    980    FORMAT(//,'PROPIEDADES DE LOS MATERIALES')
159.    WRITE(6,985)
160.    985    FORMAT('NUMERO',7X,'PROPIEDADES')
161.    DO IMATS=1,NMATS
162.    WRITE(6,991) IMATS, (PROPS(IMATS,IPROP),IPROP=1,NPROP)
163.    ENDDO
164.    991    FORMAT(1X,I4,<NDIME>('    K=',E15.2),/,
165.    .    5x,' ALPHA=',E15.2,'    ROR=',E15.2,'    ROC=',E15.2,/)
166.    ENDIF
167.    C
168.    C*** LEE Y ESCRIBE DATOS DEL PROBLEMA TRANSITORIO
169.    C
170.    IF (NTRAN.GT.0) THEN
171.    READ(5,992,ERR=4,IOSTAT=ERROR) NTIME,DTIME,ALFAT
172.    992    FORMAT(I5,2E15.5)
173.    IF(IWRIT.EQ.1) THEN
174.    WRITE(6,993) NTIME,DTIME,ALFAT
175.    993    FORMAT(1X,' NUMERO DE PASOS DE TIEMPO.....',I5,/,
176.    .    1X,' INCREMENTO DE TIEMPO.....',E15.5,/,
177.    .    1X,' ALFA DE INTEGRACION TEMPORAL..',E15.5,/)
178.    ENDIF
179.    C
180.    C*** LECTURA DE LA CONDICION INICIAL DE TIEMPO
181.    C
182.    DO IPNOD=1,NPNOD
183.    XTIME(IPNOD)=0.0
184.    ENDDO
185.    READ(5,992,ERR=4,IOSTAT=ERROR) NNCIT,CITIM
186.    IF (NNCIT.EQ.NPNOD) THEN
187.    IF (CITIM.NE.0) THEN
188.    DO IPNOD=1,NPNOD
189.    XTIME(IPNOD)=CITIM

```

```

190.         ENDDO
191.     ENDIF
192. ELSE
193.     DO INCIT=1,NNCIT
194.         READ(5,992) IPNOD,XTIME(IPNOD)
195.     ENDDO
196. ENDIF
197. IF(IWRIT.EQ.1) THEN
198.     WRITE(6,994)
199.     DO IPNOD=1,NPNOD
200.         WRITE(6,995) IPNOD,XTIME(IPNOD)
201.     ENDDO
202. 994     FORMAT (1X,'CONDICIONES INICIALES DE TEMPERATURA',/,
203.             1X,'NODO  TEMP.o')
204. 995     FORMAT (1X,I5,1X,E15.5)
205.     ENDIF
206. ENDIF
207. C
208. C*** LEE Y ESCRIBE CONDICIONES DE FRONTERA
209. C*** DE FLUJO POR CONVECCION RADIACION
210. C*** NFRON => NUMERO DE ELEMENTOS DE FRONTERA
211. C*** NNFRO => NUMERO DE NODOS DE FRONTERA
212. C*** CONVE(NFRON,1) ELEMENTO NUMERO
213. C*** CONVE(NFRON,<>1) NODOS DE FRONTERA
214. C
215.     READ(5,996,ERR=4,IOSTAT=ERROR) NFRON,NNFRO
216.     IF (IWRIT.EQ.1)
217.         WRITE(6,997) NFRON,NNFRO
218.     IF(NFRON.NE.0) THEN
219.         IF (IWRIT.EQ.1) WRITE (6,998)
220.         DO IFRON=1,NFRON
221.             READ(5,999,ERR=4,IOSTAT=ERROR)
222.             (CONVE(IFRON,INFRO),INFRO=1,NNFRO+1)
223.             IF (IWRIT.EQ.1)
224.                 WRITE(6,1000) (CONVE(IFRON,INFRO),INFRO=1,NNFRO+1)
225.         ENDDO
226.     ENDIF
227. 996 FORMAT (1X,2I4)
228. 997 FORMAT(1X,'FRONTERAS CON FLUJO DE CONVECCION/RADIACION =',I4,
229.         /,'NUMERO DE NODOS POR FRONTERA =',I4)
230. 998 FORMAT (1X,'ELEMENTO  N O D O S')
231. 999 FORMAT (1X,<NNFRO+1>I4)
232. 1000 FORMAT (3X,I4,4X,<NNFRO>(I4,1X))
233. C
234. C*** ESCRITURA PARA POST-PROCESO
235. C
236.     MNODE=NNODE
237.     IF(NNODE.EQ.6) MNODE=8
238.     IIODR=(MNODE+4)
239.     IF((NNODE.EQ.6).OR.(NNODE.EQ.8).OR.(NNODE.EQ.9)) THEN
240.         NNON=MNODE
241.         IF(NNODE.EQ.6) THEN
242.             DO IELEM=1,NELEM
243.                 IX(1,IELEM)=(LNODS(IELEM,1))
244.                 IX(2,IELEM)=(LNODS(IELEM,3))
245.                 IX(3,IELEM)=(LNODS(IELEM,5))
246.                 IX(4,IELEM)=(LNODS(IELEM,1))
247.                 IX(5,IELEM)=(LNODS(IELEM,2))
248.                 IX(6,IELEM)=(LNODS(IELEM,4))
249.                 IX(7,IELEM)=(LNODS(IELEM,6))
250.                 IX(8,IELEM)=(LNODS(IELEM,1))
251.                 IX(12,IELEM)=(MATNU(IELEM))
252.             ENDDO

```

```

253.      ELSE
254.          IF (MNODE.EQ.9) NNON=8
255.          DO IELEM=1, NELEM
256.              ICONT=0
257.              DO INODE=1, NNON, 2
258.                  ICONT=ICONT+1
259.                  IX ( ICONT, IELEM) = (LNODS ( IELEM, INODE) )
260.              ENDDO
261.              DO INODE=2, NNON, 2
262.                  ICONT=ICONT+1
263.                  IX ( ICONT, IELEM) = (LNODS ( IELEM, INODE) )
264.              ENDDO
265.              IF (NNODE.EQ.9) IX (9, IELEM) = (LNODS ( IELEM, 9) )
266.              IX ( I IODR, IELEM) = (MATNU ( IELEM) )
267.          ENDDO
268.      ENDIF
269.      ELSE
270.          IF (NNODE.EQ.20) THEN
271.              DO IELEM=1, NELEM
272.                  IX (1, IELEM) = (LNODS ( IELEM, 1) )
273.                  IX (2, IELEM) = (LNODS ( IELEM, 3) )
274.                  IX (3, IELEM) = (LNODS ( IELEM, 5) )
275.                  IX (4, IELEM) = (LNODS ( IELEM, 7) )
276.                  IX (5, IELEM) = (LNODS ( IELEM, 13) )
277.                  IX (6, IELEM) = (LNODS ( IELEM, 15) )
278.                  IX (7, IELEM) = (LNODS ( IELEM, 17) )
279.                  IX (8, IELEM) = (LNODS ( IELEM, 19) )
280.                  IX (9, IELEM) = (LNODS ( IELEM, 2) )
281.                  IX (10, IELEM) = (LNODS ( IELEM, 4) )
282.                  IX (11, IELEM) = (LNODS ( IELEM, 6) )
283.                  IX (12, IELEM) = (LNODS ( IELEM, 8) )
284.                  IX (13, IELEM) = (LNODS ( IELEM, 9) )
285.                  IX (14, IELEM) = (LNODS ( IELEM, 10) )
286.                  IX (15, IELEM) = (LNODS ( IELEM, 11) )
287.                  IX (16, IELEM) = (LNODS ( IELEM, 12) )
288.                  IX (17, IELEM) = (LNODS ( IELEM, 14) )
289.                  IX (18, IELEM) = (LNODS ( IELEM, 16) )
290.                  IX (19, IELEM) = (LNODS ( IELEM, 18) )
291.                  IX (20, IELEM) = (LNODS ( IELEM, 20) )
292.                  IX (24, IELEM) = (MATNU ( IELEM) )
293.              ENDDO
294.          ELSE
295.              DO IELEM=1, NELEM
296.                  DO INODE=1, MNODE
297.                      IX ( INODE, IELEM) = (LNODS ( IELEM, INODE) )
298.                  ENDDO
299.                  IX ( I IODR, IELEM) = (MATNU ( IELEM) )
300.              ENDDO
301.          ENDIF
302.      ENDIF
303.      DO IPNOD=1, NPNOD
304.          DO IDIME=1, NDIME
305.              X ( IDIME, IPNOD) =COORD ( IPNOD, IDIME)
306.          ENDDO
307.      ENDDO
308.      NN (1) = (NPNOD)
309.      NN (2) = (NELEM)
310.      NN (3) = (NPRES)
311.      IF (NTRAN.EQ.0) THEN
312.          NN (4) =1
313.      ELSE
314.          NN (4) =NTIME
315.      ENDIF

```



```

316.      NN(5) = (NTIPO)
317.      NN(6) = (MNODE)
318.      NN(7) = (NGDLN)
319.      NN(8) = (NMATS)
320.      NN(9) = (NGAUS)
321.      NN(10) = (NDIME)
322.      NN(11) = (NTENS)
323.      NN(12) = (NEVAB)
324.      WRITE(29) (NN(IIODE), IIODE=1, 12)
325.      WRITE(29) (IILEM, (IX(IIODE, IILEM), IIODE=1, IIODR),
326.      .
327.      WRITE(29) (IINOD, (X(IIIME, IINOD), IIIME=1, NN(10)),
328.      .
329.      GO TO 444
330. 4 CLOSE(1, STATUS='DELETE')
331.      CLOSE(3, STATUS='DELETE')
332.      CLOSE(4, STATUS='DELETE')
333.      CLOSE(10, STATUS='DELETE')
334.      CLOSE(29, STATUS='KEEP')
335.      CLOSE(5, STATUS='KEEP')
336.      FILENAME='ERROR NO LOCALIZADO'
337.      IF (ERROR.EQ.-1) FILENAME='FIN PREMATURO DEL FICHERO '
338.      IF (ERROR.EQ.41) FILENAME='NO HAY SUFICIENTE MEMORIA '
339.      IF (ERROR.EQ.62) FILENAME='CAMPO INVALIDO DE LECTURA '
340.      WRITE(6, *) 'ERROR EN LECTURA DE DATOS ', ERROR
341.      CLOSE(6, STATUS='KEEP')
342.      FILENAME5=FILENAME
343.      CALL ALERTBOX(FILENAME5)
344.      STOP
345. 444 RETURN
346.      END

1.      SUBROUTINE DBMATX
2.      C*****
3.      C
4.      C***      MULTIPLICA MATRICES D y B
5.      C
6.      C*****
7.      IMPLICIT NONE
8.      INTEGER*2      ITENS, IEVAB, JTENS
9.      INCLUDE 'dtsgral.f'
10.     INCLUDE 'calculo.f'
11.     C*****
12.     C
13.     C***      CALCULA D X B
14.     C
15.     DO ITENS=1, NTENS
16.         DO IEVAB=1, NEVAB
17.             DBMAT(ITENS, IEVAB)=0.0
18.             DO JTENS=1, NTENS
19.                 DBMAT(ITENS, IEVAB)=DBMAT(ITENS, IEVAB)+
20.                 .
21.                 DMATZ(ITENS, JTENS)*BMATZ(JTENS, IEVAB)
22.             ENDDO
23.         ENDDO
24.     ENDDO
25.     RETURN
26.     END

```

```

1.      SUBROUTINE DIAGONAL
2.      C*****
3.      C
4.      C***      DIAGONALIZA LA MATRIZ DE RIGIDEZ PARA ALFAT=0
5.      C          (CASO EXPLICITO)
6.      C          NTRAN = 1 NO DIAGONALIZA
7.      C          NTRAN = 2 DIAGONALIZA POR SUMAS
8.      C          NTRAN = 3 DIAGONALIZA POR MASA EQUIVALENTE
9.      C*****
10.     IMPLICIT NONE
11.     INCLUDE 'dtsgral.f'
12.     INCLUDE 'solu.f'
13.     INTEGER*2      NEONS, IEONS, IROWS
14.     REAL*4         B, XMASS
15.     C*****
16.     NEONS=NSVAB
17.     C
18.     C*** DIAGONALIZACION POR SUMA DE RENGLONES
19.     C*
20.     IF (NTRAN.EQ.2) THEN
21.     DO IEONS=1, NEONS
22.     B=0.0
23.     DO IROWS=1, NEONS
24.     B=B+ASTIF (IEONS, IROWS)
25.     ASTIF (IEONS, IROWS)=0.0
26.     ENDDO
27.     ASTIF (IEONS, IEONS)=B
28.     ENDDO
29.     ENDIF
30.     C
31.     C*** DIAGONALIZACION POR MASA EQUIVALENTE
32.     C
33.     IF (NTRAN.EQ.3) THEN
34.     XMASS=0.0
35.     DO IEONS=1, NEONS
36.     XMASS=XMASS+ASTIF (IEONS, IEONS)
37.     ENDDO
38.     XLUMP=XLUMP/XMASS
39.     DO IEONS=1, NEONS
40.     XMASS=ASTIF (IEONS, IEONS)
41.     DO IROWS=1, NEONS
42.     ASTIF (IEONS, IROWS)=0.0
43.     ENDDO
44.     ASTIF (IEONS, IEONS)=XMASS*XLUMP
45.     ENDDO
46.     ENDIF
47.     RETURN
48.     END

1.      SUBROUTINE DMAT(LPROP)
2.      C*****
3.      C
4.      C***      MATRIZ CONSTITUTIVA D
5.      C
6.      C*****data definition
7.      IMPLICIT NONE
8.      INTEGER*2      ITENS, JTENS, LPROP
9.      INCLUDE 'dtsgral.f'
10.     INCLUDE 'data.f'
11.     INCLUDE 'calculo.f'
12.     C*****data definition
13.     DO ITENS=1, NTENS

```

```

14.          DO JTENS=1,NTENS
15.              DMATZ (ITENS, JTENS) =0.0
16.          ENDDO
17.      ENDDO
18.      IF (NTIPO.EQ.1) THEN
19.      C
20.      C***      MATRIZ D PARA TRANSMISION DE CALOR
21.      C
22.          DO ITENS=1,NTENS
23.              DMATZ (ITENS, ITENS) =PROPS (LPROP, ITENS)
24.          ENDDO
25.      ELSE
26.          WRITE (6,900) NTIPO
27.      900      FORMAT (1X, 'TIPO DE PROBLEMA ', I3, ' NO IMPLEMENTADO')
28.      ENDIF
29.      RETURN
30.      END

1.          SUBROUTINE ENSAMBLA
2.      C*****
3.      C
4.      C*** ENSAMBLA LAS MATRICES DE RIGIDEZ Y LOS VECTORES DE FUERZAS
5.      C
6.      C*****
7.          IMPLICIT NONE
8.          INCLUDE 'dtsgral.f'
9.          INCLUDE 'data.f'
10.         INCLUDE 'calculo.f'
11.         INCLUDE 'solu.f'
12.         INTEGER*2 NTOTV, ITOTV, JTOTV, IPRES, NLOCA, IGDLN, NCONT, IELEM,
13.         .           INODE, NODEI, NROWS, NROWE, JNODE, NODEJ, JGDLN, NCOLS,
14.         .           NCOLE, ICARG
15.         INTEGER*2 MTOTV, JFFIX
16.         REAL*4     B, ZMASA
17.         DIMENSION JFFIX(200), ZMASA(60,60)
18.      C*****
19.         NTOTV=NPNOD*NGDLN
20.         MTOTV=(NTOTV)
21.         REWIND 1
22.         IF (NTRAN.GT.0) REWIND 4
23.         DO ITOTV=1,NTOTV
24.             ASLOD(ITOTV)=0.0
25.             DESPL(ITOTV)=0.0
26.             FIXED(ITOTV)=0
27.             IFFIX(ITOTV)=0
28.             DO JTOTV=1,NTOTV
29.                 ASTIF(ITOTV, JTOTV) =0.0
30.             ENDDO
31.         ENDDO
32.      C
33.      C*** ENSAMBLA VECTOR DE FUERZAS ELEMENTALES
34.      C
35.         NSVAB=1
36.         DO IPRES=1,NPRES
37.             NLOCA=(NODPR(IPRES)-1)*NGDLN
38.             DO IGDLN=1,NGDLN
39.                 NCONT=NLOCA+IGDLN
40.                 IFFIX(NCONT) =INPRE(IPRES, IGDLN)
41.                 FIXED(NCONT) =PRESC(IPRES, IGDLN)
42.             ENDDO
43.         ENDDO
44.         DO IELEM=1,NELEM

```

```

45.      READ(1) RIGID
46.      IF (NTRAN.GT.0) READ(4) ZMASA
47.      DO INODE=1, NNODE
48.          NODEI=LNODS(IELEM, INODE)
49.          DO IGDLN=1, NGDLN
50.              NROWS= ( (NODEI-1) *NGDLN) +IGDLN
51.              NROWE= ( (INODE-1) *NGDLN) +IGDLN
52.              ASLOD(NROWS)=ASLOD(NROWS)+CARGA(IELEM, NROWE)
53.              IF(NROWS.GT.NSVAB) NSVAB=NROWS
54.          C
55.      C***  ENSAMBLA LAS MATRICES DE RIGIDECES ELEMENTALES
56.      C
57.          DO JNODE=1, NNODE
58.              NODEJ=LNODS(IELEM, JNODE)
59.              DO JGDLN=1, NGDLN
60.                  NCOLS= (NODEJ-1) *NGDLN+JGDLN
61.                  NCOLE= (JNODE-1) *NGDLN+JGDLN
62.                  B=RIGID(NROWE, NCOLE)
63.                  IF (NTRAN.GT.0) B=B*ALFAT+ZMASA(NROWE, NCOLE)
64.                  ASTIF(NROWS, NCOLS)=ASTIF(NROWS, NCOLS)+B
65.              ENDDO
66.          ENDDO
67.      ENDDO
68.      ENDDO
69.      ENDDO
70.      DO ITOTV=1, NTOTV
71.          IF(IFFIX(ITOTV).EQ.0) THEN
72.              JFFIX(ITOTV)=1
73.          ELSE
74.              JFFIX(ITOTV)=0
75.          ENDIF
76.      ENDDO
77.      IF (NTRAN.GT.0) WRITE(4) ASLOD
78.      WRITE(29) MTOTV, (JFFIX(ITOTV), ITOTV=1, NTOTV)
79.      WRITE(29) MTOTV, (ASLOD(ITOTV), ITOTV=1, NTOTV)
80.      RETURN
81.      END

1.      SUBROUTINE FFORMA(S, T, Q, NDIME, NNODE, FFORM, DERIV)
2.      C*****
3.      C
4.      C***  FUNCIONES DEFORMA DE LOS ELEMENTOS
5.      C
6.      IMPLICIT NONE
7.      REAL*4 S, T, Q, S1, T1, S2, T2, Q2, SS, TT, QQ, ST, SST, STT, ST2,
8.          S9, T9
9.      INTEGER*2 NDIME, NNODE
10.     REAL*4 FFORM, DERIV
11.     DIMENSION DERIV(3, 20), FFORM(20)
12.     C*****
13.     S1=S+1.0
14.     T1=T+1.0
15.     S2=S*2.0
16.     T2=T*2.0
17.     Q2=Q*2.0
18.     SS=S*S
19.     TT=T*T
20.     QQ=Q*Q
21.     ST=S*T
22.     SST=S*S*T
23.     STT=S*T*T
24.     ST2=S*T*2.0

```



```

25.      S9=S-1.0
26.      T9=T-1.0
27.      IF (NDIME.EQ.1) THEN
28.          IF (NNODE.EQ.2) THEN
29.              C
30.              C***      ELEMENTO UNIDIMENSIONAL DE DOS NODOS
31.              C
32.                  FFORM(1)=(1-S)/2.0
33.                  FFORM(2)=(1+S)/2.0
34.                  DERIV(1,1)=-1/2.0
35.                  DERIV(1,2)=1/2.0
36.              ENDIF
37.          IF (NNODE.EQ.3) THEN
38.              C
39.              C***      ELEMENTO UNIDIMENSIONAL DE TRES NODOS
40.              C
41.                  FFORM(1)=(S-1)*S/2.0
42.                  FFORM(2)=(1+S)*(1-S)
43.                  FFORM(3)=(1+S)*S/2.0
44.                  DERIV(1,1)=S-1/2.0
45.                  DERIV(1,2)=-2.0*S
46.                  DERIV(1,3)=S+1/2.0
47.              ENDIF
48.          ENDIF
49.      IF (NDIME.EQ.2) THEN
50.          IF (NNODE.EQ.3) THEN
51.              C
52.              C***      ELEMENTO TRIANGULAR DE 3 NODOS
53.              C
54.                  FFORM(1)=1.0-S-T
55.                  FFORM(2)=S
56.                  FFORM(3)=T
57.                  DERIV(1,1)=-1.0
58.                  DERIV(1,2)=1.0
59.                  DERIV(1,3)=0.0
60.                  DERIV(2,1)=-1.0
61.                  DERIV(2,2)=0.0
62.                  DERIV(2,3)=1.0
63.              ENDIF
64.          IF (NNODE.EQ.4) THEN
65.              C
66.              C***      ELEMENTO CUADRILATERO DE 4 NODOS
67.              C
68.                  FFORM(1)=(1.0-S-T+ST)/4.0
69.                  FFORM(2)=(1.0+S-T-ST)/4.0
70.                  FFORM(3)=(1.0+S+T+ST)/4.0
71.                  FFORM(4)=(1.0-S+T-ST)/4.0
72.                  DERIV(1,1)=(-1+T)/4.0
73.                  DERIV(1,2)=(1-T)/4.0
74.                  DERIV(1,3)=(1+T)/4.0
75.                  DERIV(1,4)=(-1-T)/4.0
76.                  DERIV(2,1)=(-1+S)/4.0
77.                  DERIV(2,2)=(-1-S)/4.0
78.                  DERIV(2,3)=(1+S)/4.0
79.                  DERIV(2,4)=(1-S)/4.0
80.              ENDIF
81.          IF (NNODE.EQ.6) THEN
82.              C
83.              C***      ELEMENTO TRIANGULAR DE 6 NODOS
84.              C
85.                  FFORM(1)=1.0-(S+T)*3.0+(SS+TT)*2.0+ST*4.0
86.                  FFORM(2)=(S-SS-ST)*4.0
87.                  FFORM(3)=SS*2.0-S

```

```

88.          FFORM(4)=4.0*ST
89.          FFORM(5)=TT*2.0-T
90.          FFORM(6)=(T-TT-ST)*4.0
91.          DERIV(1,1)=-3.0+S*4.0+T*4.0
92.          DERIV(1,2)=(1.0-S*2.0-T)*4.0
93.          DERIV(1,3)=S*4.0-1.0
94.          DERIV(1,4)=T*4.0
95.          DERIV(1,5)=0.0
96.          DERIV(1,6)=-T*4.0
97.          DERIV(2,1)=-3.0+T*4.0+S*4.0
98.          DERIV(2,2)=-S*4.0
99.          DERIV(2,3)=0.0
100.         DERIV(2,4)=4.0*S
101.         DERIV(2,5)=T*4.0-1.0
102.         DERIV(2,6)=(1.0-T*2.0-S)*4.0
103.         ENDIF
104.         IF(NNODE.EQ.8) THEN
105.         C
106.         C***      ELEMENTO CUADRILATERO SERENDIPITO DE 8 NODOS
107.         C
108.             FFORM(1)=(-1.0+ST+SS+TT-SST-STT)/4.0
109.             FFORM(2)=(1.0-T-SS+SST)/2.0
110.             FFORM(3)=(-1.0-ST+SS+TT-SST+STT)/4.0
111.             FFORM(4)=(1.0+S-TT-STT)/2.0
112.             FFORM(5)=(-1.0+ST+SS+TT+SST+STT)/4.0
113.             FFORM(6)=(1.0+T-SS-SST)/2.0
114.             FFORM(7)=(-1.0-ST+SS+TT+SST-STT)/4.0
115.             FFORM(8)=(1.0-S-TT+STT)/2.0
116.             DERIV(1,1)=(T+S2-ST2-TT)/4.0
117.             DERIV(1,2)=-S+ST
118.             DERIV(1,3)=(-T+S2-ST2+TT)/4.0
119.             DERIV(1,4)=(1.0-TT)/2.0
120.             DERIV(1,5)=(T+S2+ST2+TT)/4.0
121.             DERIV(1,6)=-S-ST
122.             DERIV(1,7)=(-T+S2+ST2-TT)/4.0
123.             DERIV(1,8)=(-1.0+TT)/2.0
124.             DERIV(2,1)=(S+T2-SS-ST2)/4.0
125.             DERIV(2,2)=(-1.0+SS)/2.0
126.             DERIV(2,3)=(-S+T2-SS+ST2)/4.0
127.             DERIV(2,4)=-T-ST
128.             DERIV(2,5)=(S+T2+SS+ST2)/4.0
129.             DERIV(2,6)=(1.0-SS)/2.0
130.             DERIV(2,7)=(-S+T2+SS-ST2)/4.0
131.             DERIV(2,8)=-T+ST
132.         ENDIF
133.         IF(NNODE.EQ.9) THEN
134.         C
135.         C***      ELEMENTO CUADRILATERO LAGRANGIANO DE 9 NODOS
136.         C
137.             FFORM(1)=0.25*S9*ST*T9
138.             FFORM(2)=0.5*(1.0-SS)*T*T9
139.             FFORM(3)=0.25*S1*ST*T9
140.             FFORM(4)=0.5*S*S1*(1.0-TT)
141.             FFORM(5)=0.25*S1*ST*T1
142.             FFORM(6)=0.5*(1.0-SS)*T*T1
143.             FFORM(7)=0.25*S9*ST*T1
144.             FFORM(8)=0.5*S*S9*(1.0-TT)
145.             FFORM(9)=(1.0-SS)*(1.0-TT)
146.             DERIV(1,1)=0.25*T*T9*(-1.0+S2)
147.             DERIV(1,2)=-ST*T9
148.             DERIV(1,3)=0.25*(1.0+S2)*T*T9
149.             DERIV(1,4)=0.5*(1.0+S2)*(1.0-TT)
150.             DERIV(1,5)=0.25*(1.0+S2)*T*T1

```

```

151.          DERIV(1,6)=-ST*T1
152.          DERIV(1,7)=0.25*(-1.0+S2)*T*T1
153.          DERIV(1,8)=0.5*(-1.0+S2)*(1.0-TT)
154.          DERIV(1,9)=-S2*(1.0-TT)
155.          DERIV(2,1)=0.25*(-1.0+T2)*S*S9
156.          DERIV(2,2)=0.5*(1.0-SS)*(-1.0+T2)
157.          DERIV(2,3)=0.25*S*S1*(-1.0+T2)
158.          DERIV(2,4)=-ST*S1
159.          DERIV(2,5)=0.25*S*S1*(1.0+T2)
160.          DERIV(2,6)=0.5*(1.0-SS)*(1.0+T2)
161.          DERIV(2,7)=0.25*S*S9*(1.0+T2)
162.          DERIV(2,8)=-ST*S9
163.          DERIV(2,9)=-T2*(1.0-SS)
164.          ENDIF
165.          ENDIF
166.          IF (NDIME.EQ.3) THEN
167.              IF (NNODE.EQ.20) THEN
168.                  C
169.                  C*** ELEMENTO SOLIDO TRIDIM. HEXAGONAL SERENDIPITO DE 20 NODOS
170.                  C
171.                  FFORM(1)=(1+S)*(1-T)*(1-Q)*(S-T-Q-2)*.125
172.                  FFORM(2)=(1-TT)*(1+S)*(1-Q)*.25
173.                  FFORM(3)=(1+S)*(1+T)*(1-Q)*(S+T-Q-2)*.125
174.                  FFORM(4)=(1-SS)*(1+T)*(1-Q)*.25
175.                  FFORM(5)=(1-S)*(1+T)*(1-Q)*(-S+T-Q-2)*.125
176.                  FFORM(6)=(1-TT)*(1-S)*(1-Q)*.25
177.                  FFORM(7)=(1-S)*(1-T)*(1-Q)*(-S-T-Q-2)*.125
178.                  FFORM(8)=(1-SS)*(1-T)*(1-Q)*.25
179.                  FFORM(9)=(1-QQ)*(1+S)*(1-T)*.25
180.                  FFORM(10)=(1-QQ)*(1+S)*(1+T)*.25
181.                  FFORM(11)=(1-QQ)*(1-S)*(1+T)*.25
182.                  FFORM(12)=(1-QQ)*(1-S)*(1-T)*.25
183.                  FFORM(13)=(1+S)*(1-T)*(1+Q)*(S-T+Q-2)*.125
184.                  FFORM(14)=(1-TT)*(1+S)*(1+Q)*.25
185.                  FFORM(15)=(1+S)*(1+T)*(1+Q)*(S+T+Q-2)*.125
186.                  FFORM(16)=(1-SS)*(1+T)*(1+Q)*.25
187.                  FFORM(17)=(1-S)*(1+T)*(1+Q)*(-S+T+Q-2)*.125
188.                  FFORM(18)=(1-TT)*(1-S)*(1+Q)*.25
189.                  FFORM(19)=(1-S)*(1-T)*(1+Q)*(-S-T+Q-2)*.125
190.                  FFORM(20)=(1-SS)*(1-T)*(1+Q)*.25
191.                  DERIV(1,1)=.125*(1-T)*(1-Q)*(S2-T-Q-1)
192.                  DERIV(2,1)=.125*(1+S)*(1-Q)*(-S+T2+Q+1)
193.                  DERIV(3,1)=.125*(1+S)*(1-T)*(-S+T+Q2+1)
194.                  DERIV(1,2)=.25*(1-TT)*(1-Q)
195.                  DERIV(2,2)=.25*(-T2)*(1+S)*(1-Q)
196.                  DERIV(3,2)=.25*(TT-1)*(1+S)
197.                  DERIV(1,3)=.125*(1+T)*(1-Q)*(S2+T-Q-1)
198.                  DERIV(2,3)=.125*(1+S)*(1-Q)*(S+T2-Q-1)
199.                  DERIV(3,3)=.125*(1+S)*(1+T)*(-S-T+Q2+1)
200.                  DERIV(1,4)=.25*(-S2)*(1+T)*(1-Q)
201.                  DERIV(2,4)=.25*(1-SS)*(1-Q)
202.                  DERIV(3,4)=-.25*(1-SS)*(1+T)
203.                  DERIV(1,5)=.125*(1+T)*(1-Q)*(S2-T+Q+1)
204.                  DERIV(2,5)=.125*(1-S)*(1-Q)*(-S+T2-Q-1)
205.                  DERIV(3,5)=.125*(1-S)*(1+T)*(S-T+Q2+1)
206.                  DERIV(1,6)=-.25*(1-TT)*(1-Q)
207.                  DERIV(2,6)=-.25*(T2)*(1-S)*(1-Q)
208.                  DERIV(3,6)=-.25*(1-TT)*(1-S)
209.                  DERIV(1,7)=.125*(1-T)*(1-Q)*(S2+T+Q+1)
210.                  DERIV(2,7)=-.125*(1-S)*(1-Q)*(-S-T2-Q-1)
211.                  DERIV(3,7)=-.125*(1-S)*(1-T)*(-S-T-Q2-1)
212.                  DERIV(1,8)=-.25*(S2)*(1-T)*(1-Q)
213.                  DERIV(2,8)=-.25*(1-SS)*(1-Q)

```



```

214.      DERIV(3,8)=-.25*(1-SS)*(1-T)
215.      DERIV(1,9)=.25*(1-QQ)*(1-T)
216.      DERIV(2,9)=-.25*(1-QQ)*(1+S)
217.      DERIV(3,9)=-.25*(Q2)*(1+S)*(1-T)
218.      DERIV(1,10)=.25*(1-QQ)*(1+T)
219.      DERIV(2,10)=.25*(1-QQ)*(1+S)
220.      DERIV(3,10)=-.25*(Q2)*(1+S)*(1+T)
221.      DERIV(1,11)=-.25*(1-QQ)*(1+T)
222.      DERIV(2,11)=.25*(1-QQ)*(1-S)
223.      DERIV(3,11)=-.25*(Q2)*(1-S)*(1+T)
224.      DERIV(1,12)=-.25*(1-QQ)*(1-T)
225.      DERIV(2,12)=-.25*(1-QQ)*(1-S)
226.      DERIV(3,12)=-.25*(Q2)*(1-S)*(1-T)
227.      DERIV(1,13)=.125*(1-T)*(1+Q)*(S2-T+Q-1)
228.      DERIV(2,13)=-.125*(1+S)*(1+Q)*(S-T2+Q-1)
229.      DERIV(3,13)=.125*(1+S)*(1-T)*(S-T+Q2-1)
230.      DERIV(1,14)=.25*(1-TT)*(1+Q)
231.      DERIV(2,14)=-.25*(T2)*(1+S)*(1+Q)
232.      DERIV(3,14)=.25*(1-TT)*(1+S)
233.      DERIV(1,15)=.125*(1+T)*(1+Q)*(S2+T+Q-1)
234.      DERIV(2,15)=.125*(1+S)*(1+Q)*(S+T2+Q-1)
235.      DERIV(3,15)=.125*(1+S)*(1+T)*(S+T+Q2-1)
236.      DERIV(1,16)=-.25*(S2)*(1+T)*(1+Q)
237.      DERIV(2,16)=.25*(1-SS)*(1+Q)
238.      DERIV(3,16)=.25*(1-SS)*(1+T)
239.      DERIV(1,17)=-.125*(1+T)*(1+Q)*(-S2+T+Q-1)
240.      DERIV(2,17)=.125*(1-S)*(1+Q)*(-S+T2+Q-1)
241.      DERIV(3,17)=.125*(1-S)*(1+T)*(-S+T+Q2-1)
242.      DERIV(1,18)=-.25*(1-TT)*(1+Q)
243.      DERIV(2,18)=-.25*(T2)*(1-S)*(1+Q)
244.      DERIV(3,18)=.25*(1-TT)*(1-S)
245.      DERIV(1,19)=-.125*(1-T)*(1+Q)*(-S2-T+Q-1)
246.      DERIV(2,19)=-.125*(1-S)*(1+Q)*(-S-T2+Q-1)
247.      DERIV(3,19)=.125*(1-S)*(1-T)*(-S-T+Q2-1)
248.      DERIV(1,20)=-.25*(S2)*(1-T)*(1+Q)
249.      DERIV(2,20)=-.25*(1-SS)*(1+Q)
250.      DERIV(3,20)=.25*(1-SS)*(1-T)
251.      ENDIF
252.      ENDIF
253.      RETURN
254.      END

1.      SUBROUTINE FUERZAS
2.      C*****
3.      C
4.      C***          CALCULO DE FUERZAS NODALES EQUIVALENTES
5.      C
6.      C*****
7.      IMPLICIT NONE
8.      INCLUDE 'dtsgral.f'
9.      INCLUDE 'data.f'
10.     INTEGER*2          IELEM, IEVAB, IPUNT, IPESO, IDIST, LPROP,
11.     .                  NEDGE, NODEG, NEASS, IODEG, NOPRS
12.     REAL*4             FLUJO, TEMPE, EGASP, PESPP, PRESS
13.     DIMENSION          PRESS(20,2), NOPRS(9)
14.     CHARACTER*20       TITULO
15.     C*****
16.     EGASP=0.0
17.     C
18.     C*** INICIALIZA VECTOR DE FUERZAS NODALES EQUIVALENTES
19.     C
20.     DO IELEM=1,NELEM

```



```

21.          DO IEVAB=1,NEVAB
22.            CARGA(IELEM,IEVAB)=0.0
23.          ENDDO
24.          ENDDO
25.          READ(5,900) TITULO
26.          900 FORMAT(A20)
27.          WRITE(6,905) TITULO
28.          905 FORMAT(/,1X,A20,/)
29.          C
30.          C*** LEE PARAMETROS DE TIPO DE CARGA
31.          C
32.          READ(5,910) IPUNT,IPESO,IDIST
33.          WRITE(6,911) IPUNT,IPESO,IDIST
34.          910 FORMAT(4I5)
35.          911 FORMAT (1X,'COND. DE CARGA PUNTUAL:.....',I5,/,
36.          .          1X,'COND. DE CARGA DE GENER. INTERNA:....',I5,/,
37.          .          1X,'COND. DE CALOR DISTRIBUIDO POR LADO...',I5)
38.          C
39.          C*** CARGAS PUNTUALES
40.          C
41.          IF(IPUNT.NE.0) CALL PUNTUAL
42.          C
43.          C*** CARGAS POR GENERACION INTERNA DE CALOR
44.          C
45.          IF (IPESO.NE.0) THEN
46.            DO IELEM=1,NELEM
47.              LPROP=MATNU(IELEM)
48.              PESPP=PROPS(LPROP,NPROP)
49.              IF (NTRAN.GT.0) PESPP=PROPS(LPROP,NPROP-1)
50.              IF(PESPP.NE.0.0) CALL PESOP(PESPP,IELEM)
51.            ENDDO
52.          ENDIF
53.          C
54.          C*** CARGAS POR FLUJO Y TEMPERATURA NORMAL REPARTIDAS SOBRE
55.          C      EL LADO DE UN ELEMENTO
56.          C
57.          IF(IDIST.NE.0) THEN
58.            READ(5,930) NEDGE,NODEG
59.            WRITE(6,935) NEDGE,NODEG
60.            WRITE(6,940)
61.            930  FORMAT(2I5)
62.            935  FORMAT(5X,'NUMERO DE LADOS EXPUESTOS=',I5,/,
63.            .      5X,'NUMERO DE NODOS EN LADO  ='I5,///)
64.            940  FORMAT(1X,5X,'ELEMENTO EXPUESTO (NODOS EN NUM. GLOBAL) ',/,
65.            .      'FLUJO ACTUANTE ,TEMPERATURA EXTERNA, ALFA*TEMP/FLUJO')
66.          C
67.          C*** BUCLE SOBRE CADA LADO CARGADO
68.          C
69.          DO IDIST=1,NEDGE
70.          C
71.          C*** LEE DATOS SOBRE EL LADO CARGADO Y EL VALOR DE LA CARGA
72.          C*** ELEMENTO NUMERO Y NODOS ACTUANTES
73.          C
74.          READ(5,945) NEASS,(NOPRS(ICODEG),ICODEG=1,NODEG)
75.          WRITE(6,950) NEASS,(NOPRS(ICODEG),ICODEG=1,NODEG)
76.          945  FORMAT(I5,<NODEG>I5)
77.          950  FORMAT(I10,5X,<NODEG>I5)
78.          C
79.          C*** POR CADA NODO DEL LADO UN VALOR DE FLUJO Y TEMPERATURA EXT
80.          C
81.          DO IODEG=1,NODEG
82.            READ(5,955) FLUJO,TEMPE
83.            PRESS(ICODEG,1)=TEMPE*PROPS(MATNU(NEASS),NPROP-1)-FLUJO

```

```

84.          WRITE(6,956) FLUJO, TEMPE, PRESS (IODEG, 1)
85.          ENDDO
86.    955    FORMAT(2F10.3)
87.    956    FORMAT(3F10.3)
88.          CALL LADO (PRESS, NEASS, NOPRS, NODEG)
89.          ENDDO
90.          ENDIF
91.          WRITE(6,970)
92.    970    FORMAT(/, 1X, 5X, 'FUERZAS TOTALES NODALES EQUIVALENTES ',
93.          .          'ELEMENTALES', /)
94.          DO IELEM=1, NELEM
95.            WRITE(6,975) IELEM, (CARGA (IELEM, IEVAB), IEVAB=1, NEVAB)
96.          ENDDO
97.    975    FORMAT(1X, I4, 5X, 5E12.4, 11(/, 10X, 5E12.4))
98.          RETURN
99.          END

1.          SUBROUTINE FUERZAS_T0 (ITIME)
2.    C*****
3.    C
4.    C***          CALCULO DE FUERZAS INICIALES
5.    C
6.    C*****
7.          IMPLICIT NONE
8.          INCLUDE 'dtsgral.f'
9.          INCLUDE 'data.f'
10.         INCLUDE 'solu.f'
11.         INTEGER*2 NTOTV, ITOTV, IGD LN, NCONT, IELEM, NROWT,
12.         .         INODE, NODEI, NROWS, JNODE, NODEJ, JGD LN, NCOLS,
13.         .         NCOLE, ICARG, ITIME
14.         REAL*4     ZMASA, BSLOD, ALFAU
15.         DIMENSION ZMASA (60, 60), BSLOD (200)
16.    C*****
17.         NTOTV=NPNOD*NGD LN
18.         ALFAU=1-ALFAT
19.         REWIND 4
20.         REWIND 1
21.         DO ITOTV=1, NTOTV
22.           ASLOD (ITOTV)=0.0
23.         ENDDO
24.    C
25.    C*** ENSAMBLA VECTOR DE FUERZAS ELEMENTALES
26.    C
27.         DO IELEM=1, NELEM
28.           READ(4) ZMASA
29.           READ(1) RIGID
30.           DO INODE=1, NNODE
31.             NODEI=LNODS (IELEM, INODE)
32.             DO JNODE=1, NNODE
33.               NODEJ=LNODS (IELEM, JNODE)
34.               ASLOD (NODEI) =ASLOD (NODEI) + (-RIGID (INODE, JNODE) *ALFAU+
35.               .               ZMASA (INODE, JNODE) ) *XTIME (NODEJ)
36.             ENDDO
37.           ENDDO
38.         ENDDO
39.         READ(4) BSLOD
40.         DO ITOTV=1, NTOTV
41.           ASLOD (ITOTV) =ASLOD (ITOTV) +BSLOD (ITOTV)
42.         ENDDO
43.         IF (ITIME.GT.1) THEN
44.           WRITE(29) NTOTV, (ASLOD (ITOTV), ITOTV=1, NTOTV)
45.         ENDIF

```

```

46.         RETURN
47.         END

1.         SUBROUTINE GAUSSQ (NDIME, NNODE, NGAUS)
2.         C*****
3.         C
4.         C***          CALCULA COORDENADAS Y PESOS DE LA CUADRATURA DE GAUSS
5.         C
6.         C*****data definition
7.         IMPLICIT NONE
8.         INTEGER*2      NDIME, NNODE, NGAUS
9.         INTEGER*2      KGAUS, IGASH, JGASH
10.        INCLUDE 'gaussdat.f'
11.        C*****data ,definiton
12.        C
13.        C***  ELEMENTOS TRIANGULARES
14.        C
15.        IF (NGAUS.EQ.1) THEN
16.        C
17.        C***    CUADRATURA DE 1 PUNTO
18.        C
19.            POSGT(1)=1.0/3.0
20.            POSGT(2)=1.0/3.0
21.            PESGT(1)=1.0/2.0
22.        ENDIF
23.        IF (NGAUS.EQ.3) THEN
24.        C
25.        C***    CUADRATURA DE 3 PUNTOS
26.        C
27.            POSGT(1)=1.0D0/6.0D0
28.            POSGT(2)=2.0D0/3.0D0
29.            POSGT(3)=POSGT(1)
30.            POSGT(4)=POSGT(1)
31.            POSGT(5)=POSGT(1)
32.            POSGT(6)=POSGT(2)
33.            PESGT(1)=1.0D0/6.0D0
34.            PESGT(2)=PESGT(1)
35.            PESGT(3)=PESGT(1)
36.        ENDIF
37.        IF (NGAUS.EQ.4) THEN
38.        C
39.        C***    CUADRATURA DE 4 PUNTOS
40.        C
41.            POSGT(1)=1.0D0/3.0D0
42.            POSGT(2)=0.6D0
43.            POSGT(3)=0.2D0
44.            POSGT(4)=0.2D0
45.            POSGT(5)=POSGT(1)
46.            POSGT(6)=POSGT(3)
47.            POSGT(7)=POSGT(2)
48.            POSGT(8)=POSGT(4)
49.            PESGT(1)=-27.0D0/96.0D0
50.            PESGT(2)= 25.0D0/96.0D0
51.            PESGT(3)=PESGT(2)
52.            PESGT(4)=PESGT(2)
53.        ENDIF
54.        IF (NGAUS.EQ.7) THEN
55.        C
56.        C***    CUADRATURA DE 7 PUNTOS
57.        C
58.            POSGT(1)=0.1012865073235D0
59.            POSGT(2)=0.7974269853531D0

```

```

60.          POSGT(3)=POSGT(1)
61.          POSGT(4)=0.4701420641051D0
62.          POSGT(5)=POSGT(4)
63.          POSGT(6)=0.0597158717898D0
64.          POSGT(7)=1.0D0/3.0D0
65.          POSGT(8)=POSGT(1)
66.          POSGT(9)=POSGT(1)
67.          POSGT(10)=POSGT(2)
68.          POSGT(11)=POSGT(6)
69.          POSGT(12)=POSGT(4)
70.          POSGT(13)=POSGT(4)
71.          POSGT(14)=POSGT(7)
72.          PESGT(1)=0.1259391805448D0/2.0D0
73.          PESGT(2)=PESGT(1)
74.          PESGT(3)=PESGT(1)
75.          PESGT(4)=0.1323941527885D0/2.0D0
76.          PESGT(5)=PESGT(4)
77.          PESGT(6)=PESGT(4)
78.          PESGT(7)=0.225D0/2.0D0
79.          ENDIF
80.          C
81.          C***  ELEMENTOS RECTANGULARES Y CUBICOS
82.          IF (NGAUS.EQ.1) THEN
83.          C
84.          C***  CUADRATURA DE 1 PUNTO  o 1X1 PUNTOS
85.          C
86.          POSPG(1)=0
87.          PESPG(1)=2
88.          ENDIF
89.          IF (NGAUS.EQ.2) THEN
90.          C
91.          C***  CUADRATURA DE 2 PUNTOS  o 2X2 PUNTOS  o 2X2X2 PUNTOS
92.          C
93.          POSPG(1)=-0.577350269189626
94.          PESPG(1)=1.0
95.          ENDIF
96.          IF (NGAUS.EQ.3) THEN
97.          C
98.          C***  CUADRATURA DE 3 PUNTOS  o 3X3 PUNTOS  o 3X3X3 PUNTOS
99.          C
100.         POSPG(1)=-0.774596669241483
101.         POSPG(2)=0.0
102.         PESPG(1)=0.5555555555555556
103.         PESPG(2)=0.8888888888888889
104.         ENDIF
105.         KGAUS=NGAUS/2
106.         DO IGASH=1,KGAUS
107.             JGASH=NGAUS+1-IGASH
108.             POSPG(JGASH)=-POSPG(IGASH)
109.             PESPG(JGASH)=PESPG(IGASH)
110.         ENDDO
111.         RETURN
112.         END

1.          SUBROUTINE INVER
2.          C*****
3.          C          SUBROUTINA PARA INVERSION DE MATRICES POR ELIMINACION
4.          C          GAUSIANA
5.          C          PARAMETROS:
6.          C
7.          C          A ====>  MATRIZ A INVERTIR
8.          C          B ====>  MATRIZ INVERTIDA

```



```

9.      C          N =====>  ORDEN DE LA MATRIZ
10.     C          IERROR      INDICADOR DE ERROR
11.     C*****
12.     IMPLICIT NONE
13.     INCLUDE 'solucuas.f'
14.     INTEGER*2      I, J, K
15.     REAL*4         TOLER, P
16.     DATA TOLER /1.0E-32/
17.     DO 10 I=1,NTNOD
18.         IF(XMASA(I,I).LT.TOLER) THEN
19.             DO 20 J=1,NTNOD
20.                 XMASA(I,J)=0.0
21.             20      IF(J.EQ.I) XMASA(I,I)=1.0
22.         ENDIF
23.     10  CONTINUE
24.     IERROR=0
25.     DO 30 I=1,NTNOD
26.         DO 30 J=1,NTNOD
27.             XINV(I,J)=0.0
28.             IF(I.EQ.J) THEN
29.                 XINV(I,J)=1.
30.             ENDIF
31.     30  CONTINUE
32.     DO 60 I=1,NTNOD
33.         P=XMASA(I,I)
34.         IF(ABS(P).LT.TOLER) THEN
35.             WRITE(6,900)
36.             IERROR=1
37.             RETURN
38.         ENDIF
39.         DO 50 J=1,NTNOD
40.             XINV(I,J)=XINV(I,J)/P
41.     50      XMASA(I,J)=XMASA(I,J)/P
42.         DO 60 K=1,NTNOD
43.             IF(K.NE.I) THEN
44.                 P=XMASA(K,I)
45.                 DO 70 J=1,NTNOD
46.                     XMASA(K,J)=XMASA(K,J)-P*XMASA(I,J)
47.     70      XINV(K,J)=XINV(K,J)-P*XINV(I,J)
48.             ENDIF
49.         60  CONTINUE
50.         RETURN
51.     900  FORMAT('ERROR EN LA RUTINA DE INVERSION DE MATRICES')
52.     END

```

```

1.      SUBROUTINE JACOBM(IELEM,DJACB,KPGAU)
2.     C*****
3.     C
4.     C***      CALCULA EL JACOBIANO, SU DETERMINANTE E INVERSA,
5.     C          Y DERIVADAS CARTESIANAS DE LAS FUNCIONES DE FORMA
6.     C
7.     C*****
8.     IMPLICIT NONE
9.     INCLUDE 'dtsgral.f'
10.    INCLUDE 'gaussdat.f'
11.    INCLUDE 'calculo.f'
12.    INTEGER*2      KPGAU, IDIME, JDIME, INODE, IELEM
13.    REAL*4         XJACM, XJACI, DJACB
14.    DIMENSION      XJACM(3,3), XJACI(3,3)
15.    C*****
16.    C
17.    C***      CALCULA COORDENADAS DE LOS PUNTOS DE INTEGRACION

```

```

18. C
19. IF (KPGAU.NE.0) THEN
20.   DO IDIME=1,NDIME
21.     CORPG (IDIME,KPGAU)=0.0
22.     DO INODE=1,NNODE
23.       CORPG (IDIME,KPGAU)=CORPG (IDIME,KPGAU)+COREL (IDIME, INODE)
24.       *FFORM (INODE)
25.     ENDDO
26.   ENDDO
27. ENDIF
28. IF (NDIME.EQ.1) THEN
29. C
30. C*** CALCULA EL JACOBIANO UNIDIMENSIONAL Y SU DETERMINANTE
31. C
32.   XJACM(1,1)=0.0
33.   XJACM(1,2)=0.0
34.   DO IDIME=1,2
35.     DO INODE=1,NNODE
36.       XJACM(1, IDIME)=XJACM(1, IDIME)+
37.       .   DERIV(1, INODE)*COREL (IDIME, INODE)
38.     ENDDO
39.   ENDDO
40.   DJACB=SQRT(1+XJACM(1,2)*XJACM(1,2)/XJACM(1,1)/XJACM(1,1))
41.   *XJACM(1,1)
42.   IF (DJACB.LE.0.0) THEN
43.     WRITE (6,900) IELEM
44.     STOP
45.   ENDIF
46.   XJACI(1,1)=1/XJACM(1,1)
47. ELSE
48. C
49. C*** CALCULA MATRIZ JACOBIANO 1D 2D Y 3D
50. C
51.   DO IDIME=1,NDIME
52.     DO JDIME=1,NDIME
53.       XJACM (IDIME,JDIME)=0.0
54.       DO INODE=1,NNODE
55.         XJACM (IDIME,JDIME)=XJACM (IDIME,JDIME)+
56.         .   DERIV (IDIME, INODE)*COREL (JDIME, INODE)
57.       ENDDO
58.     ENDDO
59.   ENDDO
60. ENDIF
61. IF (NDIME.EQ.2) THEN
62. C
63. C*** CALCULA EL DETERMINANTE Y LA INVERSA DEL JACOBIANO EN 2D
64. C
65.   DJACB=XJACM(1,1)*XJACM(2,2)-XJACM(1,2)*XJACM(2,1)
66.   IF (DJACB.LE.0.0) THEN
67.     WRITE (6,900) IELEM
68.     STOP
69.   ENDIF
70.   XJACI(1,1)=XJACM(2,2)/DJACB
71.   XJACI(2,2)=XJACM(1,1)/DJACB
72.   XJACI(1,2)=-XJACM(1,2)/DJACB
73.   XJACI(2,1)=-XJACM(2,1)/DJACB
74. ENDIF
75. IF (NDIME.EQ.3) THEN
76. C
77. C*** CALCULA EL DETERMINANTE Y LA INVERSA DEL JACOBIANO EN 3D
78. C
79.   DJACB=XJACM(1,1)*XJACM(2,2)*XJACM(3,3)+XJACM(1,3)*XJACM
80.   .   (2,1)*XJACM(3,2)+XJACM(3,1)*XJACM(1,2)*XJACM(2,3)-

```

```

81.      .      XJACM(3,1)*XJACM(2,2)*XJACM(1,3)-XJACM(3,3)*XJACM
82.      .      (1,2)*XJACM(2,1)-XJACM(1,1)*XJACM(2,3)*XJACM(3,2)
83.      IF(DJACB.LE.0.0) THEN
84.      WRITE (6,900) IELEM
85.      900    FORMAT(/, 'PROGRAMA DETENIDO EN SUBROUTINA JACOBM',/,11X,
86.      .      ' DETERMINANTE DEL JACOBIANO CERO O NEGATIVO',/,10X,
87.      .      ' EN ELEMENTO NUMERO',I5)
88.      STOP
89.      ENDIF
90.      XJACI(1,1)= (XJACM(2,2)*XJACM(3,3)-XJACM(3,2)*XJACM(2,3))/DJACB
91.      XJACI(1,2)=- (XJACM(1,2)*XJACM(3,3)-XJACM(1,3)*XJACM(3,2))/DJACB
92.      XJACI(1,3)= (XJACM(1,2)*XJACM(2,3)-XJACM(2,2)*XJACM(1,3))/DJACB
93.      XJACI(2,1)=- (XJACM(2,1)*XJACM(3,3)-XJACM(3,1)*XJACM(2,3))/DJACB
94.      XJACI(2,2)= (XJACM(1,1)*XJACM(3,3)-XJACM(1,3)*XJACM(3,1))/DJACB
95.      XJACI(2,3)=- (XJACM(1,1)*XJACM(2,3)-XJACM(2,1)*XJACM(1,3))/DJACB
96.      XJACI(3,1)= (XJACM(2,1)*XJACM(3,2)-XJACM(3,1)*XJACM(2,2))/DJACB
97.      XJACI(3,2)=- (XJACM(1,1)*XJACM(3,2)-XJACM(3,1)*XJACM(1,2))/DJACB
98.      XJACI(3,3)= (XJACM(1,1)*XJACM(2,2)-XJACM(1,2)*XJACM(2,1))/DJACB
99.      ENDIF
100.     C
101.     C***  CALCULA LAS DERIVADAS CARTESIANAS DE LAS FUNCIONES DE FORMA
102.     C
103.     DO IDIME=1,NDIME
104.     DO INODE=1,NNODE
105.     DCART(IDIME,INODE)=0.0
106.     DO JDIME=1,NDIME
107.     DCART(IDIME,INODE)=DCART(IDIME,INODE)+XJACI(IDIME,JDIME)*
108.     .      DERIV(JDIME,INODE)
109.     ENDDO
110.     ENDDO
111.     ENDDO
112.     RETURN
113.     END

1.      SUBROUTINE LADO(PRESS,NEASS,NOPRS,NODEG)
2.      C*****
3.      C
4.      C***  CALCULO DE FUERZAS NODALES EQUIVALENTES
5.      C      SOBRE UN LADO DEL ELEMENTO
6.      C*****
7.      IMPLICIT NONE
8.      INCLUDE 'dtsgral.f'
9.      INCLUDE 'data.f'
10.     INCLUDE 'gaussdat.f'
11.     INCLUDE 'calculo.f'
12.     INTEGER*2      IODEG,NODEG,NEASS,NOPRS,LNODE, IDIME, IGAUS,
13.     .      IGDLN, JNODE, INODE, NLOCA, KOUNT, KNODE, NCONT,
14.     .      UNOVA, KPGAU, LPROP, JGAUS, JDIME
15.     REAL*4      PRESS, EXISP, CORD, PGASH, DGASH, DAREA, PXCOM,
16.     .      D, DLONG, ETASP, ZJACC
17.     DIMENSION  PRESS(20,2),NOPRS(9),D(2),ZJACC(3,3),
18.     .      PXCOM(20)
19.     C*****
20.     UNOVA=1
21.     DO IODEG=1,NODEG
22.     LNODE=NOPRS(IODEG)
23.     DO IDIME=1,NDIME
24.     COREL(IDIME,IODEG)=COORD(LNODE, IDIME)
25.     ENDDO
26.     ENDDO
27.     IF (NDIME.EQ.1) THEN
28.     C

```

```

29.  C*** ASOCIA TEMPERATURA EXTERNA Y EL FLUJO ACTUANTE A UN NODO EN
30.  C*** FORMA DIRECTA
31.  C
32.      JNODE=0
33.      DO INODE=1,NNODE
34.          NLOCA=LNODS (NEASS, INODE)
35.          IF (NLOCA.EQ.NOPRS (1)) JNODE=INODE
36.      ENDDO
37.      CARGA (NEASS, JNODE) =CARGA (NEASS, JNODE) +PRESS (1, 1)
38.  ENDIF
39.  IF (NDIME.EQ.2) THEN
40.  C
41.  C*** BUCLE DE INTEGRACION NUMERICA
42.  C
43.      DO IGAUS=1,NGAUS
44.          EXISP=POSPG (IGAUS)
45.  C
46.  C*** CALCULA FUNCIONES DE FORMA EN PUNTOS DE INTEGRACION
47.  C
48.          CALL FFORMA (EXISP, 0, 0, UNOVA, NODEG, FFORM, DERIV)
49.  C
50.  C*** CALCULA EL JACOBIANO
51.  C
52.      D(1)=0.0
53.      D(2)=0.0
54.      DO IODEG=1, NODEG
55.          DO IDIME=1, NDIME
56.              D (IDIME) =D (IDIME) +
57.                  DERIV (1, IODEG) *COREL (IDIME, IODEG)
58.          ENDDO
59.      ENDDO
60.      DLONG=SQRT (1+D (2) *D (2) /D (1) /D (1)) *D (1) *PESPG (IGAUS)
61.  C
62.  C*** CALCULA LAS FUERZAS NODALES EQUIVALENTES
63.  C
64.      DO IODEG=1, NODEG
65.          PXCOM (IODEG) = PRESS (IODEG, 1) *FFORM (IODEG) *DLONG
66.      ENDDO
67.  C
68.  C*** ASOCIA LAS FUERZAS NODALES EQUIVALENTES DE BORDE A UN ELEMENTO
69.  C
70.      JNODE=0
71.      DO INODE=1,NNODE
72.          NLOCA=LNODS (NEASS, INODE)
73.          IF (NLOCA.EQ.NOPRS (1)) JNODE=INODE
74.      ENDDO
75.      INODE=JNODE
76.      JNODE=JNODE+NODEG-1
77.      KOUNT=0
78.      DO KNODE=INODE, JNODE
79.          KOUNT=KOUNT+1
80.          NCONT= (KNODE-1) *NGDLN+1
81.          IF (KNODE.GT.NNODE .OR. (NNODE.EQ.9 .AND. KNODE.EQ.9)) NCONT=1
82.          CARGA (NEASS, NCONT) =CARGA (NEASS, NCONT) +PXCOM (KNODE-INODE+1)
83.      ENDDO
84.  ENDDO
85.  ENDIF
86.  IF (NDIME.EQ.3) THEN
87.      UNOVA=2
88.      KPGAU=0
89.      DO IGAUS=1,NGAUS
90.          DO JGAUS=1,NGAUS
91.              KPGAU=KPGAU+1

```



```

92.          EXISP=POSPG(IGAUS)
93.          ETASP=POSPG(JGAUS)
94.      C
95.      C*** CALCULA FUNCIONES DE FORMA EN PUNTOS DE INTEGRACION
96.      C
97.          CALL FFORMA(EXISP,ETASP,0,UNOVA,NNFRO,FFORM,DERIV)
98.      C
99.      C*** CALCULA EL JACOBIANO
100.     C
101.         DO IDIME=1,3
102.             DO JDIME=1,3
103.                 ZJACC(IDIME,JDIME)=0.0
104.                 DO IODEG=1,NODEG
105.                     ZJACC(IDIME,JDIME)=ZJACC(IDIME,JDIME)+
106.                         DERIV(IDIME,IODEG)*COREL(JDIME,IODEG)
107.                 ENDDO
108.             ENDDO
109.         ENDDO
110.         DAREA=SQRT(
111.             (ZJACC(1,2)*ZJACC(2,3)-ZJACC(1,3)*ZJACC(2,2))**2+
112.             (ZJACC(1,3)*ZJACC(2,1)-ZJACC(1,1)*ZJACC(2,3))**2+
113.             (ZJACC(1,1)*ZJACC(2,2)-ZJACC(1,2)*ZJACC(2,1))**2)
114.     C
115.     C*** CALCULA LAS FUERZAS NODALES EQUIVALENTES
116.     C
117.         DO IODEG=1,NODEG
118.             PXCOM(IODEG)=PRESS(IODEG,1)*FFORM(IODEG)*DAREA
119.         ENDDO
120.     C
121.     C*** ASOCIA LAS FUERZAS NODALES EQUIVALENTES DE BORDE A UN ELEMENTO
122.     C
123.         JNODE=0
124.         DO INODE=1,NNODE
125.             NLOCA=LNODS(NEASS,INODE)
126.             IF(NLOCA.EQ.NOPRS(1)) JNODE=INODE
127.         ENDDO
128.         INODE=JNODE
129.         JNODE=JNODE+NODEG-1
130.         KOUNT=0
131.         DO KNODE=INODE,JNODE
132.             KOUNT=KOUNT+1
133.             NCONT=(KNODE-1)*NGDLN+1
134.             IF(KNODE.GT.NNODE.OR.(NNODE.EQ.9.AND.KNODE.EQ.9)) NCONT=1
135.             CARGA(NEASS,NCONT)=CARGA(NEASS,NCONT)+PXCOM(KNODE-INODE+1)
136.         ENDDO
137.     ENDDO
138. ENDDO
139. ENDIF
140. RETURN
141. END

1.      SUBROUTINE PESOP(PESPP,IELEM)
2.      C*****
3.      C
4.      C***      CALCULO DE FUERZAS NODALES EQUIVALENTES
5.      C      POR GENERACION INTERNA DE CALOR (EQ. A PESO PROPIO)
6.      C*****
7.          IMPLICIT NONE
8.          INCLUDE 'dtsgral.f'
9.          INCLUDE 'data.f'
10.         INCLUDE 'gaussdat.f'
11.         INCLUDE 'calcula.f'

```

```

12.      INTEGER*2          INODE, LNODE, IDIME, IELEM, IGAUS, JGAUS, KPGAU,
13.      .                  KGAUS, MCONT
14.      REAL*4             EXISP, ETASP, EGASP, DVOLU, PESPP, DJACB, EGISP
15.      C*****
16.      C
17.      C*** CARGAS POR GENERACION INTERNA DE CALOR
18.      C
19.      C*** CALCULA COORDENADAS DE LOS PUNTOS NODALES
20.      C
21.      DO INODE=1, NNODE
22.          LNODE=LNODS( IELEM, INODE)
23.          DO IDIME=1, NDIME
24.              COREL( IDIME, INODE) =COORD( LNODE, IDIME)
25.          ENDDO
26.      ENDDO
27.      IF (NDIME.EQ.1) THEN
28.          DO IGAUS=1, NGAUS
29.              EXISP=POSPG( IGAUS)
30.      C
31.      C*** CALCULA FUNCIONES DE FORMA EN LOS PUNTOS DE INTEGRACION Y
32.      C EL VOLUMEN ELEMENTAL
33.      C
34.          CALL FFORMA( EXISP, 0, 0, NDIME, NNODE, FFORM, DERIV)
35.          KPGAU=1
36.          CALL JACOBM( IELEM, DJACB, KPGAU)
37.          DVOLU=DJACB*PESPG( IGAUS)
38.      C
39.      C*** CALCULA LAS CARGAS NODALES EQUIVALENTES
40.      C
41.          DO INODE=1, NNODE
42.              MCONT=INODE
43.              CARGA( IELEM, MCONT) =CARGA( IELEM, MCONT) +
44.              PESPP*FFORM( INODE) *DVOLU
45.          ENDDO
46.      ENDDO
47.      ENDIF
48.      IF (NDIME.EQ.2) THEN
49.          IF (NNODE.EQ.4.OR.NNODE.EQ.8.OR.NNODE.EQ.9) THEN
50.      C
51.      C*** BUCLE DE INTEGRACION NUMERICA PARA ELEMENTOS CUADRILATEROS
52.      C
53.          DO IGAUS=1, NGAUS
54.              DO JGAUS=1, NGAUS
55.                  EXISP=POSPG( IGAUS)
56.                  ETASP=POSPG( JGAUS)
57.      C
58.      C*** CALCULA FUNCIONES DE FORMA EN LOS PUNTOS DE INTEGRACION Y
59.      C EL VOLUMEN ELEMENTAL
60.      C
61.          CALL FFORMA( EXISP, ETASP, EGASP, NDIME, NNODE, FFORM, DERIV)
62.          KPGAU=1
63.          CALL JACOBM( IELEM, DJACB, KPGAU)
64.          DVOLU=DJACB*PESPG( IGAUS) *PESPG( JGAUS)
65.      C
66.      C*** CALCULA LAS CARGAS NODALES EQUIV PARA CADA ELEMENTO CUADRILATERO
67.      C
68.          DO INODE=1, NNODE
69.              MCONT=INODE
70.              CARGA( IELEM, MCONT) =CARGA( IELEM, MCONT) +
71.              PESPP*FFORM( INODE) *DVOLU
72.          ENDDO
73.      ENDDO
74.      ENDDO

```

```

75.         ENDIF
76.         IF (NNODE.EQ.3.OR.NNODE.EQ.6) THEN
77.     C
78.     C***     BUCLE DE INTEGRACION NUMERICA PARA ELEMENTOS TRIANGULARES
79.     C
80.         DO IGAUS=1,NGAUS
81.             EXISP=POSGT(IGAUS)
82.             ETASP=POSGT(NGAUS+IGAUS)
83.     C
84.     C***     CALCULA FUNCIONES DE FORMA EN LOS PUNTOS DE INTEGRACION Y
85.     C     EL VOLUMEN ELEMENTAL
86.     C
87.             CALL FFORMA(EXISP,ETASP,EGASP,NDIME,NNODE,FFORM,DERIV)
88.             KPGAU=1
89.             CALL JACOBM(IELEM,DJACB,KPGAU)
90.             DVOLU=DJACB*PESGT(IGAUS)
91.     C
92.     C***     CALCULA LAS CARGAS NODALES EQUIV PARA CADA ELEMENTO TRIANGULAR
93.     C
94.             DO INODE=1,NNODE
95.                 MCONT=INODE
96.                 CARGA(IELEM,MCONT)=CARGA(IELEM,MCONT)+
97.                     PESPP*FFORM(INODE)*DVOLU
98.             ENDDO
99.         ENDDO
100.        ENDIF
101.    ENDIF
102.    C
103.    C***     BUCLE DE INTEGRACION NUMERICA PARA ELEMENTOS DE SOLIDO
104.    C
105.        IF(NDIME.EQ.3) THEN
106.            KPGAU=0
107.            DO IGAUS=1,NGAUS
108.                DO JGAUS=1,NGAUS
109.                    DO KGAUS=1,NGAUS
110.                        KPGAU=KPGAU+1
111.                        EXISP=POSPG(IGAUS)
112.                        ETASP=POSPG(JGAUS)
113.                        EGISP=POSPG(KGAUS)
114.                    C
115.                    C***     CALCULA FUNCIONES DE FFORM EN LOS PUNTOS DE INTEGRACION Y
116.                    C     EL VOLUMEN ELEMENTAL
117.                    C
118.                        CALL FFORMA(EXISP,ETASP,EGISP,NDIME,NNODE,FFORM,DERIV)
119.                        CALL JACOBM(IELEM,DJACB,KPGAU)
120.                        DVOLU=DJACB*PESPG(IGAUS)*PESPG(JGAUS)*PESPG(KGAUS)
121.                    C
122.                    C***     CALCULA LAS CARGAS EQUIV NODALES PARA CADA ELEMENTO CUADRATICO
123.                    C
124.                        DO INODE=1,NNODE
125.                            MCONT=(INODE-1)*NGDLN+1
126.                            CARGA(IELEM,MCONT)=CARGA(IELEM,MCONT)+
127.                                PESPP*FFORM(INODE)*DVOLU
128.                        ENDDO
129.                    ENDDO
130.                ENDDO
131.            ENDDO
132.        ENDIF
133.    RETURN
134.    END

```

```

1.      SUBROUTINE PUNTUAL
2.      C*****
3.      C
4.      C***      CALCULO DE FUERZAS NODALES EQUIVALENTES PUNTUALES
5.      C
6.      C*****
7.      IMPLICIT NONE
8.      INTEGER*2      LODPT, IELEM, JELEM, INODE, JNODE, NLOCA, IGDLN,
9.      .              NCONT
10.     REAL*4          PNODT
11.     DIMENSION      PNODT(3)
12.     INCLUDE 'dtsgral.f'
13.     INCLUDE 'data.f'
14.     INCLUDE 'calculo.f'
15.     C*****
16.     C
17.     C*** LEE CARGAS PUNTUALES
18.     C
19.     20 READ(5,900) LODPT, (PNODT(IGDLN), IGDLN=1, NGDLN)
20.     WRITE(6,900) LODPT, (PNODT(IGDLN), IGDLN=1, NGDLN)
21.     900 FORMAT(I5,5F10.3)
22.     C
23.     C*** ASOCIA LAS CARGAS PUNTUALES NODALES CON UN ELEMENTO
24.     C
25.     DO IELEM=1,NELEM
26.     DO INODE=1,NNODE
27.     NLOCA=LNODS(IELEM,INODE)
28.     IF(LODPT.EQ.NLOCA) THEN
29.     JNODE=INODE
30.     JELEM=IELEM
31.     ENDIF
32.     ENDDO
33.     ENDDO
34.     DO IGDLN=1,NGDLN
35.     NCONT=(JNODE-1)*NGDLN+IGDLN
36.     CARGA(JELEM,NCONT)=PNODT(IGDLN)
37.     ENDDO
38.     IF(LODPT.LT.NPNOD) GO TO 20
39.     RETURN
40.     END

```

```

1.      SUBROUTINE REDUCE
2.      C*****
3.      C
4.      C***      REDUCE EL SISTEMA DE ECUACIONES GLOBALES POR ELIMINACION
5.      C              GAUSSIANA DIRECTA
6.      C
7.      C*****
8.      IMPLICIT NONE
9.      INCLUDE 'solu.f'
10.     INTEGER*2      NEONS, IEONS, IROWS, IEON1, ICOLS
11.     REAL*4          B, PIVOT, FACTR
12.     C*****
13.     NEONS=NSVAB
14.     DO 50 IEONS=1,NEONS
15.     IF(IFFIX(IEONS).EQ.1) THEN
16.     C
17.     C*** AJUSTA EL VECTOR DE CARGAS PARA DESPLAZAMIENTOS PRESCRITOS
18.     C
19.     DO IROWS=IEONS,NEONS
20.     B=ASTIF(IROWS,IEONS)*FIXED(IEONS)
21.     ASLOD(IROWS)=ASLOD(IROWS)-B

```



```

22.          ENDDO
23.          ELSE
24.      C
25.      C***  REDUCE ECUACIONES
26.      C
27.          PIVOT=ASTIF(IEONS, IEONS)
28.          IF(ABS(PIVOT).LT.1.0E-10) THEN
29.              WRITE(6,900) PIVOT, IEONS
30.              STOP
31.          ENDIF
32.          IF(IEONS.EQ.NEONS) GOTO 50
33.          IEON1=IEONS+1
34.          DO IROWS=IEON1, NEONS
35.              FACTR=ASTIF(IROWS, IEONS)/PIVOT
36.              IF(FACTR.NE.0.0) THEN
37.                  DO ICOLS=IEONS, NEONS
38.                      ASTIF(IROWS, ICOLS)=ASTIF(IROWS, ICOLS) -
39.                          FACTR*ASTIF(IEONS, ICOLS)
40.                  ENDDO
41.                  ASLOD(IROWS)=ASLOD(IROWS) - FACTR*ASLOD(IEONS)
42.                  ASTIF(IROWS, IEONS)=FACTR
43.              ENDIF
44.          ENDDO
45.      ENDIF
46.      50 CONTINUE
47.      900 FORMAT(5X, 'PIVOTE INCORRECTO=', E20.6, 5X, 'ECUACION No.=', I5)
48.      RETURN
49.      END

1.          SUBROUTINE REDUCE1
2.      C*****
3.      C
4.      C***      REDUCE EL SISTEMA DE ECUACIONES GLOBALES POR ELIMINACION
5.                  GAUSSIANA DIRECTA
6.      C
7.      C*****
8.          IMPLICIT NONE
9.          INCLUDE 'solu.f'
10.         INTEGER*2          NEONS, IEONS, IROWS, IEON1, ICOLS
11.         REAL*4             B, PIVOT, FACTR
12.      C*****
13.         NEONS=NSVAB
14.         DO 50 IEONS=1, NEONS
15.             IF(IFFIX(IEONS).EQ.1) THEN
16.                 C
17.                 C***  AJUSTA EL VECTOR DE CARGAS PARA DESPLAZAMIENTOS PRESCRITOS
18.                 C
19.                     DO IROWS=IEONS, NEONS
20.                         B=ASTIF(IROWS, IEONS)*FIXED(IEONS)
21.                         ASLOD(IROWS)=ASLOD(IROWS) - B
22.                     ENDDO
23.                 ELSE
24.                     C
25.                     C***  REDUCE ECUACIONES
26.                     C
27.                         IF(IEONS.EQ.NEONS) GOTO 50
28.                         IEON1=IEONS+1
29.                         DO IROWS=IEON1, NEONS
30.                             FACTR=ASTIF(IROWS, IEONS)
31.                             IF(FACTR.NE.0.0) THEN
32.                                 ASLOD(IROWS)=ASLOD(IROWS) - FACTR*ASLOD(IEONS)
33.                             ENDIF

```

```

34.          ENDDO
35.          ENDIF
36.    50 CONTINUE
37.          RETURN
38.          END

1.          SUBROUTINE RIGIMAT
2.          C*****
3.          C
4.          C***          CALCULA MATRIZ DE RIGIDEZ DE CADA ELEMENTO
5.          C
6.          C*****
7.          IMPLICIT NONE
8.          INTEGER*2          IELEM, INODE, IDIME, IEVAB, JEVAB, KPGAU, IGAUS,
9.          .                  JGAUS, ITENS, KGAUS, LPROP, LNODE
10.         REAL*4          EGASP, RIGID, EXISP, ETASP, DJACB, DVOLU,
11.         .                  FTFOR, ZMASA, ROCEE, PATOP, XLONG
12.         INCLUDE 'dtsgral.f'
13.         INCLUDE 'data.f'
14.         INCLUDE 'calculo.f'
15.         INCLUDE 'gaussdat.f'
16.         DIMENSION RIGID(60,60), FTFOR(60,60), ZMASA(60,60)
17.         C*****
18.         IF (NTIPO.NE.1) THEN
19.             WRITE(6,900)
20.             900  FORMAT(1X,'ERROR EN LA SELECCION DE TIPO DE ANALISIS')
21.             STOP
22.         ENDIF
23.         EGASP=0.0
24.         XLUMP=0.0
25.         C
26.         C*** BUCLE SOBRE CADA ELEMENTO
27.         C
28.         DO IELEM=1,NELEM
29.             LPROP=ABS(MATNU(IELEM))
30.         C
31.         C***          CALCULA LAS COORDENADAS DE LOS PUNTOS NODALES DEL ELEMENTO
32.         C
33.         DO INODE=1,NNODE
34.             LNODE=LNODS(IELEM,INODE)
35.             DO IDIME=1,NDIME
36.                 COREL(IDIME,INODE)=COORD(LNODE, IDIME)
37.             ENDDO
38.         ENDDO
39.         C
40.         C*** CALCULA LA MATRIZ D
41.         C
42.             CALL DMAT(LPROP)
43.         C
44.         C*** INICIALIZA LA MATRIZ DE RIGIDEZ ELEMENTAL
45.         C
46.             DO IEVAB=1,NEVAB
47.                 DO JEVAB=1,NEVAB
48.                     RIGID(IEVAB,JEVAB)=0.0
49.                 ENDDO
50.             ENDDO
51.         C
52.         C*** INICIALIZA LA MATRIZ DE MASA ELEMENTAL
53.         C
54.             IF (NTRAN.GT.0) THEN
55.                 DO IEVAB=1,NEVAB
56.                     DO JEVAB=1,NEVAB

```

```

57.             ZMASA (IEVAB, JEVAB) = 0.0
58.             ENDDO
59.             ENDDO
60.             ROCEE = PROPS (LPROP, NPROP) / DTIME
61.         ENDIF
62.     C
63.     C*** AQUI EMPIEZA
64.         KPGAU = 0
65.         IF (NDIME.EQ.1) THEN
66.             DO IGAUS = 1, NGAUS
67.                 KPGAU = KPGAU + 1
68.                 EXISP = POSPG (IGAUS)
69.                 CALL FFORMA (EXISP, 0, 0, NDIME, NNODE, FFORM, DERIV)
70.                 CALL JACOBM (IELEM, DJACB, KPGAU)
71.                 DVOLU = DJACB * PESPG (IGAUS)
72.                 CALL BMAT (IELEM, KPGAU)
73.                 CALL DBMATX
74.                 DO IEVAB = 1, NEVAB
75.                     DO JEVAB = IEVAB, NEVAB
76.                         DO ITENS = 1, NTENS
77.                             RIGID (IEVAB, JEVAB) = RIGID (IEVAB, JEVAB) +
78.                             BMATZ (ITENS, IEVAB) * DBMAT (ITENS, JEVAB) * DVOLU
79.                         ENDDO
80.                         IF (NTRAN.GT.0) THEN
81.                             PATOP = FFORM (IEVAB) * FFORM (JEVAB) * DVOLU * ROCEE
82.                             ZMASA (IEVAB, JEVAB) = ZMASA (IEVAB, JEVAB) + PATOP
83.                             IF (NTRAN.EQ.3) XLUMP = XLUMP + DVOLU * ROCEE
84.                         ENDIF
85.                     ENDDO
86.                 ENDDO
87.                 DO ITENS = 1, NTENS
88.                     DO IEVAB = 1, NEVAB
89.                         TENSZ (ITENS, IEVAB, KPGAU) = DBMAT (ITENS, IEVAB)
90.                     ENDDO
91.                 ENDDO
92.             ENDIF
93.         IF (NDIME.EQ.2) THEN
94.             IF ( (NNODE.EQ.4) .OR. (NNODE.EQ.8) .OR. (NNODE.EQ.9) ) THEN
95.                 C
96.                 C*** ELEMENTOS CUADRILATEROS
97.                 C
98.                     DO IGAUS = 1, NGAUS
99.                         DO JGAUS = 1, NGAUS
100.                            KPGAU = KPGAU + 1
101.                            EXISP = POSPG (IGAUS)
102.                            ETASP = POSPG (JGAUS)
103.                            CALL FFORMA (EXISP, ETASP, EGASP, NDIME, NNODE, FFORM, DERIV)
104.                            CALL JACOBM (IELEM, DJACB, KPGAU)
105.                            DVOLU = DJACB * PESPG (IGAUS) * PESPG (JGAUS)
106.                            CALL BMAT (IELEM, KPGAU)
107.                            CALL DBMATX
108.                            DO IEVAB = 1, NEVAB
109.                                DO JEVAB = IEVAB, NEVAB
110.                                    DO ITENS = 1, NTENS
111.                                        RIGID (IEVAB, JEVAB) = RIGID (IEVAB, JEVAB) +
112.                                        BMATZ (ITENS, IEVAB) * DBMAT (ITENS, JEVAB) * DVOLU
113.                                    ENDDO
114.                                    IF (NTRAN.GT.0) THEN
115.                                        PATOP = FFORM (IEVAB) * FFORM (JEVAB) * DVOLU * ROCEE
116.                                        ZMASA (IEVAB, JEVAB) = ZMASA (IEVAB, JEVAB) + PATOP
117.                                        IF (NTRAN.EQ.3) XLUMP = XLUMP + DVOLU * ROCEE
118.                                    ENDIF
119.                                ENDDO

```

```

120.          ENDDO
121.          ENDDO
122.          DO ITENS=1,NTENS
123.              DO IEVAB=1,NEVAB
124.                  TENSZ (ITENS,IEVAB,KPGAU)=DBMAT (ITENS,IEVAB)
125.              ENDDO
126.          ENDDO
127.          ENDDO
128.          ENDDO
129.          ENDIF
130.          IF ((NNODE.EQ.3).OR.(NNODE.EQ.6)) THEN
131.      C
132.      C*** ELEMENTOS TRIANGULARES
133.      C
134.          DO IGAUS=1,NGAUS
135.              KPGAU=KPGAU+1
136.              EXISP=POSGT(IGAUS)
137.              ETASP=POSGT(NGAUS+IGAUS)
138.              CALL FFORMA(EXISP,ETASP,EGASP,NDIME,NNODE,FFORM,DERIV)
139.              CALL JACOBM(IELEM,DJACB,KPGAU)
140.              DVOLU=DJACB*PESGT(IGAUS)
141.              CALL BMAT(IELEM,KPGAU)
142.              CALL DBMATX
143.              DO IEVAB=1,NEVAB
144.                  DO JEVAB=IEVAB,NEVAB
145.                      DO ITENS=1,NTENS
146.                          RIGID(IEVAB,JEVAB)=RIGID(IEVAB,JEVAB)+
147.                              BMATZ(ITENS,IEVAB)*DBMAT(ITENS,JEVAB)*DVOLU
148.                      ENDDO
149.                      IF (NTRAN.GT.0) THEN
150.                          PATOP=FFORM(IEVAB)*FFORM(JEVAB)*DVOLU*ROCEE
151.                          ZMASA(IEVAB,JEVAB)=ZMASA(IEVAB,JEVAB)+PATOP
152.                      IF (NTRAN.EQ.3) XLUMP=XLUMP+DVOLU*ROCEE
153.                      ENDIF
154.                  ENDDO
155.              ENDDO
156.          DO ITENS=1,NTENS
157.              DO IEVAB=1,NEVAB
158.                  TENSZ (ITENS,IEVAB,KPGAU)=DBMAT (ITENS,IEVAB)
159.              ENDDO
160.          ENDDO
161.          ENDDO
162.          ENDIF
163.          ENDIF
164.      C
165.      C*** BUCLE PARA INTEGRACION NUMERICA EN ELEMENTOS SOLIDOS
166.      C      TRIDIMENSIONALES HEXAGONALES
167.      C
168.          IF (NDIME.EQ.3) THEN
169.              DO IGAUS=1,NGAUS
170.                  DO JGAUS=1,NGAUS
171.                      DO KGAUS=1,NGAUS
172.                          KPGAU=KPGAU+1
173.                          EXISP=POSPG(IGAUS)
174.                          ETASP=POSPG(JGAUS)
175.                          EGASP=POSPG(KGAUS)
176.                          CALL FFORMA(EXISP,ETASP,EGASP,NDIME,NNODE,FFORM,DERIV)
177.                          CALL JACOBM(IELEM,DJACB,KPGAU)
178.                          DVOLU=DJACB*PESPG(IGAUS)*PESPG(JGAUS)*PESPG(KGAUS)
179.                          CALL BMAT(IELEM,KPGAU)
180.                          CALL DBMATX
181.                          DO IEVAB=1,NEVAB
182.                              DO JEVAB=IEVAB,NEVAB

```



```

183.          DO ITENS=1, NTENS
184.             RIGID(IEVAB, JEVAB)=RIGID(IEVAB, JEVAB)+
185.             BMATZ(ITENS, IEVAB)*DBMAT(ITENS, JEVAB)*DVOLU
186.          ENDDO
187.          IF (NTRAN.GT.0) THEN
188.             PATOP=FFORM(IEVAB)*FFORM(JEVAB)*DVOLU*ROCEE
189.             ZMASA(IEVAB, JEVAB)=ZMASA(IEVAB, JEVAB)+PATOP
190.             IF (NTRAN.EQ.3) XLUMP=XLUMP+DVOLU*ROCEE
191.          ENDIF
192.          ENDDO
193.          ENDDO
194.          DO ITENS=1, NTENS
195.             DO IEVAB=1, NEVAB
196.                TENSZ(ITENS, IEVAB, KPGAU)=DBMAT(ITENS, IEVAB)
197.            ENDDO
198.          ENDDO
199.          ENDDO
200.          ENDDO
201.          ENDDO
202.          ENDIF
203. C
204. C*** CALCULO DE LA APORTACION DE RIGIDEZ POR CONVECCION/RADIACION
205. C
206.          IF (NFRON.NE.0) THEN
207.             CALL CONVECC(IELEM, FTFOR, LPROP)
208.             DO IEVAB=1, NEVAB
209.                DO JEVAB=IEVAB, NEVAB
210.                   RIGID(IEVAB, JEVAB)=RIGID(IEVAB, JEVAB)+FTFOR(IEVAB, JEVAB)
211.                ENDDO
212.             ENDDO
213.          ENDIF
214. C
215. C*** CALCULA LA PARTE TRIANGULAR INFERIOR DE LA MATRIZ DE RIGIDEZ
216. C
217.          DO IEVAB=1, NEVAB
218.             DO JEVAB=IEVAB, NEVAB
219.                RIGID(JEVAB, IEVAB)=RIGID(IEVAB, JEVAB)
220.                IF (NTRAN.GT.0) ZMASA(JEVAB, IEVAB)=ZMASA(IEVAB, JEVAB)
221.            ENDDO
222.          ENDDO
223. C
224. C*** ALMACENA MAT DE RIGIDEZ ELEMENTAL, LA MAT DB PARA EL CALCULO
225. C DE TENSIONES Y LAS COORDENADAS DE LOS PUNTOS DE INTEGRACION
226. C
227.          WRITE(1) RIGID
228.          WRITE(3) TENSZ, CORPG
229.          IF (NTRAN.GT.0) WRITE(4) ZMASA
230.          ENDDO
231.          RETURN
232.          END

```

```

1.          SUBROUTINE SOLUCION
2.          C*****
3.          C
4.          C***          ENSAMBLA Y RESUELVE EL SISTEMA DE ECUACIONES MATRICIALES
5.          C
6.          C*****
7.          IMPLICIT NONE
8.          INCLUDE 'dtsgral.f'
9.          INCLUDE 'data.f'
10.         INCLUDE 'calculo.f'
11.         INCLUDE 'solu.f'

```

```

12. C*****
13. C
14. C*** SOLUCION DEL SISTEMA DE ECUACIONES UTILIZANDO ELIMINACION
15. C GAUSSIANA DIRECTA
16. C
17. CALL ENSAMBLA
18. CALL REDUCE
19. CALL SUSTITUIR
20. RETURN
21. END

1. SUBROUTINE SUAV1
2. C*****
3. C
4. C*** CALCULA LAS TENSIONES SUAVIZADAS EN LOS NODOS
5. C
6. C*****
7. implicit none
8. INCLUDE 'dtsgral.f'
9. INCLUDE 'data.f'
10. INCLUDE 'gaussdat.f'
11. INCLUDE 'calculo.f'
12. INCLUDE 'solucuas.f'
13. INTEGER*2 NSTR1, MPOST, INDC, NNOD1, NNOD2, IELEM, ICONT, INODE,
14. . LPROP, LNOD, IGAUS, JGAUS, KPGAU, MTENS, LNOD1, LPOS1,
15. . MPOS1, IPOST, LNOD2, LNOD3, LPOS1, MPOS1, NPOS1, IPNOD,
16. . MTNOD, ITNOD, IDIME, JTNOD, ISTR1, JSTR1, KTNOD, KGAUS,
17. . INSTR1, MPNOD
18. REAL*4 EGASP, EXISP, ETASP, TENSG, DJACB,
19. . DVOLU, DAREA, ELONG
20. DIMENSION TENSG(3)
21. WRITE(29) IDIME
22. IF(NNODE.EQ.20) THEN
23. WRITE(29) (0)
24. RETURN
25. ENDIF
26. REWIND (8)
27. REWIND (10)
28. EGASP=0.0
29. NSTR1=NTENS
30. MPOST=0
31. C
32. C*** CALCULO DE TENSIONES
33. C
34. WRITE(29) NSTR1
35. MTNOD=NSTR1*MPNOD
36. DO ITNOD=1, MTNOD
37. NAUXI(ITNOD)=0
38. TENST(ITNOD)=0.0
39. SMOTI(ITNOD)=0.0
40. ENDDO
41. INDC=0
42. NNOD1=0
43. IF (NNODE.EQ.8.OR.NNODE.EQ.9) THEN
44. INDC=1
45. NNOD1=7
46. NNOD2=8
47. IF(NNODE.EQ.9) NNOD2=9
48. NNODE=4
49. ENDIF
50. IF (NNODE.EQ.6) THEN
51. INDC=1

```

```

52.          NNOD1=5
53.          NNOD2=5
54.          NNODE=3
55.      ENDIF
56.      IF (INDC.NE.0) THEN
57.          OPEN(11,FILE='TENS.TMP',FORM='UNFORMATTED')
58.          WRITE(11) LNODS
59.          DO IELEM=1,NELEM
60.              ICONT=0
61.              DO INODE=1,NNOD1,2
62.                  ICONT=ICONT+1
63.                  LNODS (IELEM,ICONT)=LNODS (IELEM,INODE)
64.              ENDDO
65.          ENDDO
66.      ENDIF
67.      NTNOD=NSTR1*NNODE
68.      C
69.      C*** BUCLE SOBRE CADA ELEMENTO
70.      C
71.          DO IELEM=1,NELEM
72.              LPROP=ABS (MATNU (IELEM) )
73.          C
74.      C*** CALCULA LAS COORDENADAS DE LOS PUNTOS NODALES DEL ELEMENTO
75.      C
76.          DO INODE=1,NNODE
77.              LNODE=LNODS (IELEM,INODE)
78.              DO IDIME=1,NDIME
79.                  COREL (IDIME,INODE)=COORD (LNODE, IDIME)
80.              ENDDO
81.          ENDDO
82.      C
83.      C*** INICIALIZA LAS MATRIZ DE SUAVIZADO ELEMENTALES
84.      C
85.          DO ITNOD=1,NTNOD
86.              TENS1 (ITNOD)=0.0
87.              DO JTNOD=1,NTNOD
88.                  XMASA (ITNOD,JTNOD)=0.0
89.              ENDDO
90.              DO ISTR1=1,NSTR1
91.                  XFORT (ITNOD,ISTR1)=0.0
92.                  XFORM (ISTR1,ITNOD)=0.0
93.              ENDDO
94.          ENDDO
95.          KPGAU=0
96.          IF (NDIME.EQ.1) THEN
97.              DO IGAUS=1,NGAUS
98.                  KPGAU=KPGAU+1
99.                  EXISP=POSPG (IGAUS)
100.                 READ (10) TENSG
101.                 CALL FFORMA (EXISP,0,0,NDIME,NNODE,FFORM,DERIV)
102.                 CALL JACOBM (IELEM,DJACB,KPGAU)
103.                 DVOLU=DJACB*PESPG (IGAUS)
104.                 DO ITNOD=1,NTNOD
105.                     DO ISTR1=1,NSTR1
106.                         XFORM (ISTR1,ITNOD)=0.0
107.                         XFORT (ITNOD,ISTR1)=0.0
108.                     ENDDO
109.                 ENDDO
110.                 DO INODE=1,NNODE
111.                     MTENS=(INODE-1)*NSTR1
112.                     DO ISTR1=1,NSTR1
113.                         XFORM (ISTR1,MTENS+ISTR1)=FFORM (INODE)
114.                         XFORT (MTENS+ISTR1,ISTR1)=FFORM (INODE)

```

```

115.          ENDDO
116.      ENDDO
117.      DO ITNOD=1,NTNOD
118.          DO JSTR1=1,NSTR1
119.              TENS1(ITNOD)=TENS1(ITNOD)+
120.                  XFORT(ITNOD,JSTR1)*TENSG(JSTR1)*DVOLU
121.          DO KTNOD=1,NTNOD
122.              XMASA(ITNOD,KTNOD)=XMASA(ITNOD,KTNOD)+
123.                  XFORT(ITNOD,JSTR1)*XFORM(JSTR1,KTNOD)*DVOLU
124.          ENDDO
125.      ENDDO
126.      ENDDO
127.      ENDDO
128.      ENDIF
129.      IF (NDIME.EQ.2) THEN
130.  C
131.  C*** BUCLE PARA INTEGRACION NUMERICA EN ELEMENTOS CUADRILATEROS O
132.  C      TRIANGULARES
133.  C
134.          IF (NNODE.EQ.3.OR.NNODE.EQ.6) THEN
135.  C
136.  C*** ELEMENTOS TRIANGULARES
137.  C
138.          DO IGAUS=1,NGAUS
139.              KPGAU=KPGAU+1
140.              EXISP=POSGT(IGAUS)
141.              ETASP=POSGT(NGAUS+IGAUS)
142.              READ(10)TENS
143.              CALL FFORMA(EXISP,ETASP,EGASP,NDIME,NNODE,FFORM,DERIV)
144.              CALL JACOBM(IELEM,DJACB,KPGAU)
145.              DVOLU=DJACB*PESGT(IGAUS)
146.              DO ITNOD=1,NTNOD
147.                  DO ISTR1=1,NSTR1
148.                      XFORM(ISTR1,ITNOD)=0.0
149.                      XFORT(ITNOD,ISTR1)=0.0
150.                  ENDDO
151.              ENDDO
152.              DO INODE=1,NNODE
153.                  MTENS=(INODE-1)*NSTR1
154.                  DO ISTR1=1,NSTR1
155.                      XFORM(ISTR1,MTENS+ISTR1)=FFORM(INODE)
156.                      XFORT(MTENS+ISTR1,ISTR1)=FFORM(INODE)
157.                  ENDDO
158.              ENDDO
159.              DO ITNOD=1,NTNOD
160.                  DO JSTR1=1,NSTR1
161.                      TENS1(ITNOD)=TENS1(ITNOD)+
162.                          XFORT(ITNOD,JSTR1)*TENSG(JSTR1)*DVOLU
163.                  DO KTNOD=1,NTNOD
164.                      XMASA(ITNOD,KTNOD)=XMASA(ITNOD,KTNOD)+
165.                          XFORT(ITNOD,JSTR1)*XFORM(JSTR1,KTNOD)*DVOLU
166.                  ENDDO
167.              ENDDO
168.              ENDDO
169.              ENDDO
170.          ELSE
171.  C
172.  C*** ELEMENTOS CUADRILATEROS
173.  C
174.          DO IGAUS=1,NGAUS
175.              DO JGAUS=1,NGAUS
176.                  KPGAU=KPGAU+1
177.                  EXISP=POSPG(IGAUS)

```



```

178.          ETASP=POSPG (JGAUS)
179.          READ (10) TENS
180.          CALL FFORMA (EXISP, ETASP, EGASP, NDIME, NNODE, FFORM, DERIV)
181.          DO ITNOD=1, NTNOD
182.              DO ISTR1=1, NSTR1
183.                  XFORM (ISTR1, ITNOD) =0.0
184.                  XFORT (ITNOD, ISTR1) =0.0
185.              ENDDO
186.          ENDDO
187.          DO INODE=1, NNODE
188.              MTENS= (INODE-1) *NSTR1
189.              DO ISTR1=1, NSTR1
190.                  XFORM (ISTR1, MTENS+ISTR1) =FFORM (INODE)
191.                  XFORT (MTENS+ISTR1, ISTR1) =FFORM (INODE)
192.              ENDDO
193.          ENDDO
194.          CALL JACOBM (IELEM, DJACB, KPGAU)
195.          DVOLU=DJACB*PESPG (IGAUS) *PESPG (JGAUS)
196.          DO ITNOD=1, NTNOD
197.              DO JSTR1=1, NSTR1
198.                  TENS1 (ITNOD) =TENS1 (ITNOD) +
199.                      XFORT (ITNOD, JSTR1) *TENS (JSTR1) *DVOLU
200.              DO KTNOD=1, NTNOD
201.                  XMASA (ITNOD, KTNOD) =XMASA (ITNOD, KTNOD) +
202.                      XFORT (ITNOD, JSTR1) *XFORM (JSTR1, KTNOD) *DVOLU
203.              ENDDO
204.          ENDDO
205.          ENDDO
206.          ENDDO
207.          ENDDO
208.          ENDIF
209.          ENDIF
210.          IF (NDIME.EQ.3) THEN
211.      C
212.      C*** BUCLE PARA INTEGRACION NUMERICA EN ELEMENTOS SOLIDOS
213.      C      TRIDIMENSIONALES HEXAGONALES
214.      C
215.          DO IGAUS=1, NGAUS
216.              DO JGAUS=1, NGAUS
217.                  DO KGAUS=1, NGAUS
218.                      KPGAU=KPGAU+1
219.                      EXISP=POSPG (IGAUS)
220.                      ETASP=POSPG (JGAUS)
221.                      EGASP=POSPG (KGAUS)
222.                      READ (10) TENS
223.                      CALL FFORMA (EXISP, ETASP, EGASP, NDIME, NNODE, FFORM, DERIV)
224.                      CALL JACOBM (IELEM, DJACB, KPGAU)
225.                      DVOLU=DJACB*PESPG (IGAUS) *PESPG (JGAUS) *PESPG (KGAUS)
226.                      DO ITNOD=1, NTNOD
227.                          DO ISTR1=1, NSTR1
228.                              XFORM (ISTR1, ITNOD) =0.0
229.                              XFORT (ITNOD, ISTR1) =0.0
230.                          ENDDO
231.                      ENDDO
232.                      DO INODE=1, NNODE
233.                          MTENS= (INODE-1) *NSTR1
234.                          DO INSTR1=1, NSTR1
235.                              XFORM (ISTR1, MTENS+ISTR1) =FFORM (INODE)
236.                              XFORT (MTENS+ISTR1, ISTR1) =FFORM (INODE)
237.                          ENDDO
238.                      ENDDO
239.                      DO ITNOD=1, NTNOD
240.                          DO JSTR1=1, NSTR1

```

```

241.          TENS1 ( ITNOD ) = TENS1 ( ITNOD ) +
242.          .          XFORT ( ITNOD , JSTR1 ) * TENSG ( JSTR1 ) * DVOLU
243.          DO KTNOD = 1 , NTNOD
244.          .          XMASA ( ITNOD , KTNOD ) = XMASA ( ITNOD , KTNOD ) +
245.          .          XFORT ( ITNOD , JSTR1 ) * XFORM ( JSTR1 , KTNOD ) * DVOLU
246.          ENDDO
247.          ENDDO
248.          ENDDO
249.          ENDDO
250.          ENDDO
251.          ENDDO
252.          ENDIF
253.          IF ( INDC . EQ . 0 ) THEN
254. C
255. C*** ENSAMBLAJE MATRICES DE SUAVIZADO
256. C
257.          DO ITNOD = 1 , NTNOD
258.          DO JTNOD = 1 , NTNOD
259.          IF ( ITNOD . NE . JTNOD ) THEN
260.          XMASA ( ITNOD , ITNOD ) = XMASA ( ITNOD , ITNOD ) + XMASA ( ITNOD , JTNOD )
261.          XMASA ( ITNOD , JTNOD ) = 0 . 0
262.          ENDIF
263.          ENDDO
264.          ENDDO
265.          DO INODE = 1 , NNODE
266.          LNOD1 = ( LNODS ( IELEM , INODE ) - 1 ) * NSTR1
267.          DO ISTR1 = 1 , NSTR1
268.          LPOS1 = ( INODE - 1 ) * NSTR1 + ISTR1
269.          MPOS1 = LNOD1 + ISTR1
270.          IF ( MPOS1 . GT . MPOST ) MPOST = MPOS1
271.          TENST ( MPOS1 ) = TENST ( MPOS1 ) + TENS1 ( LPOS1 )
272.          SMOTI ( MPOS1 ) = SMOTI ( MPOS1 ) + XMASA ( LPOS1 , LPOS1 )
273.          ENDDO
274.          ENDDO
275.          ELSE
276.          CALL INVER
277.          IF ( IERROR . NE . 0 ) THEN
278.          WRITE ( 6 , 710 )
279.          710  FORMAT ( 'ERROR EN EL LA INVERSION DE LA MATRIZ DE ALISADO ' ,
280.          .          ' DE TENSIONES ' )
281.          STOP
282.          ENDIF
283.          DO ITNOD = 1 , NTNOD
284.          TEMPO ( ITNOD ) = 0 . 0
285.          DO JTNOD = 1 , NTNOD
286.          TEMPO ( ITNOD ) = TEMPO ( ITNOD ) + XINV ( ITNOD , JTNOD ) * TENS1 ( JTNOD )
287.          ENDDO
288.          ENDDO
289. C
290. C*** ENSAMBLAJE DE TENSIONES
291. C
292.          DO INODE = 1 , NNODE
293.          LNOD1 = ( LNODS ( IELEM , INODE ) - 1 ) * NSTR1
294.          DO ISTR1 = 1 , NSTR1
295.          LPOS1 = ( INODE - 1 ) * NSTR1 + ISTR1
296.          MPOS1 = LNOD1 + ISTR1
297.          IF ( MPOS1 . GT . MPOST ) MPOST = MPOS1
298.          TENST ( MPOS1 ) = TENST ( MPOS1 ) + TEMPO ( LPOS1 )
299.          NAUXI ( MPOS1 ) = NAUXI ( MPOS1 ) + 1
300.          ENDDO
301.          ENDDO
302.          ENDIF
303.          ENDDO

```

```

304.     IF (INDC.EQ.0) THEN
305.         DO IPOST=1,MPOST
306.             TENST(IPOST)=TENST(IPOST)/SMOTI(IPOST)
307.         ENDDO
308.     ELSE
309.         DO IPOST=1,MPOST
310.             IF (NAUXI(IPOST).NE.0)
311.                 TENST(IPOST)=TENST(IPOST)/FLOAT(NAUXI(IPOST))
312.         ENDDO
313.     ENDIF
314. C
315. C**** INTERPOLA RESULTADOS SI ES NECESARIO
316. C
317.     IF (INDC.NE.0) THEN
318.         REWIND(11)
319.         READ(11) LNODS
320.         DO IELEM=1,NELEM
321.             DO INODE=1,NNOD1,2
322.                 LNOD1=LNODS(IELEM,INODE)-1
323.                 LNOD2=LNODS(IELEM,INODE+1)-1
324.                 IF (INODE.EQ.NNOD1) THEN
325.                     LNOD3=LNODS(IELEM,1)-1
326.                 ELSE
327.                     LNOD3=LNODS(IELEM,INODE+2)-1
328.                 ENDIF
329.                 DO ISTR1=1,NSTR1
330.                     LPOSI=LNOD1*NSTR1+ISTR1
331.                     MPOSI=LNOD3*NSTR1+ISTR1
332.                     NPOSI=LNOD2*NSTR1+ISTR1
333.                     TENST(NPOSI)=(TENST(LPOSI)+TENST(MPOSI))/2.0
334.                 ENDDO
335.             ENDDO
336.         ENDDO
337.         CLOSE(11,STATUS='DELETE')
338.         IF (NNOD2.EQ.9) THEN
339.             NNODE=9
340.             DO IELEM=1,NELEM
341.                 DO ISTR1=1,NSTR1
342.                     LNODE=LNODS(IELEM,9)
343.                     NPOSI=(LNODE-1)*NSTR1+ISTR1
344.                     TENST(NPOSI)=0.0
345.                     DO INODE=1,NNOD1,2
346.                         LPOSI=(LNODS(IELEM,INODE)-1)*NSTR1+ISTR1
347.                         TENST(NPOSI)=TENST(NPOSI)+TENST(LPOSI)
348.                     ENDDO
349.                     TENST(NPOSI)=TENST(NPOSI)/4.0
350.                 ENDDO
351.             ENDDO
352.         ELSE
353.             NNODE=NNOD1+1
354.         ENDIF
355.     ENDIF
356. C
357. C
358. C*** IMPRESION DE FLUJOS SUAVIZADAS
359. C
360.     WRITE(6,904)
361.     904 FORMAT(/,10X,'FLUJOS ALISADAS NODALES',/)
362.     WRITE(6,905)
363.     905 FORMAT(1X,'NODO',2X,'X-FLUJ.',5X,'Y-FLUJ.',5X,'Z-FLUJ. ')
364.     DO IPNOD=1,NPNOD
365.         NPOSI=(IPNOD-1)*NSTR1
366.         DO ISTR1=1,NSTR1

```

```

367.          MPOSI=NPOSI+ISTR1
368.          TENSG(ISTR1)=TENST(MPOSI)
369.          ENDDO
370.          WRITE(6,920) IPNOD,(TENSG(ISTR1),ISTR1=1,NSTR1)
371.          MPNOD=(IPNOD)
372.          WRITE(29) MPNOD,(TENSG(ISTR1),ISTR1=1,NSTR1)
373.          ENDDO
374.          WRITE(8) TENST
375.          920 FORMAT(I5,6E12.5)
376.          RETURN
377.          END

```

```

1.          SUBROUTINE SUSTITUIR
2.          C*****
3.          C
4.          C***          REALIZA LA SUSTITUCION HACIA ATRAS
5.          C
6.          C*****
7.          IMPLICIT NONE
8.          INCLUDE 'dtsgral.f'
9.          INCLUDE 'solu.f'
10.         INTEGER*2          MPNOD,NEONS,IEONS,NEON1,NBACK,NBAC1,ICOLS,
11.         .                  KOUNT,IPNOD,NGUSH,NCONT,NGISH,IGASH,MCONT,
12.         .                  NLOCA,IGDLN
13.         REAL*4            PIVOT,RESID
14.
15.         C*****
16.         NEONS=NSVAB
17.         DO 5 IEONS=1,NEONS
18.           REACT(IEONS)=0.0
19.         5 CONTINUE
20.         NEON1=NEONS+1
21.         DO 30 IEONS=1,NEONS
22.           NBACK=NEON1-IEONS
23.           PIVOT=ASTIF(NBACK,NBACK)
24.           RESID=ASLOD(NBACK)
25.           IF(NBACK.EQ.NEONS) GO TO 20
26.           NBAC1=NBACK+1
27.           DO 10 ICOLS=NBACK,NEONS
28.             RESID=RESID-ASTIF(NBACK,ICOLS)*DESPL(ICOLS)
29.           10 CONTINUE
30.           20 IF(IFFIX(NBACK).EQ.0) DESPL(NBACK)=RESID/PIVOT
31.             IF(IFFIX(NBACK).EQ.1) DESPL(NBACK)=FIXED(NBACK)
32.             IF(IFFIX(NBACK).EQ.1) REACT(NBACK)=-RESID
33.           30 CONTINUE
34.           KOUNT=0
35.           WRITE(6,900)
36.           900 FORMAT(/,5X,'TEMPERATURAS')
37.           WRITE(6,905)
38.           905 FORMAT(6X,'NODO',2X,'TEMPERATURA')
39.           DO 450 IPNOD=1,NPNOD
40.             NCONT=IPNOD*NGDLN
41.             NGISH=NCONT-NGDLN+1
42.             MPNOD=(IPNOD)
43.             WRITE(6,920) IPNOD,(DESPL(IGASH),IGASH=NGISH,NCONT)
44.             WRITE(29) MPNOD,(DESPL(IGASH),IGASH=NGISH,NCONT)
45.           450 CONTINUE
46.           920 FORMAT(I10,3E14.6)
47.           WRITE(6,925)
48.           925 FORMAT(/,'0',5X,'REACCIONES')
49.           WRITE(6,930)
50.           930 FORMAT('0',5X,'NODO',6X,'REACCION')

```



```

51.          MCONT=0
52.          DO 481 IPNOD=1,NPNOD
53.          NLOCA=(IPNOD-1)*NGDLN
54.          DO 484 IGDLN=1,NGDLN
55.          NGUSH=NLOCA+IGDLN
56.          IF(IFFIX(NGUSH).GT.0) GO TO 483
57.      484 CONTINUE
58.          GO TO 481
59.      483 MCONT=MCONT+1
60.          DO 482 IGDLN=1,NGDLN
61.          NCONT=NLOCA+IGDLN
62.      482 TREAC(IGDLN)=REACT(NCONT)
63.          WRITE(6,945) IPNOD, (TREAC(IGDLN), IGDLN=1,NGDLN)
64.      481 CONTINUE
65.      945 FORMAT(I10,3E14.6)
66.          RETURN
67.          END

1.          SUBROUTINE TENSIONES
2.          C*****
3.          C
4.          C***          CALCULA LAS TENSIONES Y ESFUERZOS EN LOS PUNTOS DE GAUSS
5.          C
6.          C*****
7.          IMPLICIT NONE
8.          INCLUDE 'dtsgral.f'
9.          INCLUDE 'data.f'
10.         INCLUDE 'calculo.f'
11.         INCLUDE 'solu.f'
12.         INTEGER*2          NSTR1, IELEM, INODE, LNODE, NPOSN, IGDLN,
13.         .                  KPGAU, IGAUS, JGAUS, KGAUS, ITENS, KCONT, IDIME
14.         REAL*4             TENSG, DESEL
15.         DIMENSION          TENSG(3), DESEL(1,20)
16.         C*****
17.         REWIND(3)
18.         REWIND(10)
19.         WRITE(6,900)
20.     900 FORMAT(/,10X, 'TENSIONES',/)
21.         C
22.         C*** SELECCION DE TIPO DE ANALISIS
23.         C
24.         IF(NDIME.EQ.1) WRITE(6,905)
25.         IF(NDIME.EQ.2) WRITE(6,906)
26.         IF(NDIME.EQ.3) WRITE(6,907)
27.         NSTR1=NTENS
28.     905 FORMAT(' ', 'G.P.', 2X, 'X-COORD', 4X, 'X-FLUJ. ')
29.     906 FORMAT(' ', 'G.P.', 2X, 'X-COORD', 2X, 'Y-COORD', 4X, 'X-FLUJ.', 5X,
30.         .      'Y-FLUJ. ')
31.     907 FORMAT(' ', 'G.P.', 2X, 'X-COORD', 2X, 'Y-COORD', 2X, 'Z-COORD', 4X,
32.         .      'X-FLUJ.', 5X, 'Y-FLUJ.', 5X, 'Z-FLUJ. ')
33.         C
34.         C*** BUCLE SOBRE CADA ELEMENTO
35.         C
36.         DO IELEM=1,NELEM
37.         C
38.         C*** LEE LA MATRIZ DB, Y LAS COORD. DE LOS PUNTOS DE INTEGRACION
39.         C
40.         READ(3) TENSZ, CORPG
41.         C
42.         WRITE(6,910) IELEM
43.     910 FORMAT(/,5X, 'ELEMENTO NO.=', I5)
44.         C

```

```

45. C*** IDENTIFICA LAS TEMPERATURAS Y LOS PUNTOS NODALES DEL ELEMENTO
46. C
47.     DO INODE=1,NNODE
48.         LNODE=LNODS(IELEM,INODE)
49.         NPOSN=(LNODE-1)*NGDLN
50.         DO IGDLN=1,NGDLN
51.             NPOSN=NPOSN+1
52.             DESEL(IGDLN,INODE)=DESPL(NPOSN)
53.         ENDDO
54.     ENDDO
55.     KPGAU=0
56.     IF (NDIME.EQ.1) THEN
57. C
58. C***         BUCLE SOBRE CADA PUNTO DE INTEGRACION
59. C
60.         DO IGAUS=1,NGAUS
61.             KPGAU=KPGAU+1
62. C
63. C*** CALCULO DE TENSIONES EN LOS PUNTOS DE INTEGRACION
64. C
65.         DO ITENS=1,NTENS
66.             TENSG(ITENS)=0.0
67.             KCONT=0
68.             DO INODE=1,NNODE
69.                 DO IGDLN=1,NGDLN
70.                     KCONT=KCONT+1
71.                     TENSG(ITENS)=TENSG(ITENS)+
72.                     TENSZ(ITENS,KCONT,KPGAU)*DESEL(IGDLN,INODE)
73.                 ENDDO
74.             ENDDO
75.         ENDDO
76. C
77. C*** SALIDA DE TENSIONES
78. C
79.         WRITE(6,920) KPGAU,(CORPG(IDIME,KPGAU),IDIME=1,NDIME),
80.         .         (TENSG(ITENS),ITENS=1,NTENS)
81.         WRITE(10)TENSG
82.         ENDDO
83.     ENDIF
84.     IF (NDIME.EQ.2) THEN
85.         IF (NNODE.EQ.4.OR.NNODE.EQ.8.OR.NNODE.EQ.9) THEN
86. C
87. C***BUCLE SOBRE CADA PUNTO DE INTEGRACION EN ELEMENTOS CUADRILATEROS
88. C
89.         DO IGAUS=1,NGAUS
90.             DO JGAUS=1,NGAUS
91.                 KPGAU=KPGAU+1
92. C
93. C*** CALCULO DE TENSIONES EN LOS PUNTOS DE INTEGRACION
94. C
95.         DO ITENS=1,NTENS
96.             TENSG(ITENS)=0.0
97.             KCONT=0
98.             DO INODE=1,NNODE
99.                 DO IGDLN=1,NGDLN
100.                     KCONT=KCONT+1
101.                     TENSG(ITENS)=TENSG(ITENS)+
102.                     TENSZ(ITENS,KCONT,KPGAU)*DESEL(IGDLN,INODE)
103.                 ENDDO
104.             ENDDO
105.         ENDDO
106. C
107. C*** ESCRIBE RESULTADOS

```

```

108. C
109.          WRITE(6,920) KPGAU, (CORPG(IDIME,KPGAU), IDIME=1,NDIME),
110.          .          (TENSG(ITENS), ITENS=1,NTENS)
111.          WRITE(10)TENSG
112.          ENDDO
113.          ENDDO
114.          ENDIF
115.          IF(NNODE.EQ.3.OR.NNODE.EQ.6) THEN
116. C
117. C*** BUCLE SOBRE CADA PUNTO DE INTEGRACION EN ELEMENTOS TRIANGULARES
118. C
119.          DO IGAUS=1,NGAUS
120.          KPGAU=KPGAU+1
121. C
122. C*** CALCULO DE TENSIONES EN LOS PUNTOS DE INTEGRACION
123. C
124.          DO ITENS=1,NTENS
125.          TENSG(ITENS)=0.0
126.          KCONT=0
127.          DO INODE=1,NNODE
128.          DO IGDLN=1,NGDLN
129.          KCONT=KCONT+1
130.          TENSG(ITENS)=TENSG(ITENS)+
131.          .          TENSZ(ITENS,KCONT,KPGAU)*DESEL(IGDLN,INODE)
132.          ENDDO
133.          ENDDO
134.          ENDDO
135. C
136. C*** SALIDA DE TENSIONES
137. C
138.          WRITE(6,920) KPGAU, (CORPG(IDIME,KPGAU), IDIME=1,NDIME),
139.          .          (TENSG(ITENS), ITENS=1,NTENS)
140.          WRITE(10)TENSG
141.          ENDDO
142.          ENDIF
143.          ENDIF
144.          IF (NDIME.EQ.3) THEN
145. C
146. C*** BUCLE SOBRE CADA PUNTO DE INTEGRACION EN ELEMENTOS SOLIDOS
147. C TRIDIMENSIONALES
148. C
149.          DO IGAUS=1,NGAUS
150.          DO JGAUS=1,NGAUS
151.          DO KGAUS=1,NGAUS
152.          KPGAU=KPGAU+1
153. C
154. C*** CALCULO DE TENSIONES EN PUNTOS DE INTEGRACION
155. C
156.          DO ITENS=1,NTENS
157.          TENSG(ITENS)=0.0
158.          KCONT=0
159.          DO INODE=1,NNODE
160.          DO IGDLN=1,NGDLN
161.          KCONT=KCONT+1
162.          TENSG(ITENS)=TENSG(ITENS)+
163.          .          TENSZ(ITENS,KCONT,KPGAU)*DESEL(IGDLN,INODE)
164.          ENDDO
165.          ENDDO
166.          ENDDO
167. C
168. C*** ESCRIBE RESULTADOS
169. C
170.          WRITE(6,930) KPGAU, (CORPG(IDIME,KPGAU), IDIME=1,NDIME),

```

```
171. .                               (TENSG(ITENS), ITENS=1, NTENS)
172.     WRITE(10) TENSG
173.     ENDDO
174.     ENDDO
175.     ENDDO
176.     ENDDO
177.     ENDDO
178. 920 FORMAT(I5, 2F10.4, 6E12.5, F10.4)
179. 930 FORMAT(I5, 3F10.4, 6E12.5)
180.     RETURN
181.     END
```


***CALTEP*: Programa para el cálculo transitorio de la ecuación de Poisson**

**F. Zárate
E. Oñate**

***CALTEP*: Programa para el cálculo transitorio de la ecuación de Poisson**

F. Zárate

E. Oñate

Publicación CIMNE Nº 27, Enero 1993

**Centro Internacional de Métodos Numéricos en Ingeniería
Gran Capitán s/n, 08034 Barcelona, España**

***CALTEP*: Programa para el cálculo transitorio de la ecuación de Poisson**

F. Zárate

E. Oñate

Publicación CIMNE Nº 27, Enero 1993

Centro Internacional de Métodos Numéricos en Ingeniería

Gran Capitán s/n, 08034 Barcelona, España

INDICE

CALTEP Programa para el cálculo transitorio de la ecuación de Poisson.

1	INTRODUCCION.....	1
2	ECUACIONES BASICAS.....	1
3	SOLUCION POR EL METODO DE LOS ELEMENTOS FINITOS	4
3.1	FORMULACION BASICA	4
3.2	DISCRETIZACION POR ELEMENTOS FINITOS.....	5
3.3	ECUACIONES DE LA DISCRETIZACION.....	6
3.3.1	CASO ESTACIONARIO	6
3.3.2	CASO TRANSITORIO	6
3.3.2.1	DIAGONALIZACION POR SUMA DE FILAS.....	7
3.3.2.2	DIAGONALIZACION POR CONSERVACION DE LA MASA EQUIVALENTE	7
4	CARACTERISTICAS DEL PROGRAMA CALTEP	8
5	ORGANIZACION GENERAL CALTEP	9
5.1	ETAPAS BASICAS. DIAGRAMA DE FLUJO PRINCIPAL	9
5.2	SELECCION DE LOS NOMBRES DE LAS VARIABLES	12
5.3	TRANSMISION DE INFORMACION ENTRE SUBROUTINAS	13
5.4	LISTADO DE LA SUBROUTINA PRINCIPAL DE CALTEP	14
6	DESCRIPCION DE LA SUBROUTINA DATOS	16
6.1	PARAMETROS DE CONTROL	16
6.2	DATOS GEOMETRICOS	17
6.3	CONDICIONES DE NODOS PRESCRITOS	17
6.4	PROPIEDADES DEL MATERIAL	18
6.5	CONDICIONES INICIALES DE TEMPERATURA	18
6.6	CONDICIONES DE FLUJO DE CONVECCION / RADIACION EN LA FRONTERA	19
6.7	SUBROUTINA GAUSS	19
6.8	PREPARACION AUTOMATICA DE DATOS	19
6.9	SUBROUTINA DE CONTROL DE DATOS	20
7	DESCRIPCION DE LAS SUBROUTINAS DE LA MATRIZ DE RIGIDEZ	20
7.1	MATRIZ DE RIGIDEZ: SUBROUTINA RIGIMAT	20
7.1.1	CASO ESTACIONARIO	20
7.1.2	CASO TRANSITORIO	21
7.2	MATRIZ DE RIGIDEZ: SUBROUTINA CONVECC	23
8	DESCRIPCION DE LA SUBROUTINA FUERZAS	25
8.1	CONSIDERACIONES GENERALES	25
8.2	FLUJOS PUNTUALES NODALES	26
8.3	GENERACION INTERNA	26
8.4	TEMPERATURA AMBIENTAL Y FLUJO EXTERIOR REPARTIDO SOBRE UN LADO	27
9	SOLUCION DEL SISTEMA DE ECUACIONES. SUBROUTINA SOLUCION	28
10	CALCULO DE LOS FLUJOS ELEMENTALES. SUBROUTINA TENSION	32
11	EJEMPLOS DE UTILIZACION DEL PROGRAMA CALTEP	33
11.1	FLUJO A TRAVES DE UNA BARRA DELGADA	33
11.2	GENERACION INTERNA DE CALOR EN UN DOMINIO CUADRADO	35
11.2.1	CASO ESTACIONARIO	35
11.2.2	CASO TRANSITORIO $A=1.0$	41
11.2.3	CASO TRANSITORIO $A=2/3$	43
	REFERENCIAS	47
	APENDICE I	I.i
	I.I INSTRUCCIONES PARA ENTRADA DE DATOS	I.i
	I.II LISTADO DE VARIABLES DEL PROGRAMA CALTEP	I.ii
	APENDICE II	II.ii
	II.I LISTADO DEL PROGRAMA CALTEP	II.ii

1 INTRODUCCION

Esta publicación explica la utilización de un programa de elementos finitos que permita resolver la ecuación de Poisson transitoria, que rige una gran cantidad de problemas físicos como son la transmisión del calor a través de diversos medios, el flujo de un liquido a través de un medio permeable, problemas de magnetismo, etc.

En la publicación describe las etapas que intervienen en un programa de elementos finitos para cálculo de la ecuación de Poisson transitoria incidiendo principalmente en la metodología general de la programación de las diferentes subrutinas, así como en su aplicación a varios problemas. En la última parte de esta publicación se presentan diversos ejemplos de aplicación del programa, así como un listado completo del mismo, una descripción de sus variables más significativas y las instrucciones para la entrada de datos.

2 ECUACIONES BASICAS

La modelización del comportamiento de muchos fenómenos naturales puede describirse utilizando la formulación diferencial cuasi armónica conocida en el argot matemático como la ecuación de Poisson.

Dicha ecuación puede modelizar, entre otros problemas, el mecanismo de conducción de calor a través de un cuerpo, o bien, el flujo de un liquido en un medio permeable. En un siguiente apartado se dará una descripción de los diversos fenómenos naturales que rige esta ecuación, así como la equivalencia de las variables usadas. Sin embargo, para poder dar un significado tangible a la descripción de las ecuaciones que se utilizaran, se ha tomado como ejemplo concreto el problema de la conducción de calor, ya que se puede considerar como la aplicación mas sencilla y generalizada.

La ecuación de Poisson para un dominio bidimensional puede expresarse en la forma siguiente:

$$\rho c \frac{\partial \phi}{\partial t} = \mathbf{D} \nabla^2 \phi + \rho r \quad \text{en } \Omega \quad (\text{i})$$

En donde ϕ representa la temperatura, t la variable tiempo, ρ la densidad del dominio, c el calor específico y ρr la densidad por fuente de calor interno; mientras que \mathbf{D} corresponde a la matriz constitutiva, formada por las conductividades térmicas, K , para las distintas dimensiones en que se describe el dominio Ω ; de esta manera, para un dominio bidimensional corresponde a:

$$\mathbf{D} = \begin{bmatrix} K_x & \mathbf{0} \\ \mathbf{0} & K_y \end{bmatrix} \quad (\text{ii})$$

Alternativamente \mathbf{D} queda descrita para el caso uni y tridimensional como:

$$\mathbf{D} = [K_x] \text{ caso unidimensional.} \quad (\text{iii})$$

$$\mathbf{D} = \begin{bmatrix} K_x & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & K_y & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & K_z \end{bmatrix} \text{ caso tridimensional.} \quad (\text{iv})$$

Las condiciones de contorno a las que se encuentra sujeta la formulación anterior se esquematizan en la Figura 1, y son conocidas en términos matemáticos como las condiciones de (a) *Dirichlet* que fija la temperatura ϕ , a un valor fijo y conocido de antemano sobre un contorno particular, y (b) *Cauchy* que fija el gradiente de la temperatura normal a la superficie.

$$\text{(a) } \phi - \bar{\phi} = \mathbf{0} \text{ en } \Gamma_\phi \quad (\text{v})$$

$$\text{(b) } \mathbf{n}^T \mathbf{q} + \alpha(\phi - \phi_{ext}) + \bar{\mathbf{q}} = \mathbf{0} \text{ en } \Gamma_q \quad (\text{vi})$$

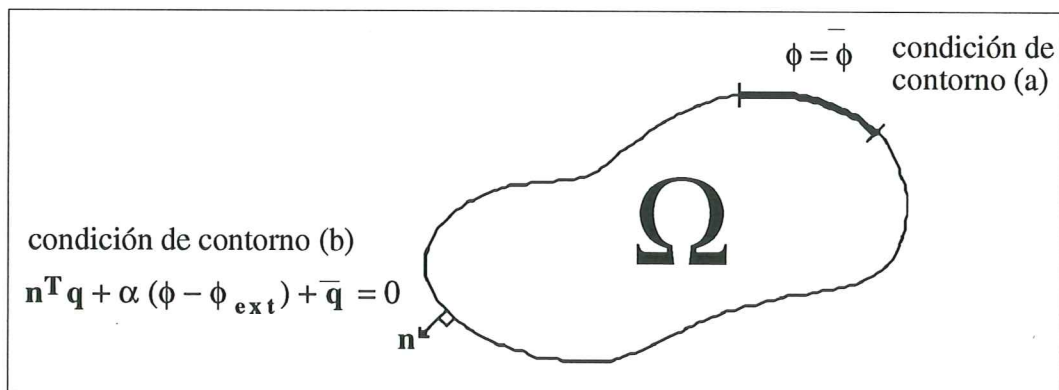


Figura 1 Región bidimensional con las condiciones de contorno permisibles.

En las ecuaciones anteriores $\bar{\phi}$ es la temperatura con valor conocido en la frontera, α representa al coeficiente de convección-radiación, $\bar{\mathbf{q}}$ es el flujo o gradiente de la temperatura, con valor conocido en la frontera, ϕ_{ext} es la temperatura en el exterior del dominio, mientras que el resto de las variables quedan definidas con las siguientes expresiones:

Vector de normales al contorno:

$$\mathbf{n} = [n_x, n_y]^T \quad (\text{vii})$$

Vector de gradientes de la temperatura:

$$\mathbf{q}_n = [q_x, q_y] = -\mathbf{D}\mathbf{g} = -\mathbf{D} \nabla\phi \quad (\text{viii})$$

En donde el gradiente ∇ corresponde a:

$$\nabla = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right]^T \quad (\text{ix})$$

Resulta inmediato la expresión de las ecuaciones (vii), (viii) y (ix) para los casos uni y tridimensionales, por lo que se omiten sus desarrollos.

Dependiendo de los valores escogidos para los parámetros de la ecuación (vi) es posible reproducir los siguientes casos en el contorno:

a) Contorno aislante. Matemáticamente conocido como la condición de *Newman* o condición de contorno natural que representa un flujo de temperatura cero en la interfase del dominio con el medio circundante, de manera que:

$$\mathbf{n}^T \mathbf{q} = 0 \text{ por lo que } \bar{q} = 0 \text{ y } \alpha = 0 \quad (\text{vi.a})$$

b) Contorno con perdida o aporte de flujo externo. La condición señalada queda representada matemáticamente por:

$$\mathbf{n}^T \mathbf{q} = -\bar{q} \text{ por lo que } \alpha = 0 \quad (\text{vi.b})$$

c) Contorno con perdida o aporte de calor por convección - radiación. En este caso existe una interacción del calor existente en el medio exterior ϕ_{ext} y la temperatura del dominio ϕ que de acuerdo con las leyes de la termodinámica se presentara un flujo en la interfase, representado por:

$$\mathbf{n}^T \mathbf{q} = -\alpha(\phi - \phi_{ext}) \text{ con lo que } \bar{q} = 0 \quad (\text{vi.c})$$

El factor α estrictamente representa el fenómeno de convección, mientras que el fenómeno de radiación sigue una ley mas compleja pero que a efectos de simplificación se puede expresar de la misma manera que la expresada en (vi.c), por lo que α representa la contribución de ambos

Como se ha mencionado en párrafos anteriores, los términos de las ecuaciones (i) a (ix) tienen diferentes significados físicos, dependiendo del problema que se modelice. A continuación se listan algunos de los casos que son mas comunes:

	Problema Térmico	Filtración en medios poroso	Electromagnetismo	Torsión de barras
ϕ	Temperatura	Altura Piezométrica	Potencial magnético	Esfuerzo torsor
\mathbf{D}	Matriz de conductividad térmica	Matriz de permeabilidad	Matriz de reluctancia	Matriz de rigidez
r	Fuente de calor interno por unidad de masa	Caudal aportado en el medio	Fuente magnética interna	-
\mathbf{q}	vector de flujo de calor	Flujo de agua en el contorno	Flujo magnético externo	Momento torsor
ρ	Densidad	Densidad	Densidad	Densidad
c	Calor específico por unidad de masa	-	Aportación magnética por unidad de masa	-
α	Coefficiente de convección-radiación	-	-	-

Figura 2 Reinterpretación física de las variables de la ecuación de Poisson.

3 SOLUCION POR EL METODO DE LOS ELEMENTOS FINITOS

3.1 Formulación Básica.

La aplicación del método de los elementos finitos exige como punto de partida la existencia de una forma integral expresando el mecanismo global del sistema. Dicha forma integral puede obtenerse aplicando el método de los residuos ponderados a la ecuación diferencial (i) y a la condición de contorno (vi) como:

$$\int_{\Omega} [\nabla^T \mathbf{D} \nabla \phi + \rho r] \partial \Omega - \int_{\Omega} \rho c \frac{\partial \phi}{\partial t} \partial \Omega + \oint_{\Gamma_q} [\mathbf{n}^T \mathbf{D} \nabla \phi + \alpha(\phi - \phi_{ext}) + \bar{\mathbf{q}}] \partial \Gamma_q = 0 \quad (\text{x})$$

La condición de contorno (v) no necesita incluirse en la expresión anterior puesto que se satisface imponiendo directamente el valor de ϕ al valor prescrito en los contornos adecuados (condición forzada).

Tras integrar por partes el termino $\nabla^T \mathbf{D} \nabla \phi$ y reagrupar la ecuación (x) se obtiene (después de hacer $\mathbf{W} = -\mathbf{W}$)

$$\int_{\Omega} \mathbf{W}^T \rho c \frac{\partial \phi}{\partial t} \partial \Omega + \int_{\Omega} \nabla^T \mathbf{W}^T \mathbf{D} \nabla \phi \partial \Omega + \oint_{\Gamma_q} \mathbf{W}^T \alpha \phi \partial \Gamma_q = \int_{\Omega} \mathbf{W}^T \rho r \partial \Omega - \oint_{\Gamma_q} \mathbf{W}^T \bar{\mathbf{q}}_n \partial \Gamma_q + \oint_{\Gamma_q} \mathbf{W}^T \alpha \phi_{ext} \partial \Gamma_q - \oint_{\Gamma_{\phi}} \mathbf{W}^T \mathbf{q}_n \partial \Gamma_{\phi} \quad (\text{xi})$$

Observese que la ultima integral de (xi) incluye el flujo normal en los contornos donde ϕ esta prescrita. En la practica este flujo se calcula "a posteriori", una vez obtenidas las temperaturas en todos los nodos.

3.2 Discretización por Elementos Finitos

Después de discretizar el dominio en elementos en la forma clásica, (figura 3), las temperaturas se interpolan en el interior de cada elemento como:

$$\phi = \sum N_i \phi_i = \mathbf{N} \mathbf{a}^{(e)} \quad (\text{xii})$$

donde N_i son las funciones de forma definidas en cada elemento y $\mathbf{a}^{(e)}$ contiene los valores de las temperaturas nodales del elemento (e).

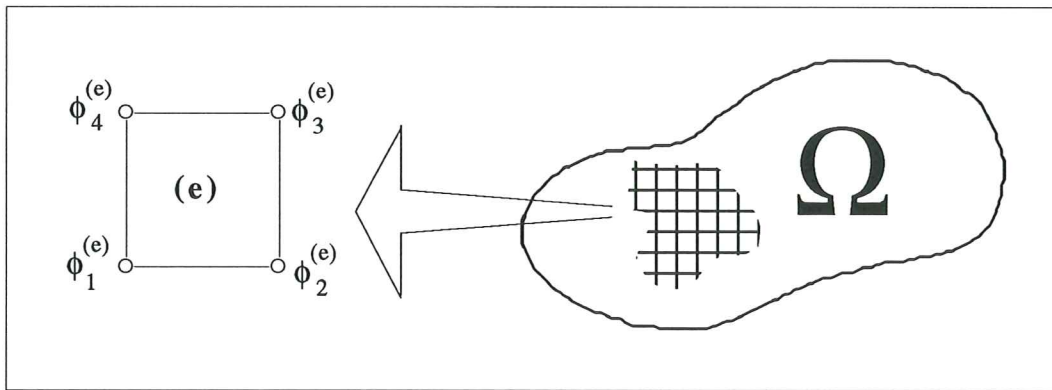


Figura 3 Discretización del dominio en Elementos Finitos.

El vector de gradientes en cada elemento se obtiene por:

$$\mathbf{g} = \nabla \phi = \nabla \mathbf{N} \mathbf{a}^{(e)} = \mathbf{B} \mathbf{a}^{(e)} \quad (\text{xiii})$$

donde

$$\mathbf{B} = [\mathbf{B}_1, \dots, \mathbf{B}_n] \quad (\text{xiv})$$

siendo

$$\mathbf{B}_i = \left[\frac{\partial N_i}{\partial x} \right], \quad \mathbf{B}_i = \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} \quad \text{y} \quad \mathbf{B}_i = \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} \quad (\text{xv})$$

para problemas uni, bi y tridimensionales, respectivamente.

El vector de flujos puede calcularse en función de los valores nodales como:

$$\mathbf{q} = -\mathbf{DBa}^{(e)} \quad (\text{xvi})$$

3.3 Ecuaciones de la discretización

Sustituyendo la discretización (xii) y (xiii) en (xi) y haciendo $\mathbf{W}=\mathbf{N}$ (método de Galerkin) se obtiene un sistema matricial de ecuaciones que puede escribirse en la forma:

$$\mathbf{M} \frac{\partial \mathbf{a}}{\partial t} + \mathbf{Ka} = \mathbf{f} \quad (\text{xvii})$$

En (xvii), \mathbf{a} es el vector de incógnitas que contiene la temperatura de todos los nodos de la malla, y \mathbf{M} , \mathbf{K} , y \mathbf{f} son la matriz de masa, la matriz de rigidez y el vector de fuerzas nodales que pueden obtenerse ensamblando las contribuciones elementales definidas por

$$\mathbf{M}^{(e)} = \int_{\Omega^{(e)}} \rho c \mathbf{N}^T \mathbf{N} \, d\Omega^{(e)} \quad (\text{xviii})$$

$$\mathbf{K}^{(e)} = \int_{\Omega^{(e)}} \mathbf{B}^T \mathbf{DB} \, d\Omega^{(e)} + \alpha \oint_{\Gamma_q^{(e)}} \mathbf{N}^T \mathbf{N} \, d\Gamma_q^{(e)} \quad (\text{xix})$$

$$\mathbf{f}^{(e)} = \int_{\Omega^{(e)}} \mathbf{N}^T \rho r \, d\Omega^{(e)} - \oint_{\Gamma_q^{(e)}} \mathbf{N}^T \bar{q} \, d\Gamma_q^{(e)} + \oint_{\Gamma_q^{(e)}} \mathbf{N}^T \alpha \phi_{ext} \, d\Gamma_q^{(e)} - \oint_{\Gamma_\phi^{(e)}} \mathbf{n}^T \mathbf{N}^T \mathbf{q}_n \, d\Gamma_\phi^{(e)} \quad (\text{xx})$$

3.3.1 Caso Estacionario.

La solución estacionaria implica únicamente resolver el sistema de ecuaciones

$$\mathbf{Ka} = \mathbf{f} \quad (\text{xxi})$$

Obtenidas las temperaturas nodales pueden obtenerse los gradientes térmicos y los flujos de calor en cada punto mediante las ecuaciones (xiii) y (xvi), respectivamente. Asimismo, pueden obtenerse los flujos de calor a través de los contornos con temperaturas prescritas.

3.3.2 Caso Transitorio.

La solución del problema transitorio exige integrar en el tiempo la ecuación (xvii) . Utilizando un esquema de diferencias finitas trapezoidal generalizado se obtiene:

$$\left[\frac{\mathbf{M}}{\Delta t} + \alpha \mathbf{K} \right] \mathbf{a}^t = \alpha \mathbf{f}^t + (1 - \alpha) \mathbf{f}^{t-1} + \left[\frac{\mathbf{M}}{\Delta t} - (1 - \alpha) \mathbf{K} \right] \mathbf{a}^{t-1} \quad (\text{xxii})$$

donde α define el punto de "colocación" de la ecuación diferencial (xvii) , es decir:

$$\mathbf{a}^\alpha = \alpha \mathbf{a}^t + (1 - \alpha) \mathbf{a}^{t-1} \quad (\text{xxiii})$$

La ecuación (xxiii) permite obtener el valor de las temperaturas en el tiempo t en función de las temperaturas en el instante $t-1$ y de los valores de las "fuerzas" en t y $t-1$.

Puede demostrarse que el esquema trapezoidal de la ecuación (xxiii) es incondicionalmente estable para $\alpha \geq \frac{1}{2}$

En la practica se recomienda tomar $\alpha = \frac{2}{3}$ (Galerkin) ó $\alpha = 1$ (Fuertemente implícito) [21]. La integración de $\alpha = 0$ (método explícito de Euler)[21] es ventajosa si la matriz \mathbf{M} es diagonal, puesto que en este caso la solución es explícita y no precisa la inversión de ninguna matriz. La única dificultad en este caso se debe a que el sistema de integración es condicionalmente estable y exige la utilización de incrementos de tiempo pequeños que vienen gobernados por la condición:

$$\Delta t = \frac{L}{\omega} \quad (\text{xxiv})$$

siendo ω el mayor valor propio del sistema homogéneo:

$$[\mathbf{K} + \mathbf{M}\omega] \mathbf{a} = \mathbf{0} \quad (\text{xxv})$$

El programa **CALTEP** permite la integración explícita con dos posibilidades para diagonalizar la matriz \mathbf{M} :

3.3.2.1 Diagonalización por suma de filas.

Se expresa por:

$$\begin{aligned} \mathbf{M}_{Dii} &= \sum_{j=1} \mathbf{M}_{ij} \\ \mathbf{M}_{Dij} &= \mathbf{0} \end{aligned} \quad (\text{xxvi})$$

en donde la diagonal de la matriz de masa es la suma de de todos los elementos contenidos en la fila i -ésima

3.3.2.2 Diagonalización por conservación de la masa:

Se expresa por:

$$\mathbf{M}_{Dii} = p \int_{\Omega} \mathbf{N}_i^T \rho c \mathbf{N}_i \partial \Omega$$

$$\mathbf{M}_{,dij} = \mathbf{0} \quad (\text{xxvii})$$

con

$$\sum_i \mathbf{M}_{,ii} = \int_{\Omega} \rho c \partial \Omega = p \quad (\text{xxviii})$$

de manera que la nueva matriz \mathbf{M} conserve la misma masa que la matriz original.

4 CARACTERISTICAS DEL PROGRAMA CALTEP

En los apartados siguientes se presenta la descripción del programa *CALTEP* para Cálculo Transitorio de la Ecuación de Poisson por el Método de Elementos Finitos con detalles de las subrutinas más relevantes del mismo. La versión de *CALTEP* que se presenta escrita en FORTRAN tiene las siguientes características generales:

Características del material

- Material lineal isótropo.

Elementos utilizables

Pueden utilizarse los elementos isoparamétricos siguientes:

Elemento

- Elemento lineal de dos nodos
- Elemento lineal de tres nodos
- Elemento triangular de tres nodos
- Elemento triangular de 6 nodos
- Elemento cuadrilátero lagrangianos de 4 nodos
- Elemento cuadrilátero serendípito de 8 nodos
- Elemento cuadrilátero lagrangiano de 9 nodos
- Elemento hexagonal serendípito de 20 nodos tridimensional

Condiciones de contorno

Para cada problema se admiten únicamente cargas *estáticas* de los tipos siguientes:

Problema

Tipos de carga

- Problema Unico - Temperaturas puntuales nodales

- Generación Interna de calor
- Flujo exterior uniformemente repartido
sobre los lados de los elementos
(condición de Convección/Radiación)
- Temperatura del medio externo cte.
- Temperatura puntual inicial
(para el problema transitorio)

Una vez familiarizado el lector con los detalles del programa podrá modificarlo fácilmente para incluir otros tipos de elementos y condiciones de contorno, así como para extender su rango de aplicación a otro tipo de problemas.

5 ORGANIZACION GENERAL CALTEP

5.1 Etapas básicas. Diagrama de flujo principal

En la Figura 4 se muestra el diagrama de flujo principal del programa *CALTEP*. Para centrar conceptos definiremos seguidamente las etapas fundamentales asociadas al análisis de un problema cualquiera mediante un programa de elementos finitos, así como la relación de cada etapa con las subrutinas del diagrama de la Figura 4.

Etapa 1: Selección del elemento

La elección del elemento es función de la tipología del problema y de la precisión buscada. Una vez escogido el elemento quedan definidas sus funciones de forma. En el apartado anterior se han definido los tipos de elementos incluidos en la versión del programa que aquí se presenta.

Etapa 2: Discretización de la geometría en elementos finitos

Esta etapa puede representar un porcentaje alto del esfuerzo total de cálculo si la geometría del problema a modelizar es compleja. En ella hay que definir perfectamente la topología de la malla (que de nuevo depende de la geometría y la precisión buscada), las coordenadas de los nodos, las propiedades del material de cada elemento y las condiciones de contorno. Esta etapa se denomina generalmente *preproceso* y puede automatizarse en gran manera si se dispone de los programas de generación de malla adecuados [5], [6], [7]. Esta automatización es mucho más esencial si se utilizan técnicas de solución adaptables.

Es importante destacar que el coste de la solución del sistema de ecuaciones global depende en gran manera de: a) la numeración de los nodos de la malla (ejemplo: si se utiliza el método de eliminación de Gauss [2-4]), o b) la numeración de los elementos (ejemplo: si se utiliza el método frontal [2]). Conviene, por tanto, cuidar la topología de la malla y adecuarla en lo posible al método de solución de ecuaciones utilizado. Para ello puede hacerse uso de técnicas especiales de optimización de la numeración de nodos y/o elementos [2], [4]. En el Apartado 9 se volverá a tratar este tema.

Etapa 3: Entrada de datos (Subrutina DATOS)

Esta etapa consiste en la lectura por el ordenador de los datos generados en la discretización. Dicha lectura se efectúa en la subrutina **DATOS** detallada en el Apartado 6.

Etapa 4: Cálculo de la matriz de rigidez de los elementos (Subrutina RIGIMAT)

En la etapa siguiente se calculan las matrices de rigidez $\mathbf{K}^{(e)}$ de cada uno de los elemento de la malla. Dicho cálculo se efectúa en la subrutina **RIGIMAT** y su mayor o menor complejidad depende del tipo de elemento utilizado. En el caso de tener un problema transitorio se calculará también la correspondiente matriz de masa, así como la aportación a la matriz de rigidez por condiciones de contorno generadas por convección-radiación, en la subrutina **CONVECC**.

Etapa 5: Cálculo del vector de temperaturas nodales (Subrutina FUERZAS)

La siguiente etapa es el cálculo del vector de temperaturas nodales equivalentes $\mathbf{f}^{(e)}$ para cada elemento y se efectúa en la subrutina **FUERZAS**. De nuevo su mayor o menor complejidad depende del elemento utilizado y también de las cargas exteriores consideradas. La descripción de **FUERZAS** se incluyen en el Apartado 8. En caso de solucionar un problema transitorio, se considera las aportaciones necesarias de las condiciones fijadas inicialmente, y de la matriz de rigidez por el esquema de integración en el tiempo, que en cada incremento de tiempo Δt se ira actualizando.

Etapa 6: Solución del sistema de ecuaciones global (Subrutina SOLUCION)

Conocidas las matrices de rigidez y los vectores de temperaturas nodales de cada elemento la etapa siguiente es el ensamblaje de dichas matrices y vectores en la ecuación

de equilibrio global $Ka = f$, y su solución para obtener las temperaturas nodales a . Este proceso se efectúa en la subrutina SOLUCION y para el mismo puede utilizarse toda una

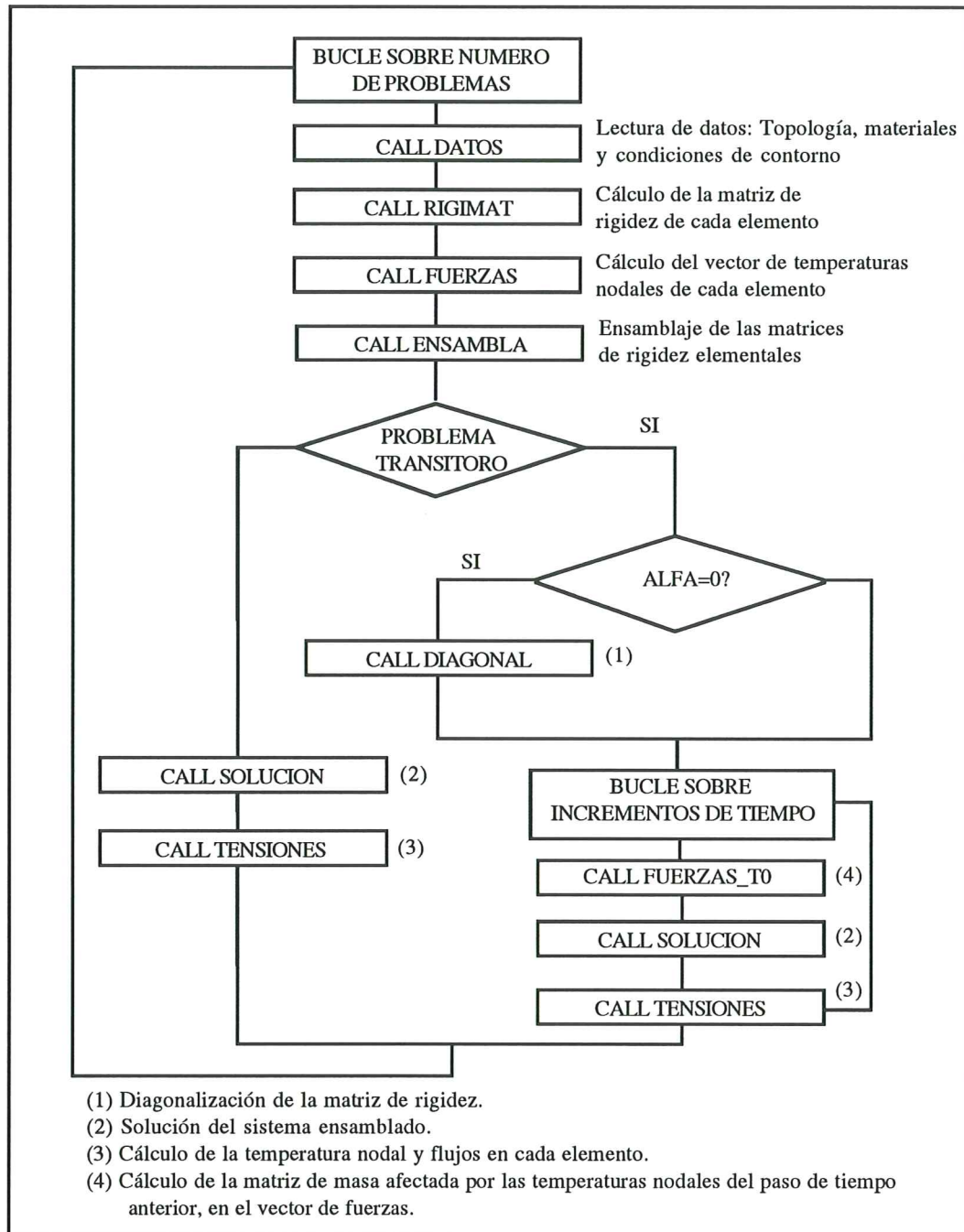


Figura 4 Diagrama de flujo del programa CALTEP.

variedad de técnicas de cálculo numérico perfectamente desarrolladas. En el Apartado 9 se hace referencia a algunas de ellas al tratar con más detalle el contenido de esta subrutina.

Etapa 7: Cálculo de gradientes (Subrutina TENSIONES)

La etapa final consiste en calcular los flujos o gradientes de la temperatura en los diferentes elementos a partir de los valores nodales. Dicho cálculo se lleva a cabo en la subrutina **TENSIONES** que se describe en el Apartado 10. Esta etapa de *postproceso* va también asociada en la práctica a la representación gráfica de los resultados del cálculo (temperaturas y flujos). En caso de tener un problema transitorio, se harán los cambios necesarios sobre los vectores de temperaturas y se cerrará un ciclo de tiempo para continuar iterativamente, hasta llegar a la convergencia, o bien, al número de iteraciones previstas. La programación del postproceso no es un problema trivial, fundamentalmente en problemas tridimensionales, siendo necesario un conocimiento profundo de técnicas de dibujo por ordenador. No obstante, generalmente puede hacerse uso de programas comerciales "*ad hoc*" y el problema se reduce a compatibilizar los formatos de resultados del cálculo con los necesarios para su representación gráfica [5-9].

La versión 1/92 de **CALTEP** para micro ordenadores incluye software para representación gráfica de datos y resultados compatibles con las subrutinas gráficas de MICROSOFT 5.0.y Finder 6.0.2 o menores para Macintosh.

5.2 Selección de los nombres de las variables

Una norma elemental de buena programación es mantener un criterio uniforme para escoger el nombre de las variables del programa [2]. Salvo contadas excepciones, en **CALTEP** todas las variables tienen cinco letras y además se ha tratado de que el nombre de cada una esté lo más relacionado posible con su función en el programa. Por otro lado, todas las variables que empiezan con la letra N indican "número de", si empiezan con M indican "número máximo de" y si lo hacen con I, J, K, indican un valor determinado de la variable. Así, por ejemplo, NELEM es el número de elementos de la malla, MELEM es el número máximo de elementos que puede analizar el programa, IELEM es un elemento determinado de los NELEM, etc.

Estas normas pueden resultar algo tediosas a los que se inician en la programación del método de los elementos finitos. No obstante, su estricto cumplimiento es de gran utilidad, tanto para el estudio y utilización de un programa determinado, como para su posterior modificación por personas ajenas a su desarrollo inicial (e incluso para el propio autor).

En el Apéndice I se presenta una relación de las principales variables del programa juntamente con la explicación de su significado.

5.3 Transmisión de información entre subrutinas

En *CALTEP* se utilizan bloques COMMON para transmitir toda la información necesaria (escalar y vectorial) entre subrutinas. Dichos bloques son iguales en *todas* las subrutinas, y sus vectores se han dimensionado en función del problema de mayor tamaño que puede analizarse con el programa. Se ha escogido esta alternativa frente a otras opciones posibles (dimensionamiento dinámico, transmisión por argumentos, etc.) por su valor didáctico para los no iniciados en programación.

CALTEP utiliza fundamentalmente cuatro bloques COMMON. El primero denominado DTSGRAL almacena los parámetros de control definidos en la subrutina DATOS y que son necesarios en todo el programa. El bloque COMMON DTSGRAL tiene la forma siguiente:

```
COMMON /DTSGRAL/  NTRAN, NDIME, NELEM, NEVAB, NGAUS, NGDLN,
                  NMATS, NNODE, NPNOD, NPRES, NPROB, NPROP,
                  NTENS, NTIPO, NFRON, NNFRO, NTIME, ITIME,
                  ALFAT, XLUMP
```

El segundo bloque COMMON se denomina DATA y almacena un conjunto de matrices y vectores relacionados con la geometría de la malla, las propiedades de los materiales, las condiciones de contorno y las temperaturas nodales. La forma de dicho bloque es

```
COMMON /DATA/      COORD (MPNOD, MDIME), CARGA (MELEM, NEVAB),
                  INPRE (MPRES, MGDLN), NODPR (MPRES),
                  LNODS (MELEM, MNODE), MATNU (MELEM),
                  PROPS (MMATS, MPROP), PRESC (MPRES, MGDLN),
                  CONVE (MPRES, MNODE), XTIME (MPNOD)
```

El tercer bloque COMMON se denomina CALCULO y almacena un conjunto de matrices y vectores que están específicamente relacionados con el cálculo de la matriz de rigidez y el vector de temperaturas nodales. Dicho bloque tiene la forma general siguiente:

```
COMMON/ CALCULO/  BMATZ (MTENS, MEVAB), COREL (MDIME, MNODE),
                  CORPG (MDIME, MGASP), DERIV (MDIME, MNODE),
                  DBMAT (MTENS, MEVAB), DCART (MDIME, MNODE),
                  DMATZ (MTENS, MTENS), FFORM (MNODE),
                  TENSZ (MTENS, MEVAB, MGASP)
```

El cuarto bloque COMMON se denomina GAUSSDAT y almacena los datos generales asociados a los puntos de gauss relacionados con el cálculo. Dicho bloque tiene la forma general siguiente:

COMMON/GAUSSDAT/POSGT(MGAUS*MDIME), POSPG(MGAUS),
PESGT(MGAUS), PESPG(MGAUS)

Los argumentos de las matrices y vectores de los bloques COMMON se han dimensionado en el programa de manera que sirvan para todos los problemas resolubles. Dichos valores se muestran en la Figura 5.

MDIME = 3	MNODE = 27
MELEM = 200	MPNOD = 200
MEVAB = 81	MPRES = 200
MGAUS = 14	MPROP = 10
MGDLN = 3	MTENS = 3
MMATS = 10	MTOTV = 600

Figura 5 Dimensiones de las matrices y vectores de los bloques COMMON DATA, CALCULO y GAUSSDAT.

En el Apéndice I puede encontrarse el significado de las variables de los bloques COMMON anteriores.

5.4 Listado de la subrutina principal de CALTEP

A continuación se muestra un listado de la subrutina principal de CALTEP incluyendo los bloques COMMON.

```

1.      PROGRAM CALTEP
2.      C*****
3.      C
4.      C*** SUBROUTINA PRINCIPAL
5.      C
6.      C*****
7.      C*****COMMON STATEMENTS
8.      IMPLICIT NONE
9.      INCLUDE 'dtsgral.f'
10.     INCLUDE 'data.f'
11.     INCLUDE 'gaussdat.f'
12.     INCLUDE 'calculo.f'
13.     INCLUDE 'solu.f'
14.     INCLUDE 'solucas.f'
15.     INTEGER*2 MPROB,IPROB,IVARI,ITIME
16.     CHARACTER*80 TITULO
17.     CHARACTER*10 INPUT,OUTPUT
18.     CHARACTER*13 AUX3
19.     DATA AUX3 /'/aux3/zarate/'/
20.     C*****COMMON STATEMENTS
21.     OPEN (2,FILE='COMAN.DAT',FORM='FORMATTED')
22.     READ(2,800)INPUT
23.     READ(2,800)OUTPUT
24.     CLOSE(2,STATUS='KEEP')
25.     800 FORMAT(A10)

```



```

26.          OPEN (5,FILE=INPUT,FORM='FORMATTED')
27.          OPEN (6,FILE=OUTPUT,FORM='FORMATTED')
28.          OPEN (1,FILE=AUX3//'TEMP1.TMP',FORM='UNFORMATTED')
29.          OPEN (3,FILE=AUX3//'TEMP3.TMP',FORM='UNFORMATTED')
30.          OPEN (4,FILE=AUX3//'TEMP4.TMP',FORM='UNFORMATTED')
31.          OPEN (8,FILE=AUX3//'TEMP8.TMP',FORM='UNFORMATTED')
32.          OPEN(10,FILE=AUX3//'TEMP10.TMP',FORM='UNFORMATTED')
33.          OPEN (29,FILE=AUX3//'CASEF.POS', FORM='UNFORMATTED')
34.          C
35.          C*** LEE NUMERO DE PROBLEMAS A ANALIZAR
36.          C
37.              READ(5,900) NPROB
38.          900  FORMAT(I5)
39.              WRITE(6,905) NPROB
40.          905  FORMAT(/10X,'NUMERO DE PROBLEMAS= ',I5)
41.          C
42.          C*** BUCLE SOBRE NUMERO DE PROBLEMAS
43.          C
44.              MPROB=INT2(NPROB)
45.              ITIME=0
46.              WRITE(29) MPROB
47.              DO IPROB=1,NPROB
48.                  REWIND 1
49.                  REWIND 3
50.                  REWIND 4
51.                  WRITE(6,910) IPROB
52.          910  FORMAT(////,6X,'PROBLEMA NO.',I3,////)
53.                  READ(5,915) TITULO
54.          915  FORMAT(A80)
55.                  WRITE(6,920) TITULO
56.          920  FORMAT(A80,////)
57.                  CALL DATOS
58.                  CALL GAUSSQ(NDIME,NNODE,NGAUS)
59.                  CALL RIGIMAT
60.                  CALL FUERZAS
61.                  CALL ENSAMBLA
62.                  IF (NTRAN.EQ.0) THEN
63.                      CALL REDUCE
64.                      CALL SUSTITUIR
65.                      CALL TENSIONES
66.                      CALL SUAV1
67.                      CALL FEMV(ITIME,TITULO)
68.                  ELSE
69.                      IF (ALFAT.EQ.0) CALL DIAGONAL
70.                      DO ITIME=1,NTIME
71.                  C
72.          C***          APLICA LAS CONDICIONES DE TIEMPO T-1
73.                  C
74.                      CALL FUERZAS_T0
75.                  C
76.          C***          RESUELVE POR ELIMINACION GAUSSIANA
77.                  C
78.                      IF (ITIME.EQ.1) THEN
79.                          CALL REDUCE
80.                      ELSE
81.                          CALL REDUCE1
82.                      ENDIF
83.                      CALL SUSTITUIR
84.                  C
85.          C***          CALCULA LAS TENSIONES EN LOS ELEMENTOS
86.                  C
87.                      CALL TENSIONES
88.                  C
89.          C***          SUAVIZADO DE TENSIONES
90.                  C
91.                      CALL SUAV1

```

```

92.          CALL FEMV(ITIME,TITULO)
93.          C
94.          C***      EL TIEMPO T SE CONVIERTE EN T-1
95.          C
96.          DO IVARI=1,NGDLN*NPNOD
97.             XTIME(IVARI)=DESEPL(IVARI)
98.          ENDDO
99.          C
100.         ENDDO
101.        ENDIF
102.       ENDDO
103.       CLOSE(1,STATUS='DELETE')
104.       CLOSE(3,STATUS='DELETE')
105.       CLOSE(4,STATUS='DELETE')
106.       CLOSE(8,STATUS='DELETE')
107.       CLOSE(10,STATUS='DELETE')
108.       CLOSE(29,STATUS='KEEP')
109.       CLOSE(5,STATUS='KEEP')
110.       CLOSE(6,STATUS='KEEP')
111.       STOP
112.       END

```

6 DESCRIPCION DE LA SUBROUTINA DATOS

Recordemos que en la subrutina **DATOS** se lee toda la información relacionada con la geometría de la malla y las propiedades de los materiales. **DATOS** está organizada de manera que sirva para cualquier problema térmico analizado con elementos uni, bi o tridimensionales. Seguidamente se describen las partes fundamentales de esta subrutina.

6.1 Parámetros de control

Dado que **CALTEP** se ha escrito para que pueda ser utilizable en más de un tipo de problemas, es esencial definir al principio una serie de parámetros de control que establezcan las características propias de cada problema. Dichos parámetros son los siguientes:

- * Número total de puntos nodales en la malla (NPNOD)
- * Número total de elementos en la malla (NELEM)
- * Número total de nodos con movimientos prescritos (NPRES)
- * Indicador de tipo de problema (NTRAN=0 estacionario, =1 transitorio.sin diagonalización,=2 transitorio con diagonalización por suma de filas, =3 transitorio.con diagonalizacion por masa equivalente)
- * Número de nodos por elemento (NNODE)
- * Número de materiales diferentes en la estructura (NMATS)
- * Orden de la cuadratura de Gauss utilizada (NGAUS)
- * Número de coordenadas necesarias para definir un nodo (NDIME)

- * Indicador de escritura (IWRITE= 0 deshabilitada, =1 habilitada.)

Recordemos que cuando uno de dichos parámetros comienza por la letra M su valor es el *máximo* que admite el programa.

6.2 Datos geométricos

La geometría de la malla puede definirse a partir de las conexiones nodales de cada elemento y las coordenadas de los nodos.

a) *Definición de las conexiones nodales de los elementos*

La geometría de cada elemento se define listando de manera sistemática sus nodos. Como en principio cada elemento puede tener diferentes propiedades del material, es conveniente asignar un número de identificación del material a cada elemento. Por consiguiente, los datos a leer para cada elemento son:

- * Número del elemento (NUMEL)
- * Número del tipo de material del elemento (MATNU (NUMEL))
- * Lista de los nodos del elemento (LNODS (NUMEL,INODE),
INODE =1, NNODE)

La definición de los nodos del elemento debe seguir siempre un mismo orden, siguiendo una secuencia antihoraria que comience por un nodo esquina cualquiera.

b) *Definición de las coordenadas de los nodos*

Las coordenadas de los nodos se definen siempre con relación a un sistema cartesiano global. La información a leer para cada nodo es la siguiente:

- * Número del nodo (IPNOD)
- * Coordenadas cartesianas del nodo (COORD (IPNOD,IDIME),
IDIME =1, NDIME)

6.3 Condiciones de nodos prescritos

Las condiciones de contorno en los NPRES nodos con movimientos prescritos se leen de acuerdo con la secuencia siguiente:

- * Número del nodo prescrito (NODPR (IPRES))
- * Indicador de los movimientos nodales prescritos (INPRE (IPRES,IGDLN), IGDLN=1, NGDLN)

- * Valores de los movimientos nodales prescritos (PRESC (IPRES,IGDLN), IGDLN=1, NGDLN)
- * Si INPRE (IPRES,IGDLN)=1 indica que el grado de libertad IGDLN del nodo prescrito IPRES está coaccionado. Por el contrario INPRE=0 denota que dicho grado de libertad está libre. Esto permite coaccionar de manera selectiva los movimientos de cada nodo.

6.4 Propiedades del material

Deben leerse los parámetros del material necesarios para formar la matriz constitutiva **D**. Así, para cada material diferente la información a suministrar es la siguiente:

- * Número del tipo de material (NUMAT)
- * Propiedades del tipo de material en cuestión (PROPS (NUMAT,IPROP), IPROP=1, NPROP)

El orden de lectura de las propiedades del material es el siguiente: Conductividad térmica (tantos valores como dimensiones cartesianas tenga el problema), coeficiente de convección/radiación en fronteras del elemento, valor del calor de generación interna del elemento y en el caso de resolver problemas temporales, se deberá incluir el valor de la densidad del material.

6.5 Condiciones Iniciales de temperatura

En el caso de realizar un análisis transitorio es necesario proporcionar el numero de iteraciones que se deseen, el incremento de tiempo a utilizar y el parámetro α de integración en el tiempo; debido al esquema de iteración temporal implementado se recomienda utilizar incrementos de tiempo adecuados, para no provocar oscilaciones en la integración temporal. La información a suministrar es la siguiente:

- * Número del iteraciones (NTIME)
- * Incremento de tiempo (DTIME)
- * Punto de integración en el intervalo Δt (ALFAT)

A continuación se darán el número de nodos con valores prescritos de las temperaturas iniciales nodales, pudiendo simplificar su captura si todos los nodos de la malla tienen un mismo valor, al introducir (NPNOD) en (NNCIT) y a continuación el valor inicial en (CITIM). Si existen nodos con temperaturas iniciales variables se dará en (NNCIT) el numero de estos y en (CITIM) el valor correspondiente, por lo que el programa asume que los nodos restantes inician con una temperatura de cero.

6.6 Condiciones de flujo de Convección/Radiación en la frontera

Cuando se tenga el caso de elementos que presenten radiación o convección en alguno de sus lados, se deberá proporcionar el numero de elementos existentes en (NNFRON) y el numero de nodos que tienen por lados en (NNFRO) (En el caso de no existir deberá darse el valor de cero a ambas variables). Si fue considerada la situación de tener convección o radiación, la información a suministrar es la siguiente:

Número del elemento con la condición establecida (CONVE(IFRON, 1))

Nodos del lado considerado en numeración global

(CONVE(IFRON,1+INFRO))

6.7 Subrutina GAUSS

La función de esta subrutina es definir las coordenadas y los pesos de la cuadratura de Gauss seleccionada para las integrales del elemento. En elementos unidimensionales, cuadriláteros y hexagonales el orden de la cuadratura utilizada lo define la variable NGAUS, siendo entonces la cuadratura de orden: NGAUS en una dimensión, NGAUS X NGAUS en dos dimensiones y NGAUS X NGAUS X NGAUS en tres dimensiones. Las coordenadas y los pesos de los puntos de integración se almacenan en las variables POSGP(•) y PESPG(•).

En elementos triangulares NGAUS define asimismo el orden de la cuadratura, estando en este caso las coordenadas y los pesos de los puntos de integración asignados a las variables POSGT(•) y PESGT(•).

La subrutina Gauss de la versión de *CALTEP* que se lista en el Apéndice II permite la utilización de cuadraturas con NGAUS=1, 2 y 3 para elementos unidimensionales y cuadriláteros y hexagonales, y NGAUS=1, 3 y 7 para elementos triangulares.

6.8 Preparación automática de datos

En muchos casos el mayor esfuerzo en el análisis de una estructura por elementos finitos se invierte en la preparación de los datos. Por este motivo, es conveniente disponer de medios informáticos auxiliares que permitan automatizar gran parte de las operaciones de la entrada de datos. Ejemplos de esto son las pantallas gráficas, plotters, digitalizadores y copadoras gráficas. Asimismo, es muy útil disponer de programas para generación automática de mallas en una, dos y tres dimensiones y para su representación gráfica [5-7].

Con estos medios se pueden conseguir importantes ahorros en el tiempo de preparación de datos, así como en el de detección de los errores que inevitablemente se producen.

Los programas de generación y dibujo de la malla se incluirían en una hipotética subrutina **GENER** que podría ser llamada por **DATOS**. La descripción de dicha subrutina cae fuera del alcance de este capítulo. Los lectores interesados en este tema pueden encontrar abundante información en las referencias [2], [5-14].

6.9 Subrutina de control de datos

Una vez leídos los datos geométricos y del material de la malla es importante realizar una serie de comprobaciones básicas que garanticen mínimamente la ausencia de errores antes de comenzar el cálculo de las matrices y vectores de cada elemento. Dichas comprobaciones pueden ser tan simples como comprobar que no se han asignado valores absurdos a los parámetros de control; que dichos valores son compatibles con las dimensiones de los vectores de los bloques **COMMON**; que todos los nodos aparecen alguna vez en algún elemento; que no hay dos nodos con el mismo número o mismas coordenadas, etc. Asimismo, puede comprobarse que las dimensiones características de la matriz de rigidez global (ancho de banda [2], ancho de frente [2], perfil [15, 16], etc.) no son superiores a los límites establecidos de acuerdo con el método de solución del sistema de ecuaciones utilizado. Dichas comprobaciones podrían efectuarse en una subrutina auxiliar **COMPROB** que sería llamada por **DATOS** una vez leídos los datos fundamentales de la geometría y propiedades mecánicas de la malla.

Por razones de brevedad no se incluyen aquí subrutinas de control de datos. Dichas subrutinas son fáciles de escribir, aunque siempre deben adecuarse al sistema de generación de datos disponible (manual, automático, etc.). En [2] se pueden encontrar ejemplos de subrutinas de control de datos compatibles con la organización del programa aquí presentado.

7.1 MATRIZ DE RIGIDEZ: SUBROUTINA RIGIMAT

Describiremos brevemente las operaciones para el cálculo de la matriz de rigidez para el caso estacionario y transitorio, incluidos en **RIGIMAT**.

7.1.1 Caso Estacionario

En todos los casos utilizaremos una formulación *isoparamétrica* [4]. Recordemos la expresión general de la matriz de rigidez de un elemento plano [4]:

$$\mathbf{K}_{ij}^{(e)} = \int_{-1}^1 \int_{-1}^1 \mathbf{B}_i^T \mathbf{D} \mathbf{B}_j \left| \mathbf{J}^{(e)} \right| \delta \xi \delta \eta \quad (\text{xxix})$$

En donde $\mathbf{J}^{(e)}$ es la clásica matriz Jacobiano de la transformación de coordenadas cartesianas (x,y) a naturales (ξ,η) .

Los elementos de $\mathbf{K}_{ij}^{(e)}$ se calculan numéricamente. Así, denominando

$$\mathbf{T}_{ij}^{(e)}(\xi, \eta) = \mathbf{B}_i^T \mathbf{D} \mathbf{B}_j \left| \mathbf{J}^{(e)} \right| \quad (\text{xxx})$$

la integración numérica para un *elemento cuadrilátero* con una cuadratura de Gauss de orden $m \times m$ se escribe como

$$\mathbf{K}_{ij}^{(e)} = \sum_{p=1}^m \sum_{q=1}^m \mathbf{T}_{ij}^{(e)}(\xi_p, \eta_q) W_p W_q \quad (\text{xxxii})$$

donde ξ_p y η_q son las coordenadas naturales de los puntos de integración y W_p y W_q los correspondientes pesos [4].

Para un elemento *triangular* con una cuadratura de orden m se tiene

$$\mathbf{K}_{ij}^{(e)} = \sum_{p=1}^m \mathbf{T}_{ij}^{(e)}(\alpha_p, \beta_q) W_p \quad (\text{xxxiii})$$

En el programa se incluyen los elementos lineales isoparamétricos de dos y tres nodos, los elementos cuadriláteros isoparamétricos serendípticos de cuatro y ocho nodos y el lagrangiano de nueve nodos, y los triangulares de tres y seis nodos así como el elemento hexagonal isoparamétrico serendípito de veinte nodos (Figura 6).

En todos los elementos cuadriláteros se recomienda una cuadratura 2×2 . Por otra parte, los elementos triangulares lineal y cuadrático precisan cuadraturas de uno y tres puntos, respectivamente [4]. Mientras que para el elemento hexagonal isoparamétrico serendípito de veinte nodos cuya matriz de rigidez se presenta a continuación, se recomienda evaluar con una cuadratura de orden $2 \times 2 \times 2$ por

$$\mathbf{K}_{ij}^{(e)} = \sum_{p=1}^2 \sum_{q=1}^2 \sum_{r=1}^2 \mathbf{T}_{ij}^{(e)}(\xi_p, \eta_q, \zeta_r) W_p W_q W_r \quad (\text{xxxiiii})$$

En el caso de los elementos lineales, una cuadratura de orden 2 es suficiente.

7.1.2 Caso Transitorio

Al realizar la integración en el tiempo es necesario incluir en la matriz de rigidez elemental, la aportación de una matriz de masa elemental que se evalúa de acuerdo con:

$$\mathbf{M}_{ij}^{(e)} = \int_{-1}^1 \int_{-1}^1 \rho c \mathbf{N}_i^T \mathbf{N}_j \left| \mathbf{J}^{(e)} \right| \delta \xi \delta \eta \quad (\text{xxxv})$$

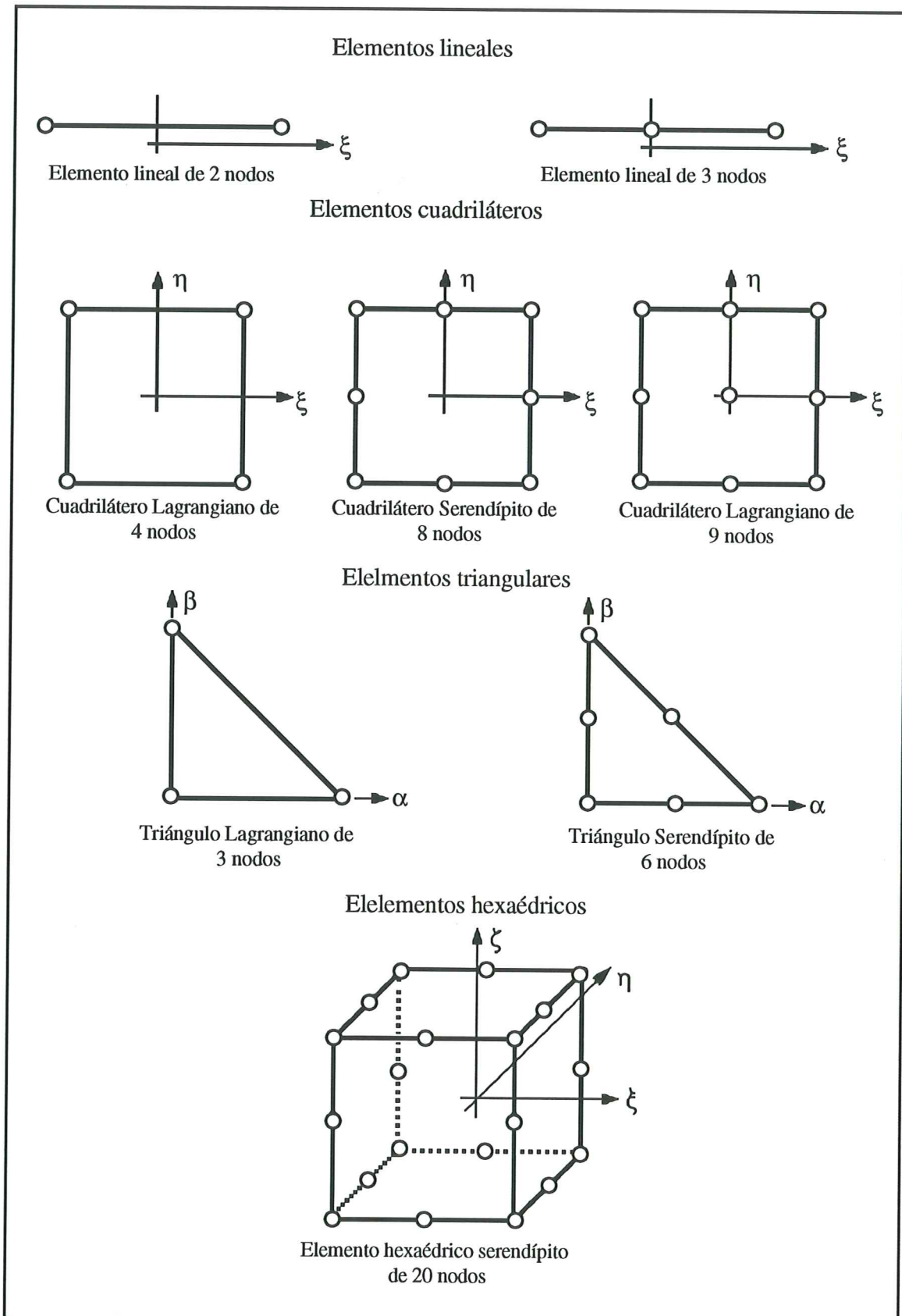


Figura 6 Elementos incluidos en el programa CALTEP

como se encontró en (xviii) y de ahí se concluye que el sistema a resolver para un paso de tiempo es el propuesto en (xxii), por lo que la matriz de rigidez encontrada en el apartado 7.1.1, se incrementa termino a termino con la encontrada en la ecuación (xxxiv), siendo afectada la matriz $\mathbf{K}^{(e)}$ por el coeficiente α de integración en el tiempo.. Las ventajas de este esquema de integración temporal, además de su sencillez de implementación es que presenta una estabilidad incondicional ante los valores de α mayores a 0.5 y para α igual a cero no es necesario invertir la matriz de rigidez.

Es obvio que en el caso transitorio, también se verán afectados los términos de carga según se muestra en la ecuación (xxii).

En la Figura 7 se presenta el diagrama de flujo de **RIGIMAT**. Obsérvese que se han supuesto propiedades del material constantes sobre el elemento y el cálculo de la matriz constitutiva **D** se efectúa antes del bucle sobre los puntos de integración. En el caso de propiedades variables bastaría con incluir la evaluación de **D** dentro de dicho bucle.

El contenido de las diferentes subrutinas que intervienen en **RIGIMAT** puede estudiarse con detalle en el listado que se presenta en el Apéndice II.

7.2 MATRIZ DE RIGIDEZ: SUBROUTINA CONVECC

La aportación a la matriz de rigidez por efectos de convección y radiación, expresado por el tercer elemento del primer término de la ecuación (xi), debe de ser considerada en todos los elementos de contorno donde esté presente dicho fenómeno.

En el caso de un elemento bidimensional isoparamétrico, la expresión general es:

$$\mathbf{K}_{ij}^{(\alpha e)} = \int_{-1}^1 \mathbf{N}_i^T \mathbf{N}_j \alpha |\mathbf{J}^{(e)}| d\xi \quad (\text{xxxv})$$

Debemos observar que la integración es unidimensional, ya que la pérdida de calor por convección y radiación se produce sobre los lados del elemento y no sobre todo el elemento en si, por lo que la matriz jacobiano de transformación de coordenadas deberá corresponder a la del lado en cuestión, siendo la integración sobre esa línea; por lo tanto es importante resaltar que la matriz $\mathbf{K}^{(\alpha e)}$ contendrá elementos nulos con excepción de aquellos $\mathbf{K}_{ij}^{(\alpha e)}$ que correspondan a los del lado con la condición de contorno impuesta. Para mayor claridad del concepto, se puede observar la figura 8.

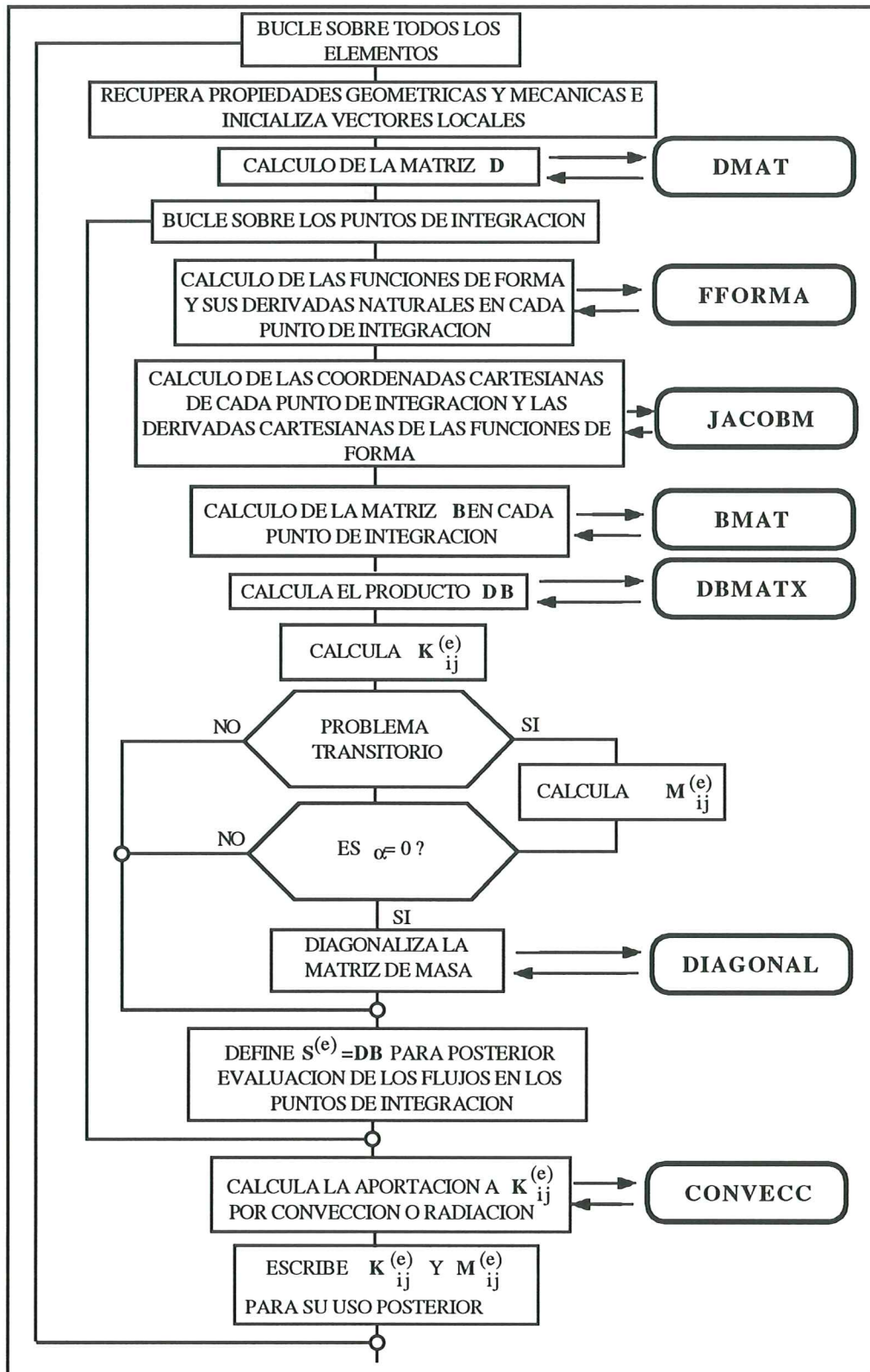


Figura 7 Diagrama de flujo de la subrutina RIGIMAT

Análogamente, para el caso tridimensional, la frontera de convección y radiación corresponderá a una superficie, por lo que la expresión es:

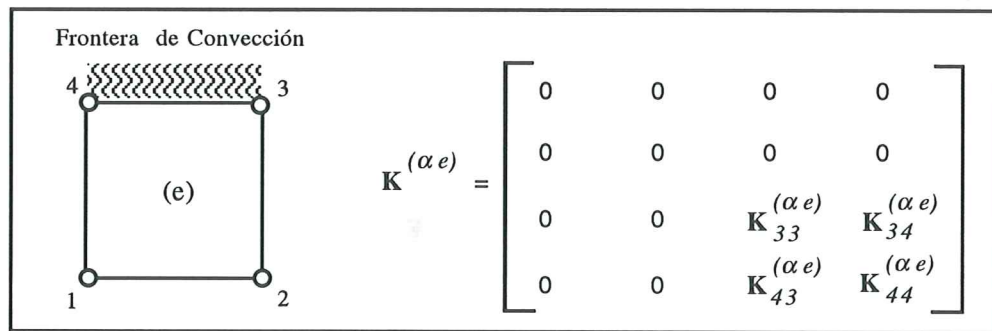


Figura 8 Aportación correspondiente por Convección y Radiación

$$K_{ij}^{(\alpha e)} = \int_{-1}^1 \int_{-1}^1 N_i^T N_j \alpha |J^{(e)}| \partial \xi \partial \eta \tag{xxxvi}$$

correspondiendo a una integral de superficie.

En el caso unidimensional, se considera que existe el fenómeno sobre la superficie perimetral del elemento de línea y no en la sección transversal de este, siendo la expresión idéntica a la ecuación (xxxv) y en este caso la matriz $K^{(\alpha e)}$ será llena.

Al tener de esta manera, en todos los casos vistos, una matriz $K^{(\alpha e)}$ de igual dimensión que la matriz de rigidez, la matriz de rigidez del elemento será la adición de ambas.

$$K_{ij}^{(e)} = K_{ij}^{(e)} + K_{ij}^{(\alpha e)} \tag{xxxvii}$$

8 SUBROUTINA FUERZAS

8.1 Consideraciones generales.

Presentaremos seguidamente la subrutina de cálculo de temperaturas nodales equivalentes para los diferentes elementos considerados. Se tendrán en cuenta los siguientes tipos de acciones exteriores:

- * *Flujos puntuales nodales* (IPUNT \neq 0).
- * *Generación Interna* (IPESO \neq 0).
- * *Flujo y Temperatura Externa sobre el elemento* (IDIST \neq 0).

La actuación de los distintos tipos de temperaturas se controla asignando un valor diferente de cero a los parámetros de control IPUNT, IPESO e ILADO.

En la Figura 9 se presenta el diagrama de flujo de la subrutina FUERZAS. En los apartados siguientes se detallan los módulos de temperaturas puntuales, Generación Interna y Flujo y Temperatura Externa sobre el elemento.

8.2 flujos puntuales nodales

Para mayor sencillez consideraremos que los flujos puntuales actúan directamente sobre un nodo. El vector de flujos nodales es en este caso simplemente

$$\mathbf{f}_i = \mathbf{p}_i \quad (\text{xxxviii})$$

siendo \mathbf{p}_i el vector de flujos puntuales sobre el nodo de numeración global i . Aunque este termino no se incluye explícitamente en la ecuación (xx) el vector \mathbf{f} corresponde directamente al flujo nodal equivalente por lo que la aplicación de una flujo sobre un nodo en particular queda representada directamente por en vector \mathbf{f} , particularizado en la ecuación (xvii).

8.3 Generación interna

La generación interna de calor equivale a una temperatura "música" actuando por unidad de superficie/volumen. Así pues, las temperaturas por generación interna para los diferentes elementos de *CALTEP* se obtienen por [4], que de acuerdo con la formulación descrita, y aplicando las transformaciones isoparamétricas correspondientes, se tiene que el primer termino del segundo miembro de la ecuación (xx) se expresa como:

Elementos unidimensionales

$$\mathbf{f}_i^{(e)} = \int_{l^{(e)}} \mathbf{N}_i^T \rho r \partial l^{(e)} = \sum_{p=1}^{n_p} \left(\mathbf{N}_i^T \rho r |J^{(e)}| \right)_p W_p \quad (\text{xxxix})$$

Elementos bidimensionales

$$\mathbf{f}_i^{(e)} = \int_A \mathbf{N}_i^T \rho r \partial A^{(e)} = \sum_{p=1}^{n_p} \sum_{q=1}^{n_q} \left(\mathbf{N}_i^T \rho r |J^{(e)}| \right)_{p,q} W_p W_q \quad (\text{xl})$$

Elementos de sólido tridimensional

$$\mathbf{f}_i^{(e)} = \int \int \int_V \mathbf{N}_i^T \rho r \partial V^{(e)} = \sum_{p=1}^{n_p} \sum_{q=1}^{n_q} \sum_{r=1}^{n_r} \left(\mathbf{N}_i^T \rho r |J^{(e)}| \right)_{p,q,r} W_p W_q W_r \quad (\text{xli})$$

La cuadratura para el calculo de estos términos, suele ser la misma que para el de la matriz de rigidez, aún a costa de introducir un cierto error en el cálculo. No obstante, este

error es de poca importancia y suele compensarse con los errores en la evaluación de la matriz de rigidez.

8.4 Temperatura ambiental y flujo Exterior repartido sobre un lado

Sobre los lados de los elementos limítrofes de la geometría, es común que exista una temperatura externa diferente que coaccione el comportamiento del flujo de temperaturas en el interior de la malla, o bien que exista un flujo de calor entrante o saliente, normal a la superficie del lado del elemento; en análisis bidimensional pueden actuar un flujo repartido por *unidad de longitud* en direcciones normal y tangencial, (aunque en realidad el flujo tangencial no afecta la solución de la ecuación diferencial y se desprecia). Estas cargas de calor no tienen porqué ser uniformes y su intensidad puede variar a lo largo del lado. Dicha variación se define por los valores de las temperaturas en los nodos del lado cargado.

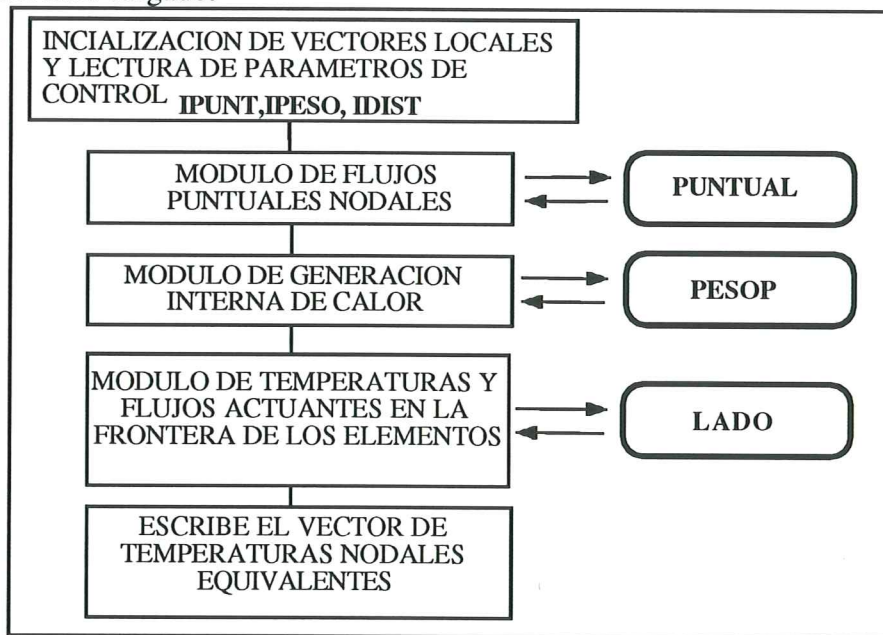


Figura 9 Diagrama de flujo de la subrutina FUERZAS para el cálculo de las temperaturas nodales equivalentes

Para ser coherente con el orden de numeración de las conexiones nodales, los nodos del lado cargado deben listarse en una secuencia antihoraria.

Un flujo normal a un lado se considerará positiva si va dirigida *hacia el interior* del elemento. Esta definición es necesaria para evitar confusiones cuando los flujos distribuidos actúan sobre lados comunes a dos elementos, siendo importante advertir que en dicho caso la carga generada sólo debe asignarse a un solo elemento.

El vector de las temperaturas equivalentes nodales se obtiene como se explica con detalle en la referencia [4]. La expresión final de dicho vector es la combinación de los elementos segundo y tercero de la ecuación (xx) que expresado para el caso unidimensional con la formulación isoparamétrica y la integración numérica se describe como:

$$\mathbf{f}_i^{(e)} = \oint_{\Gamma_q} N_i^T [\alpha \phi_{ext} - \bar{q}] \partial \Gamma_q^{(e)} = \sum_{p=1}^{n_p} \left(N_i^T [\alpha \phi_{ext} - \bar{q}] \mathbf{J}^{(e)} \right) W_p \quad (\text{xlii})$$

que de manera similar puede ser expresada para los dominios bi y tridimensionales.

9 SOLUCION DEL SISTEMA DE ECUACIONES. SUBROUTINA SOLUCION

Existen muchas técnicas para resolver el sistema de ecuaciones $\mathbf{Ka} = \mathbf{f}$ resultante del ensamblaje de las ecuaciones de equilibrio de los diferentes elementos. Entre los procedimientos más populares podemos citar los:

Métodos directos, como

- * Eliminación Gaussiana [2], [3], [11], [17]
- * Método frontal [2], [18]
- * Reducción de Choleski (o Choleski modificado) [3, 17]
- * Reducción de Crout [3, 17]
- * Método del perfil [15, 16], etc.

Métodos iterativos, como

- * Método iterativo de Gauss-Seidel [3, 17]
- * Métodos de gradiente conjugado [3, 17]
- * Métodos de relajación [3, 17]

La descripción detallada de estos métodos puede encontrarse en la mayoría de libros de cálculo numérico, y en particular en las referencias citadas. Aquí emplearemos el método de eliminación Gaussiana, por ser quizás el más sencillo y fácil de implementar en un programa de elementos finitos. El método de eliminación Gaussiana se basa en la reducción del sistema de ecuaciones $\mathbf{Ka} = \mathbf{f}$ a la siguiente forma triangular

$$\begin{aligned} K_{1,1}a_1 + K_{1,2}a_2 + K_{1,3}a_3 + \dots + K_{1,n-1}a_{n-1} + K_{1,n}a_n &= f_1 \\ K'_{2,2}a_2 + K'_{2,3}a_3 + \dots + K'_{2,n-1}a_{n-1} + K'_{2,n}a_n &= f'_2 \\ K'_{3,3}a_3 + \dots + K'_{3,n-1}a_{n-1} + K'_{3,n}a_n &= f'_3 \\ &\dots \\ K'_{n-1,n-1}a_{n-1} + K'_{n-1,n}a_n &= f'_{n-1} \\ K'_{n,n}a_n &= f'_n \end{aligned}$$

(xliii)

donde las primas indican que los coeficientes de la matriz de rigidez global y los términos de temperatura se han modificado durante la etapa de reducción.

El sistema (xliii) permite calcular sistemáticamente todas las incógnitas resolviendo las ecuaciones *reducidas* en orden inverso desde la última hasta la primera, ya que cada nueva ecuación sólo introduce una nueva incógnita. Así, la última ecuación proporciona directamente el valor de a_n ; sustituyendo dicho valor en la ecuación anterior puede obtenerse el valor de a_{n-1} y así sucesivamente. Esta etapa de la solución se denomina *sustitución hacia atrás*.

Por consiguiente, las tres etapas características del método de eliminación Gaussiana son las siguientes:

1) *Etapa de ensamblaje*. Consiste en ensamblar la matriz de rigidez global K y el vector de temperaturas nodales f a partir de las contribuciones de los diferentes elementos de la malla.

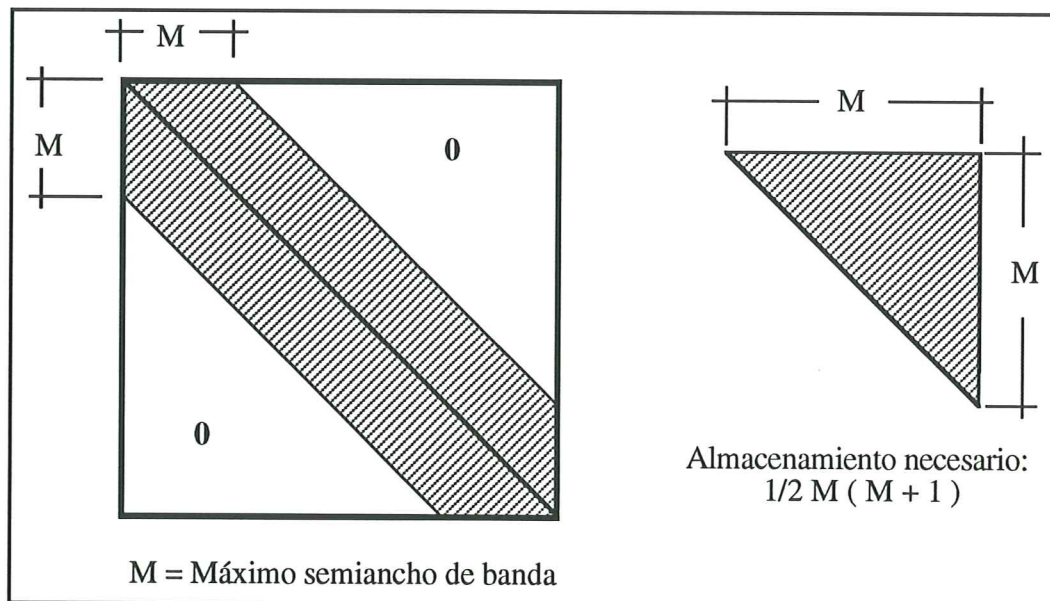


Figura 10 Matriz de rigidez en banda.

2) *Etapa de reducción*. Consiste en reducir el sistema de ecuaciones original $Ka = f$ a la forma (xliv). Esto se efectúa empleando la ecuación i -ésima para eliminar la variable a_i de todas las ecuaciones inferiores, es decir, de la ecuación $i + 1$ a la n . Formalmente esto puede efectuarse sustrayendo de la ecuación r -ésima ($i < r \leq n$) la ecuación i -ésima multiplicada por $(i)K_{ri} / (i)K_{ii}$, donde el índice i indica que dichos coeficientes se han modificado $i - 1$ veces antes de la eliminación de a_i . Por ejemplo, la primera ecuación se utiliza para eliminar a_1 de las ecuaciones 2 a n , como :

$$K_{1,1}a_1 + K_{1,2}a_2 + K_{1,3}a_3 + \dots + K_{1,n}a_n = f_1$$

$$0a_1 + \left(K_{2,2} - K_{1,2} \frac{K_{2,1}}{K_{1,1}} \right) a_2 + \left(K_{2,3} - K_{1,3} \frac{K_{2,1}}{K_{1,1}} \right) a_3 + \dots + \left(K_{2,n} - K_{1,n} \frac{K_{2,1}}{K_{1,1}} \right) a_n = f_2 - f_1 \frac{K_{2,1}}{K_{1,1}} = f'_2$$

$$\dots \dots \dots$$

$$0a_1 + \left(K_{n,2} - K_{1,2} \frac{K_{n,1}}{K_{1,1}} \right) a_2 + \left(K_{n,3} - K_{1,3} \frac{K_{n,1}}{K_{1,1}} \right) a_3 + \dots + \left(K_{n,n} - K_{1,n} \frac{K_{n,1}}{K_{1,1}} \right) a_n = f_n - f_1 \frac{K_{n,1}}{K_{1,1}} = f'_n \quad (\text{xliv})$$

A continuación se utiliza la ecuación 2 para eliminar a_2 de todas las ecuaciones y así sucesivamente. Adviértase que se mantiene la simetría en el sistema de ecuaciones modificado.

Si un movimiento está prescrito, por ejemplo $a_2 = \bar{a}_2$, la incógnita pasa a ser la reacción correspondiente t_2 . En este caso, la eliminación de a_2 es trivial y todo lo que hay que hacer es sustituir $a_2 = \bar{a}_2$ en todas las ecuaciones y transferir la cantidad conocida $K'_{r2} \bar{a}_2$ ($3 \leq r \leq n$) al segundo miembro de cada ecuación, como se ilustra seguidamente

$$K_{1,1}a_1 + K_{1,2} \bar{a}_2 + K_{1,3}a_3 + \dots + K_{1,n}a_n = f_1$$

$$0a_1 + K'_{2,2} \bar{a}_2 + K'_{2,3}a_3 + \dots + K'_{2,n}a_n = f'_2$$

$$0a_1 + 0a_2 + K'_{3,3}a_3 + \dots + K'_{3,n}a_n = f'_3 - K'_{3,2} \bar{a}_2$$

$$\dots \dots \dots$$

$$0a_1 + 0a_2 + K'_{n,3}a_3 + \dots + K'_{n,n}a_n = f'_n - K'_{n,2} \bar{a}_2 \quad (\text{xlv})$$

Si el movimiento está prescrito a un valor nulo es más sencillo prescindir directamente de la fila y columna correspondientes a dicho movimiento y calcular la reacción a posteriori una vez resuelto el sistema.

En el proceso de reducción puede hacerse uso de las ventajas de la *simetría* de la matriz de rigidez global. Ello permite reducir sólo los términos sobre y por arriba de la diagonal principal de \mathbf{K} , obteniéndose el resto por simetría, con el consiguiente ahorro de cálculo.

En la mayor parte de los problemas estructurales \mathbf{K} tiene forma de banda, con ceros fuera de una banda simétrica con respecto a la diagonal principal (Figura 10). Dicha forma en banda permite introducir considerables economías en el proceso de solución del sistema, ya que en cada instante sólo es necesario almacenar la matriz triangular superior de una submatriz de dimensiones $M \times M$, siendo M el semi-ancho de banda. Por consiguiente, los requisitos de almacenaje son de $1/2M(M+1)$ números. El valor de M depende de cómo se numeran los nodos en la malla. Puede demostrarse que $M=(D+1)*NDGDL$ siendo D la máxima diferencia entre los números de dos nodos pertenecientes a cualquiera de los elementos de la malla [2].

Lo anterior evidencia la importancia de escoger una numeración nodal tal que el valor de M sea mínimo. Esto en general no es sencillo, sobre todo para mallas grandes, y en estos casos suele acudir a programas que automáticamente proporcionan una numeración nodal óptima. Una alternativa es utilizar otros métodos de solución en los que el orden de numeración nodal sea irrelevante. Entre éstos se encuentra el método **FRONTAL** en el que la capacidad de almacenamiento necesaria viene condicionada por la numeración de los elementos y no de los nodos, lo que facilita la etapa de preproceso. El método **FRONTAL** es una modificación del de eliminación Gaussiana y se basa en realizar el ensamblaje de las ecuaciones y la eliminación de las variables de manera simultánea. Así, una vez ensamblada la matriz de rigidez de un nuevo elemento, se eliminan las variables nodales que no se verán afectadas por posteriores ensamblajes, con lo que el espacio asignado a los coeficientes de rigidez de dichas variables puede ser ocupado por el de nuevas variables al ensamblar las ecuaciones, de otro elemento, y así sucesivamente. Esto reduce considerablemente los requisitos de almacenaje y el número de operaciones aritméticas, a costa, sin embargo, de incrementar sensiblemente la complejidad del algoritmo de solución. Una descripción detallada del método **FRONTAL** puede encontrarse en la referencia [2].

En el Apéndice II se presenta el listado de las subrutinas **SOLUCION**, **ENSAMBLA**, **REDUCE** y **SUSTITUIR** para resolver el sistema de ecuaciones utilizando el método de eliminación Gaussiana explicado. Como nota característica se destaca que en dichas subrutinas los términos de la matriz de rigidez se almacenan en un vector siguiendo una secuencia a lo largo de la banda, prescindiéndose de los coeficientes nulos situados fuera de la misma, con la consiguiente economía de memoria.

El método de solución seguido para el problema de integración en tiempo, es esencialmente el mismo, con la salvedad de que en la parte triangular inferior de la matriz de rigidez, que el sistema de reducción gaussiana convierte en ceros, se almacenan los valores de los elementos K'_{ij}/K'_{ii} de manera que para resolver el sistema en todas las

iteraciones temporales, no será necesario resolver nuevamente la matriz de rigidez, ya que esta es constante en todo el proceso, y esta información almacenada servirá para reducir el vector de temperaturas nodales, con el consecuente ahorro de tiempo de cómputo.

10 CALCULO DE LOS FLUJOS ELEMENTALES. SUBROUTINA TENSION

Como es usual los flujos se calculan inicialmente en los puntos de integración y a partir de dichos valores se procede a su extrapolación a los nodos y al subsecuente alisado nodal si se desea [4]. Así, las tensiones en el punto de Gauss p se obtienen por

$$\mathbf{q}_p = \sum_{i=1}^n (\mathbf{DB}_i)_p \mathbf{a}_i \quad (\text{xlvi})$$

Recordemos que durante el cálculo de la matriz de rigidez en **RIGIMAT** se evalúa el producto **DB** en cada punto de Gauss, almacenándose la matriz resultante en un fichero, juntamente con las coordenadas cartesianas de los puntos de Gauss. Por consiguiente, para efectuar los productos de (xlvi) basta con leer para cada elemento la información almacenada en dicho fichero, evitándose así repetir el cálculo de las matrices **B** y **D** en cada punto de Gauss.

En el Apéndice II se presenta el listado de la subrutina **TENSIONES** donde se efectúan esas operaciones.

Se incluye en **CALTEP**, una subrutina de extrapolación de los valores de los flujos de los valores de los flujos en los puntos de Gauss a los nodos siguiendo la técnica de extrapolación y alisado global descritas en la referencia [4].

11 EJEMPLOS DE UTILIZACION DEL PROGRAMA CALTEP

A efectos meramente ilustrativos se presentan dos sencillos ejemplos de aplicación del programa *CALSEF* a los problemas siguientes [11]:

- Flujo de calor a través de una barra delgada con frontera de convección y radiación, analizada con dos elementos unidimensionales.
- Generación de calor en un dominio cuadrado, caso estacionario, transitorio con $\alpha = 1$ y transitorio con $\alpha=2/3$; analizado con una malla de 5 X 5 elementos triangulares de 3 nodos.

Para cada problema analizado se presentan los listados de los datos y de la salida de resultados. Los flujos nodales que se listan se han obtenido mediante la técnica de extrapolación y alisado global, utilizando un campo *lineal* de flujos. Esto proporciona directamente los valores alisados de las tensiones en los nodos esquina de cada elemento.

11.1 Flujo a través de una barra delgada.

Dada la geometría del problema y características del problema, su análisis es muy sencillo como se observa en la figura 11, por lo que pasaremos directamente a la codificación del fichero de entrada:

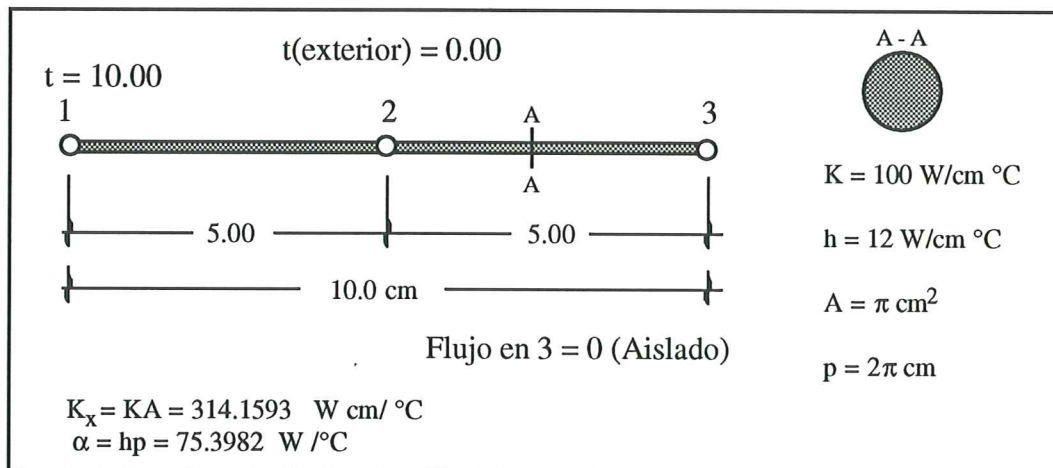


Figura 11 Geometría y discretización del problema del flujo de calor a través de una barra delgada.

LISTADO DE DATOS (BARRA DELGADA)

```

FLUJO DE CALOR EN BARRA DELGADA (LINEAL 2 NODOS)
3  2  1  0  2  1  2  1  1
1  1  1  2
2  1  2  3
1  0.00000
2  5.00000
3  10.00000
1  1  10.0
1  314.1593  75.3982  0.0
2  2
1  1  2
2  2  3
sin condiciones de carga
0  0  0

```

LISTADO DE RESULTADOS (BARRA DELGADA)

```

NUMERO DE PROBLEMAS= 1
PROBLEMA NO. 1
FLUJO DE CALOR EN BARRA DELGADA (LINEAL 2 NODOS)
NPNOB = 3 NELEM = 2 NPRES = 1 NTRAN = 0 *NTIPO = 1
NNODE = 2 *NGDLN = 1 NMATS = 1 *NPROP = 3 NGAUS = 2
NDIME = 1 *NTENS = 1 *NEVAB = 2 IWRTT = 1
ELEMENTO PROPIEDAD NUMERO DE NODOS
1 1 1 2
2 1 2 3
COORDENADAS DE PUNTOS NODALES
NODO X Y Z
1 0.000
2 5.000
3 10.000
NODOS RESTRINGIDOS Y PRESCRIPCIONES
NODO CODIGO VALORES PRESCRITOS
1 1 10.00000
PROPIEDADES DE LOS MATERIALES
NUMERO PROPIEDADES
1 K= 0.31E+03
ALPHA= 0.75E+02 ROr= 0.00E+00 ROc=
FRONTERAS CON FLUJO DE CONVECCION/RADIACION = 2
NUMERO DE NODOS POR FRONTERA = 2
ELEMENTO N O D O S
1 1 2
2 2 3
sin condiciones de carga
COND. DE CARGA PUNTUAL:..... 0
COND. DE CARGA DE GENER. INTERNA:.... 0
COND. DE CALOR DISTRIBUIDO POR LADO.. 0

```


FUERZAS TOTALES NODALES EQUIVALENTES ELEMENTALES

1	0.0000E+00	0.0000E+00
2	0.0000E+00	0.0000E+00

TEMPERATURAS

NODO	TEMPERATURA
1	0.100000E+02
2	0.607128E-06
3	0.737209E-13

REACCIONES

NODO	REACCION
1	0.188496E+04

TENSIONES

G.P. X-COORD X-FLUJ.

ELEMENTO NO.=	1
1	1.0566 -628.3185
2	3.9434 -628.3185

ELEMENTO NO.=	2
1	6.0566 0.0000
2	8.9434 0.0000

FLUJOS ALISADAS NODALES

NODO	X-FLUJ.	Y-FLUJ.	Z-FLUJ.
1	-0.62832E+03		
2	-0.31416E+03		
3	-0.38147E-04		

11.2 Generación interna de calor en un dominio cuadrado.

11.2.1 Caso estacionario

La geometría del problema en este caso nos permite analizar un cuarto del dominio, debido a la simetría que guarda con los ejes XX e YY aplicando las características mostradas en la figura 12, tanto en las condiciones de contorno como en la malla utilizada. El caso que se resuelve es el estacionario; se muestra a continuación los datos del programa

11.2.1.1 LISTADO DE DATOS (GENERACION DE CALOR EN DOMINIO CUADRADO) CASO ESTACIONARIO.

GENERACION DE CALOR EN DOM. CUADRADO (TRIANGULAR 3 NODOS)								
36	50	11	0	3	1	3	2	1
1	1	1	2	8				
2	1	2	3	9				
3	1	3	4	10				
4	1	4	5	11				
5	1	5	6	12				
6	1	8	7	1				
7	1	9	8	2				

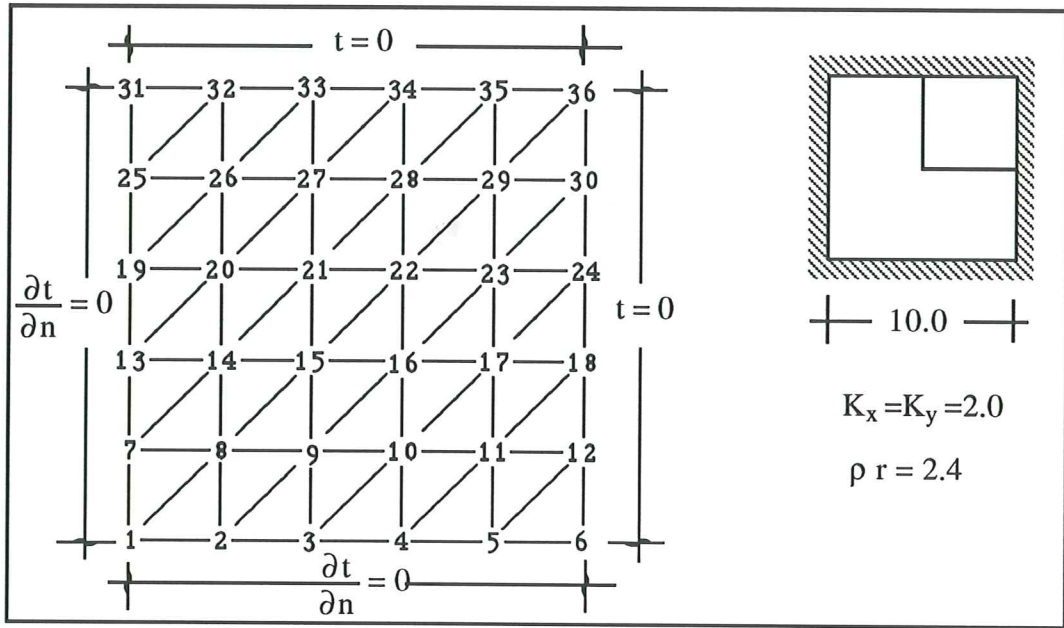


Figura 12 Geometría y discretización del problema de generación interna de calor en un dominio cuadrado.

8	1	10	9	3
9	1	11	10	4
10	1	12	11	5
11	1	7	8	14
12	1	8	9	15
13	1	9	10	16
14	1	10	11	17
15	1	11	12	18
16	1	14	13	7
17	1	15	14	8
18	1	16	15	9
19	1	17	16	10
20	1	18	17	11
21	1	13	14	20
22	1	14	15	21
23	1	15	16	22
24	1	16	17	23
25	1	17	18	24
26	1	20	19	13
27	1	21	20	14
28	1	22	21	15
29	1	23	22	16
30	1	24	23	17
31	1	19	20	26
32	1	20	21	27
33	1	21	22	28
34	1	22	23	29
35	1	23	24	30
36	1	26	25	19
37	1	27	26	20
38	1	28	27	21
39	1	29	28	22
40	1	30	29	23
41	1	25	26	32
42	1	26	27	33
43	1	27	28	34
44	1	28	29	35
45	1	29	30	36
46	1	32	31	25

```

47  1  33  32  26
48  1  34  33  27
49  1  35  34  28
50  1  36  35  29
  1  0.00000  0.00000
  2  1.00000  0.00000
  3  2.00000  0.00000
  4  3.00000  0.00000
  5  4.00000  0.00000
  6  5.00000  0.00000
  7  0.00000  1.00000
  8  1.00000  1.00000
  9  2.00000  1.00000
10  3.00000  1.00000
11  4.00000  1.00000
12  5.00000  1.00000
13  0.00000  2.00000
14  1.00000  2.00000
15  2.00000  2.00000
16  3.00000  2.00000
17  4.00000  2.00000
18  5.00000  2.00000
19  0.00000  3.00000
20  1.00000  3.00000
21  2.00000  3.00000
22  3.00000  3.00000
23  4.00000  3.00000
24  5.00000  3.00000
25  0.00000  4.00000
26  1.00000  4.00000
27  2.00000  4.00000
28  3.00000  4.00000
29  4.00000  4.00000
30  5.00000  4.00000
31  0.00000  5.00000
32  1.00000  5.00000
33  2.00000  5.00000
34  3.00000  5.00000
35  4.00000  5.00000
36  5.00000  5.00000
  6  1
12  1
18  1
24  1
30  1
31  1
32  1
33  1
34  1
35  1
36  1
  1          2.0e+00          2.0          1.0          2.4
  0  0
    generación interna de calor
  0  1  0

```

11.2.1.2 LISTADO DE RESULTADOS (GENERACION DE CALOR EN DOMINIO CUADRADO) CASO ESTACIONARIO.

PROBLEMA NO. 1

HEAT GENERATION IN A SQUARE DOMAIN (TRIANGULAR 3 NODOS)

NPNOD = 36 NELEM = 50 NPRES = 11 NTRAN = 0 *NTIPO = 1
 NNODE = 3 *NGDLN = 1 NMATS = 1 *NPROP = 4 NGAUS = 3
 NDIME = 2 *NTENS = 2 *NEVAB = 3 IWRTIT = 1

ELEMENTO	PROPIEDAD	NUMERO DE NODOS		
1	1	1	2	8
2	1	2	3	9
3	1	3	4	10
.
48	1	34	33	27
49	1	35	34	28
50	1	36	35	29

COORDENADAS DE PUNTOS NODALES

NODO	X	Y	Z
1	0.000	0.000	
2	1.000	0.000	
3	2.000	0.000	
.	.	.	.
34	3.000	5.000	
35	4.000	5.000	
36	5.000	5.000	

NODOS RESTRINGIDOS Y PRESCRIPCIONES

NODO	CODIGO	VALORES PRESCRITOS
6	1	0.00000
12	1	0.00000
18	1	0.00000
24	1	0.00000
30	1	0.00000
31	1	0.00000
32	1	0.00000
33	1	0.00000
34	1	0.00000
35	1	0.00000
36	1	0.00000

PROPIEDADES DE LOS MATERIALES

NUMERO	PROPIEDADES			
1	K=	0.20E+01	K=	0.20E+01
	ALPHA=	0.10E+01	ROR=	0.24E+01 ROC=
FRONTERAS CON FLUJO DE CONVECCION/RADIACION = 0				
NUMERO DE NODOS POR FRONTERA = 0				

generación interna

COND. DE CARGA PUNTUAL:.....	0
COND. DE CARGA DE GENER. INTERNA:....	1
COND. DE CALOR DISTRIBUIDO POR LADO..	0

FUERZAS TOTALES NODALES EQUIVALENTES ELEMENTALES

1	0.4000E+00	0.4000E+00	0.4000E+00
2	0.4000E+00	0.4000E+00	0.4000E+00
3	0.4000E+00	0.4000E+00	0.4000E+00

.

.

.

48	0.4000E+00	0.4000E+00	0.4000E+00
49	0.4000E+00	0.4000E+00	0.4000E+00
50	0.4000E+00	0.4000E+00	0.4000E+00

TEMPERATURAS

NODO	TEMPERATURA
1	0.898176E+01
2	0.858176E+01
3	0.761063E+01
4	0.595499E+01
5	0.347666E+01
6	0.000000E+00
7	0.858176E+01
8	0.826732E+01
9	0.735289E+01
10	0.576634E+01
11	0.337582E+01
12	0.000000E+00
13	0.761063E+01
14	0.735289E+01
15	0.656727E+01
16	0.518164E+01
17	0.306029E+01
18	0.000000E+00
19	0.595499E+01
20	0.576634E+01
21	0.518164E+01
22	0.413267E+01
23	0.248370E+01
24	0.000000E+00
25	0.347666E+01
26	0.337582E+01
27	0.306029E+01
28	0.248370E+01
29	0.154185E+01
30	0.000000E+00
31	0.000000E+00
32	0.000000E+00
33	0.000000E+00
34	0.000000E+00
35	0.000000E+00
36	0.000000E+00

0 REACCIONES

0	NODO	REACCION
	6	-0.387666E+01
	12	-0.795164E+01
	18	-0.732058E+01
	24	-0.616741E+01
	30	-0.428370E+01
	31	-0.387666E+01
	32	-0.795164E+01
	33	-0.732058E+01
	34	-0.616741E+01
	35	-0.428370E+01
	36	-0.800000E+00

TENSIONES

G.P. X-COORD Y-COORD X-FLUJ. Y-FLUJ.

ELEMENTO NO.=	1
1	0.3333 0.1667-0.8000E+00-0.62887E+00
2	0.8333 0.1667-0.8000E+00-0.62887E+00
3	0.8333 0.6667-0.8000E+00-0.62887E+00

```

ELEMENTO NO.=      2
1   1.3333   0.1667-0.19423E+01-0.51549E+00
2   1.8333   0.1667-0.19423E+01-0.51549E+00
3   1.8333   0.6667-0.19423E+01-0.51549E+00

```

```

.
.
.

```

```

ELEMENTO NO.=      49
1   3.6667   4.8333  0.00000E+00-0.49674E+01
2   3.1667   4.8333  0.00000E+00-0.49674E+01
3   3.1667   4.3333  0.00000E+00-0.49674E+01

```

```

ELEMENTO NO.=      50
1   4.6667   4.8333  0.00000E+00-0.30837E+01
2   4.1667   4.8333  0.00000E+00-0.30837E+01
3   4.1667   4.3333  0.00000E+00-0.30837E+01

```

FLUJOS ALISADAS NODALES

NODO	X-FLUJ.	Y-FLUJ.	Z-FLUJ.
1	-0.71444E+00	-0.71444E+00	
2	-0.15237E+01	-0.59108E+00	
3	-0.28089E+01	-0.46943E+00	
4	-0.43497E+01	-0.31876E+00	
5	-0.62205E+01	-0.13445E+00	
6	-0.69533E+01	0.00000E+00	
7	-0.59108E+00	-0.15237E+01	
8	-0.12145E+01	-0.12145E+01	
9	-0.24529E+01	-0.99529E+00	
10	-0.39104E+01	-0.70666E+00	
11	-0.56904E+01	-0.34046E+00	
12	-0.68189E+01	-0.67225E-01	
13	-0.46943E+00	-0.28089E+01	
14	-0.99529E+00	-0.24529E+01	
15	-0.21020E+01	-0.21020E+01	
16	-0.34165E+01	-0.15432E+01	
17	-0.50792E+01	-0.78964E+00	
18	-0.63309E+01	-0.21035E+00	
19	-0.31876E+00	-0.43497E+01	
20	-0.70666E+00	-0.39104E+01	
21	-0.15432E+01	-0.34165E+01	
22	-0.25745E+01	-0.25745E+01	
23	-0.39762E+01	-0.13619E+01	
24	-0.53518E+01	-0.38439E+00	
25	-0.13445E+00	-0.62205E+01	
26	-0.34046E+00	-0.56904E+01	
27	-0.78964E+00	-0.50792E+01	
28	-0.13619E+01	-0.39762E+01	
29	-0.22055E+01	-0.22055E+01	
30	-0.37116E+01	-0.62790E+00	
31	0.00000E+00	-0.69533E+01	
32	-0.67224E-01	-0.68189E+01	
33	-0.21035E+00	-0.63309E+01	
34	-0.38439E+00	-0.53518E+01	
35	-0.62790E+00	-0.37116E+01	
36	-0.15419E+01	-0.15419E+01	

En la siguiente gráfica se muestra la distribución de temperaturas en el dominio mediante isotermas, así como las distribuciones de los flujos calóricos en X e Y.

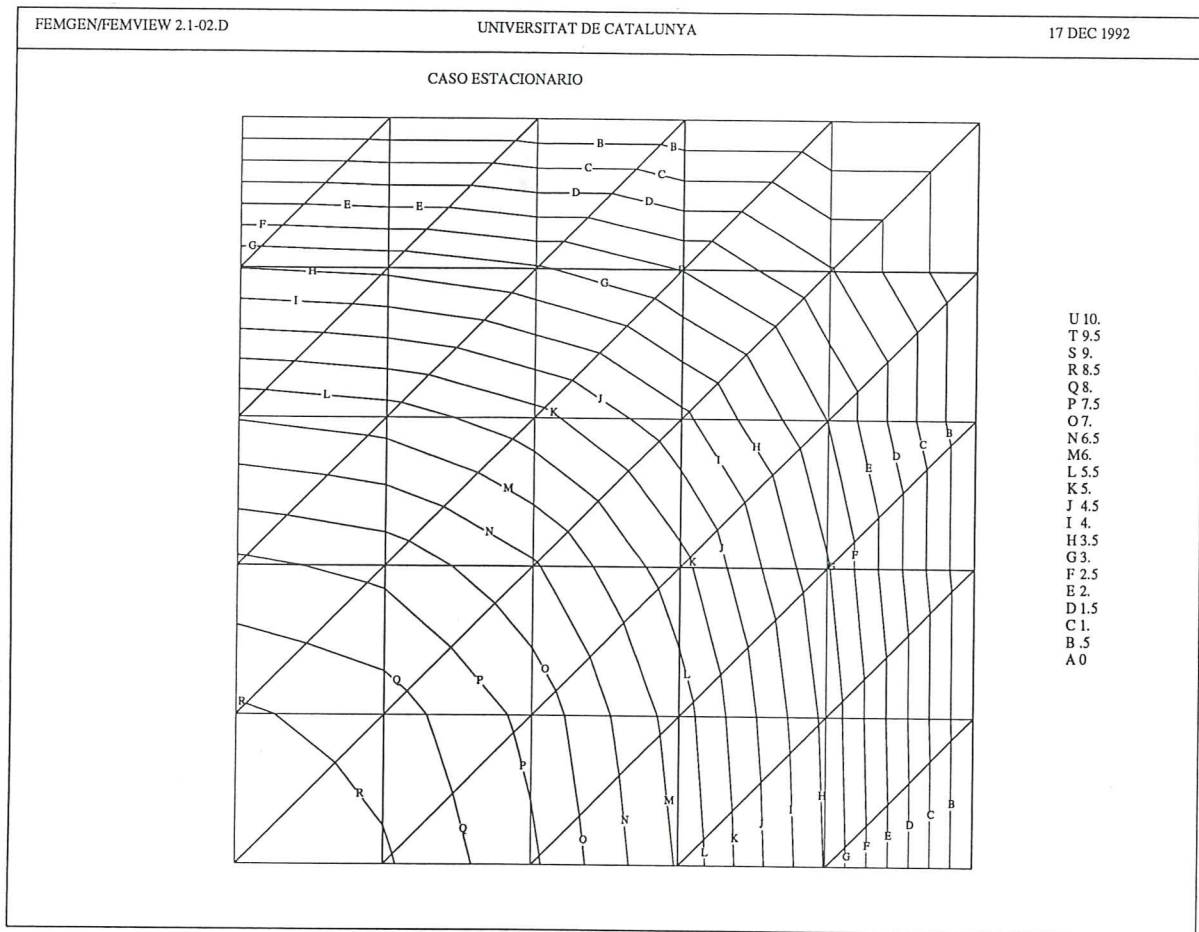


Figura 13 Distribución de Isotermas en caso estacionario.

11.2.2 Caso transitorio $\alpha = 1.0$

La geometría del problema, en este caso es la correspondiente a la figura 12, considerando un valor de para rr. y de 1.0 para a, por lo que se utiliza un esquema de integración temporal fuertemente implícito [21], utilizando 25 iteraciones en el tiempo, partiendo de una temperatura inicial de todos los nodos igual a cero.

De acuerdo con la metodología establecida para la lectura de datos (apéndice I), a las tarjetas del apartado 11.2.1 se adicionan las que se describen en el siguiente párrafo.

11.2.2.1 LISTADO DE DATOS CASO TRANSITORIO $\alpha = 1.0$.(TARJETAS ADICIONALES)

La primer tarjeta corresponde a los datos generales, mientras que la siguiente tarjeta que se modifica, se refiere a las propiedades del material tipo 1, precediendo las correspondientes a la integración temporal.

```

36  50  11  1  3  1  3  2  1
.
.
.
1      2.0e+00      2.0      1.0      2.4      1.0
25     1.0000      1.00
36     0.0000
0      0
      generación interna de calor
.
.
.

```

11.2.2.2 LISTADO DE RESULTADOS CASO TRANSITORIO $\alpha = 1.0$

Dado que los resultados numéricos son muy extensos, solo se muestra los valores de la temperatura de dos nodos, durante toda la evolución del problema; así mismo se gráfica las isotermas sobre el dominio para los pasos de tiempo 1, 9 17 y 25

TEMPERATURAS PASO 1 NODO TEMPERATURA 1 0.219147E+01 2 0.215197E+01	TEMPERATURAS PASO 2 NODO TEMPERATURA 1 0.399268E+01 2 0.388487E+01	TEMPERATURAS PASO 3 NODO TEMPERATURA 1 0.537256E+01 2 0.519438E+01
TEMPERATURAS PASO 4 NODO TEMPERATURA 1 0.639050E+01 2 0.615336E+01	TEMPERATURAS PASO 5 NODO TEMPERATURA 1 0.712784E+01 2 0.684554E+01	TEMPERATURAS PASO 6 NODO TEMPERATURA 1 0.765746E+01 2 0.734191E+01
TEMPERATURAS PASO 7 NODO TEMPERATURA 1 0.803644E+01 2 0.769684E+01	TEMPERATURAS PASO 8 NODO TEMPERATURA 1 0.830718E+01 2 0.795032E+01	TEMPERATURAS PASO 9 NODO TEMPERATURA 1 0.850044E+01 2 0.813123E+01
TEMPERATURAS PASO 10 NODO TEMPERATURA 1 0.863836E+01 2 0.826033E+01	TEMPERATURAS PASO 11 NODO TEMPERATURA 1 0.873676E+01 2 0.835244E+01	TEMPERATURAS PASO 12 NODO TEMPERATURA 1 0.880697E+01 2 0.841815E+01
TEMPERATURAS PASO 13 NODO TEMPERATURA 1 0.885705E+01 2 0.846504E+01	TEMPERATURAS PASO 14 NODO TEMPERATURA 1 0.889279E+01 2 0.849849E+01	TEMPERATURAS PASO 15 NODO TEMPERATURA 1 0.891829E+01 2 0.852235E+01
TEMPERATURAS PASO 16 NODO TEMPERATURA 1 0.893647E+01 2 0.853937E+01	TEMPERATURAS PASO 17 NODO TEMPERATURA 1 0.894945E+01 2 0.855152E+01	TEMPERATURAS PASO 18 NODO TEMPERATURA 1 0.895871E+01 2 0.856018E+01
TEMPERATURAS PASO 19 NODO TEMPERATURA 1 0.896531E+01 2 0.856637E+01	TEMPERATURAS PASO 20 NODO TEMPERATURA 1 0.897003E+01 2 0.857078E+01	TEMPERATURAS PASO 21 NODO TEMPERATURA 1 0.897339E+01 2 0.857392E+01
TEMPERATURAS PASO 22	TEMPERATURAS PASO 23	TEMPERATURAS PASO 24

NODO	TEMPERATURA
1	0.897579E+01
2	0.857617E+01

NODO	TEMPERATURA
1	0.897750E+01
2	0.857777E+01

NODO	TEMPERATURA
1	0.897872E+01
2	0.857891E+01

TEMPERATURAS PASO 25

NODO	TEMPERATURA
1	0.897959E+01
2	0.857973E+01

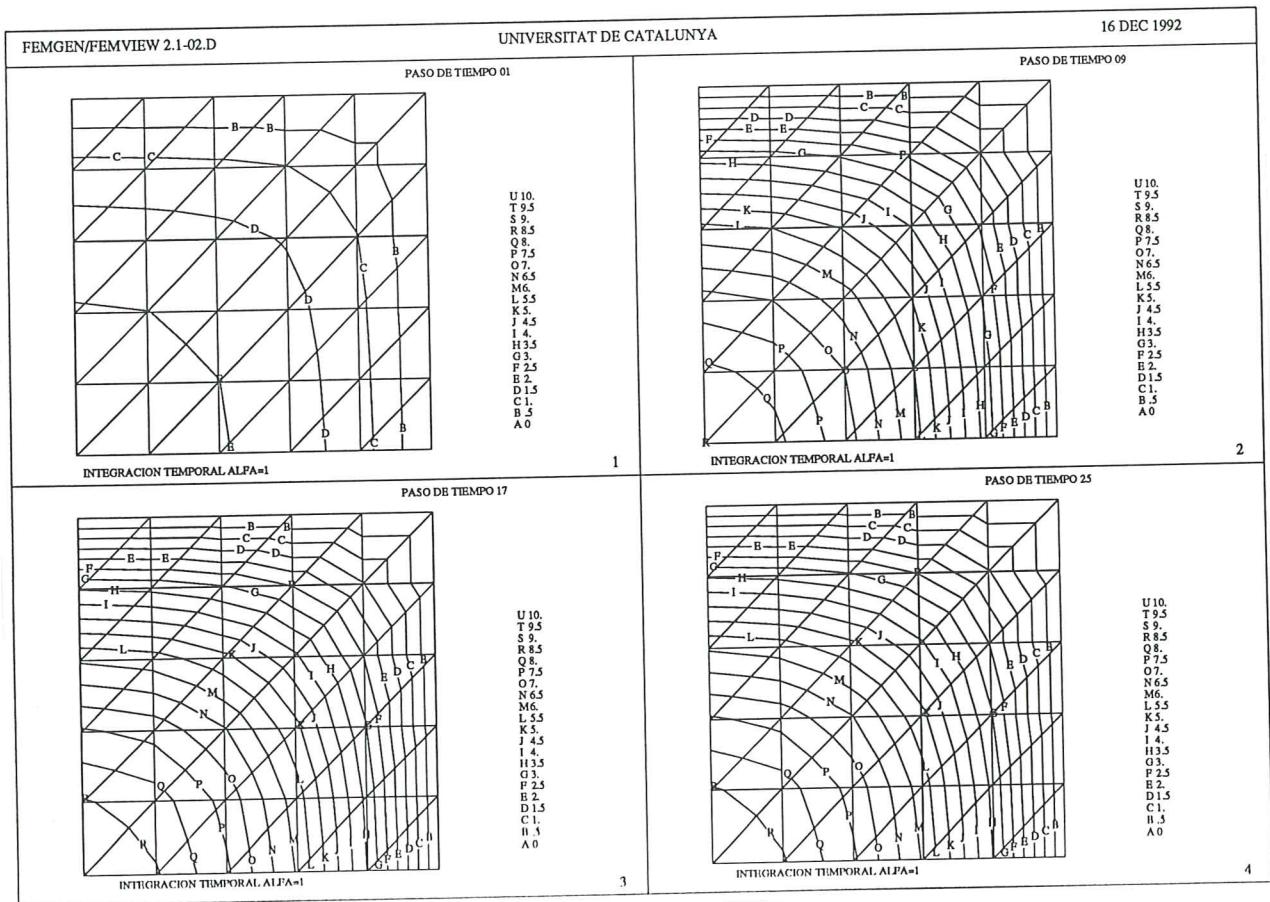


Figura 14 Distribución de isothermas

11.2.3 Caso transitorio $\alpha = 2/3$

La geometría del problema corresponde integramente a los datos del apartado 11.2.2, con excepción del valor de a que, usando la integración de Galerkin corresponde a $2/3$.

11.2.3.1 LISTADO DE DATOS CASO TRANSITORIO $\alpha = 2/3$.(TARJETAS ADICIONALES)

La tarjeta que se modifica respecto al problema anterior corresponde a la referente a la integración temporal.

25 1.0000 0.66666

11.2.3.2 LISTADO DE RESULTADOS CASO TRANSITORIO $\alpha = 2/3$

Dado que los resultados numéricos son muy extensos, solo se muestra los valores de la temperatura de dos nodos, durante toda la evolución del problema; así mismo se gráfica las isotermas sobre el dominio para los pasos de tiempo 1, 9 17 y 25

TEMPERATURAS PASO 1 NODO TEMPERATURA 1 0.231373E+01 2 0.228730E+01	TEMPERATURAS PASO 2 NODO TEMPERATURA 1 0.430091E+01 2 0.419724E+01	TEMPERATURAS PASO 3 NODO TEMPERATURA 1 0.577800E+01 2 0.558338E+01
TEMPERATURAS PASO 4 NODO TEMPERATURA 1 0.679258E+01 2 0.653223E+01	TEMPERATURAS PASO 5 NODO TEMPERATURA 1 0.748559E+01 2 0.718147E+01	TEMPERATURAS PASO 6 NODO TEMPERATURA 1 0.795982E+01 2 0.762518E+01
TEMPERATURAS PASO 7 NODO TEMPERATURA 1 0.828354E+01 2 0.792823E+01	TEMPERATURAS PASO 8 NODO TEMPERATURA 1 0.850478E+01 2 0.813530E+01	TEMPERATURAS PASO 9 NODO TEMPERATURA 1 0.865590E+01 2 0.827675E+01
TEMPERATURAS PASO 10 NODO TEMPERATURA 1 0.875915E+01 2 0.837339E+01	TEMPERATURAS PASO 11 NODO TEMPERATURA 1 0.882968E+01 2 0.843941E+01	TEMPERATURAS PASO 12 NODO TEMPERATURA 1 0.887786E+01 2 0.848451E+01
TEMPERATURAS PASO 13 NODO TEMPERATURA 1 0.891078E+01 2 0.851532E+01	TEMPERATURAS PASO 14 NODO TEMPERATURA 1 0.893327E+01 2 0.853637E+01	TEMPERATURAS PASO 15 NODO TEMPERATURA 1 0.894863E+01 2 0.855075E+01
TEMPERATURAS PASO 16 NODO TEMPERATURA 1 0.895913E+01 2 0.856058E+01	TEMPERATURAS PASO 17 NODO TEMPERATURA 1 0.896630E+01 2 0.856729E+01	TEMPERATURAS PASO 18 NODO TEMPERATURA 1 0.897120E+01 2 0.857187E+01
TEMPERATURAS PASO 19 NODO TEMPERATURA 1 0.897454E+01 2 0.857501E+01	TEMPERATURAS PASO 20 NODO TEMPERATURA 1 0.897683E+01 2 0.857714E+01	TEMPERATURAS PASO 21 NODO TEMPERATURA 1 0.897839E+01 2 0.857861E+01
TEMPERATURAS PASO 22 NODO TEMPERATURA 1 0.897946E+01 2 0.857961E+01	TEMPERATURAS PASO 23 NODO TEMPERATURA 1 0.898019E+01 2 0.858029E+01	TEMPERATURAS PASO 24 NODO TEMPERATURA 1 0.898069E+01 2 0.858075E+01
TEMPERATURAS PASO 25 NODO TEMPERATURA 1 0.898103E+01 2 0.858107E+01		

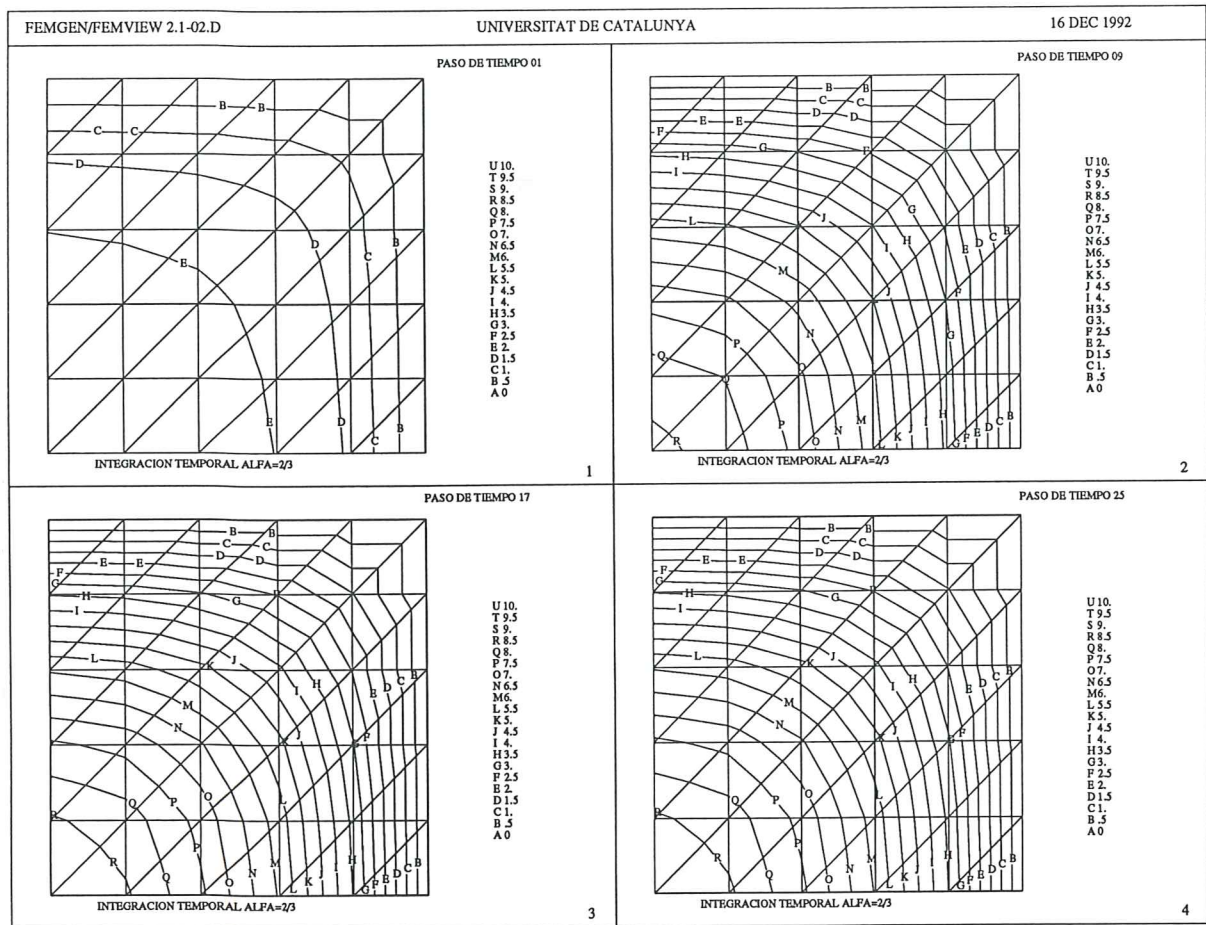


Figura 15 Distribución de isotermas

Con objeto de comparar ambos esquemas de integración, en la figura 16 se presenta la convergencia al valor estacionario de la temperatura del nodo 1

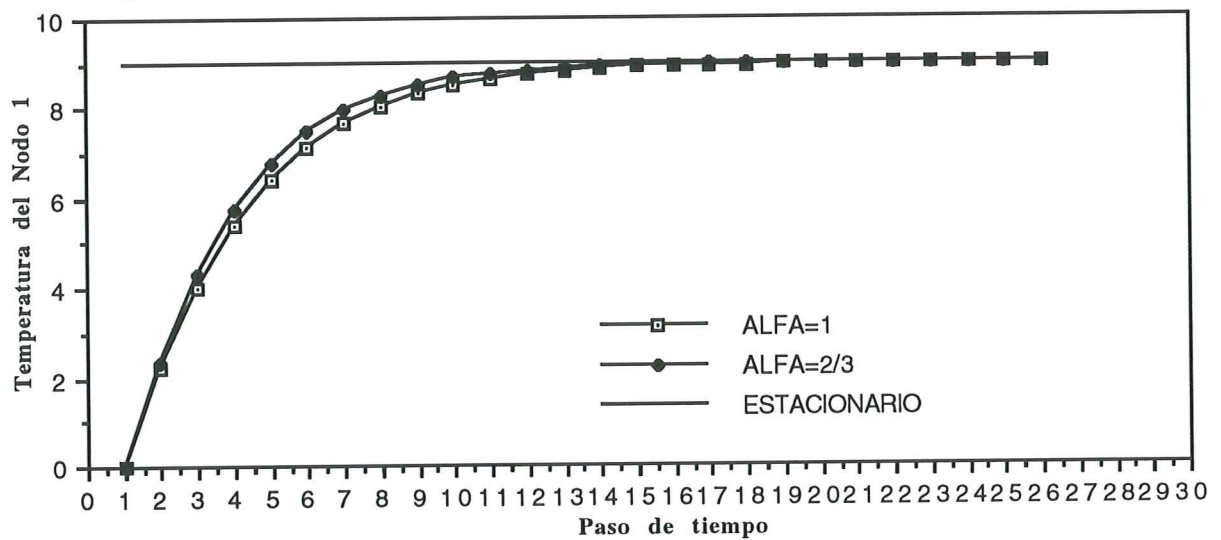


Figura 16 Convergencia de temperatura con distintos esquemas de integración

REFERENCIAS

1. Mackerle, J. y Fredriksson B. *Handbook of Finite Element Software*, Chartwell-Bratt Ltd., Bromley, U.K., 1990.
2. Hinton, E. y Owen, D.R.J., *Finite element programming*, Academic Press, 1979.
3. Pres, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., *Numerical Recipes. The art of Scientific Computing*, Cambridge Univ. Press, 1986.
4. Oñate, E. "Cálculo de estructuras por el método de los elementos finitos", CIMNE, Barcelona, 1992.
5. FEMGEN/FEMVIEW, The Finite Element Pre and Post-processor, User Manual, Femview Limited, Leicester, U.K., 1991.
6. I-DEAS, General capabilities, P-0100, Structural Dynamics Research Corporation, Milford, Ohio, USA, 1991.
7. PATRAN Plus User Manual, PDA Engineering, PATRAN Division, Costa Mesa, California, USA, 1991.
8. Bugada, G., "FLAVIA. Programa para visualización gráfica en 2 y 3 dimensiones", Publicación CIMNE, Barcelona, 1992.
9. Fong, H. M., "Interactive graphics and commercial finite element code, Mechanical" Engng., 106(6), pp. 18-27, 1989.
10. Bonet, J. y Peraire, J., "An alternate digital tree algorithm for geometric searching and intersection problems", Int. J. Num. Meth. Engng. Vol. 31, pp 1-17, 1991.
11. Hinton, E. y Owen, D.R.J., *Introduction to finite element computations*, Pineridge Press, 1980.
12. Nguye, V. P., "Automatic mesh generation with tetraedon elements", Int. J. Num. Meth. Engng. Vol. 18, pp 273-289, 1982.
13. Peiró, J. *A finite element procedure for the solution of the Euler equations on unstructured meshes*, Ph.D. Thesis, Civil Eng. Dpt., Univ. College of Swansea, U.K., 1989.
14. Pissanetzky, S., "Kubik: An automatic three dimensional mesh generator", Int. J. Num. Meth. Engng. Vol. 17, pp 255-269, 1987.
15. Zienkiewicz, O.C., *El Método de los Elementos Finitos*, 3ra. Edición, Ed. Reverté, Barcelona, 1979.
16. Zienkiewicz, O.C., *The Finite Element Method*, 4ra. Edición, Ed. Mc. Graw Hill, Vol. I, 1989, Vol II, 1991.
17. Ralston, A., *Introducción al análisis numérico*, Limusa-Wiley, 1970.
18. Irons, B.M. y Ahmad, S., *Techniques of finite elements*, Ellis Harwood, Chichester, 1980.
19. Timoshenko, S.P. y Goodier, J.N., *Teoría de la elasticidad*, Edic. Urmo, 1968.
20. Timoshenko, S.P. y Woinowsky-Krieger S., *Teoría de Placas y Láminas*, Edic. Urmo, Bilbao, 1990.
21. Marshall, Guillermo, *Solución numérica de ecuaciones diferenciales*, Tomo I, d. Reverté Argentina S.A. Argentina 1985.

APENDICE I

INSTRUCCIONES PARA ENTRADA DE DATOS Y LISTADO DE VARIABLES DEL PROGRAMA CALTEP

I.I INSTRUCCIONES PARA ENTRADA DE DATOS

Las instrucciones se han agrupado en "tarjetas" formateadas. Cada "grupo de tarjetas" incluye un conjunto de datos similares; ejemplo: coordenadas nodales, cargas puntuales, etc.

GRUPO DE TARJETAS 1. NUMERO DE PROBLEMAS Y TITULO DE CADA PROBLEMA.

TARJETA 1.1	I5	NPROB	Número de problemas a analizar.
TARJETA 1.2	A80	TITULO	Título de cada problema.

NOTA: Tantas tarjetas 1.2 como número de problemas a analizar.

GRUPO DE TARJETAS 2. PARAMETROS DE CONTROL.

TARJETA 2.1	9I5	NPNOD	Número total de nodos.
		NELEM	Número de elementos.
		NPRES	Número de nodos con temp. prescritas.
		NTRAN	Indicador de tipo de problema.
			=0 Caso estacionario
			=1 Caso transitorio, si $\alpha=0$, no diagonaliza.
			=2 Caso transitorio, si $\alpha=0$, diagonalización por suma de filas
			=3 Caso transitorio, si $\alpha=0$, diagonalización conservación de masa.
		NNODE	Número de nodos por elemento.
		NMATS	Número de materiales.
		NGAUS	Número de puntos de Gauss.
		NDIME	Número de dimensiones.
		IWRIT	Indicador para escritura de datos.
			=1 Escribe fichero de resultados.

GRUPO DE TARJETAS 3. CONECTIVIDADES NODALES DE CADA ELEMENTO

TARJETA 3.1	15I5	NUMEL	Número de elemento.
		MATNU(NUMEL)	Tipo de material en elemento.
		LNODS(NUMEL,INODE)	Conectividades (bucle sobre NNODE)
TARJETA 3.2	15I5		Continúan conectividades en elem. 3D.

NOTAS:

Tantas tarjetas 3.1 como número de elementos (NELEM)
La tarjeta 3.2 se utilizar solamente para NNODE=20

GRUPO DE TARJETAS 4. COORDENADAS NODALES

TARJETA 4.1 I5,3F10.5
IPNOD Número de nodo.
COORD(IPNOD,1) Coordenada X del nodo.
COORD(IPNOD,2) Coordenada Y del nodo (en 2D)
COORD(IPNOD,3) Coordenada Z del nodo (en 3D)

NOTA:

Tantas tarjetas 4.1 como número de nodos (NPNOD)

GRUPO DE TARJETAS 5. MOVIMIENTOS PRESCRITOS

TARJETA 5.1 1X,I4,3X,<NGDLN>I1,<NGDLN>F10.5
NODPR(IPRES) Nodo.
INPRE(IPRES,NGLN) Código de temperatura prescrito
=0 Temperatura libre.
=1 Temperatura prescrita.
PRESC(IPRES,NGDLN) Valor de la Temperatura prescrita.

NOTA:

Tantas tarjetas 5.1 como número de nodos prescritos (NPRES)

GRUPO DE TARJETAS 6. PROPIEDADES DE LOS MATERIALES

TARJETA 6.1 1X,I4,5E15.5
NUMAT Número de material.
PROPS(NUMAT,1..NDIME) Conductividad Térmica en X, Y Z.
PROPS(NUMAT,NDIME+1) Alpha.
PROPS(NUMAT,NDIME+2) roR.
PROPS(NUMAT,NDIME+3) roC.

NOTA:

Tantas tarjetas 6.1 como número de propiedades (NDIME+2) si NTRAN = 0 ó (NDIME+3) si NTRAN > 0

GRUPO DE TARJETAS 7. INTEGRACION EN EL TIEMPO

TARJETA 7.1 I5,2E15.5
NTIME Número de iteraciones en el tiempo.
DTIME Incremento de tiempo en cada iteración.
ALFAT Esquema de Intergación temporal.

TARJETA 7.2 I5,2E15.5
NNCIT Número de nodos con condición de temperatura inicial distinta de cero.
CITIM Temperatura inicial generalizada.

TARJETA 7.3 I5,2E15.5
IPNOD Nodo con condición de temperatura inicial distinta de cero.
XTIME(IPNOD) Temperatura inicial del nodo (bucle sobre NNCIT).

NOTAS:

Solo sera necesario introducir el grupo 7 cuando NTRAN>0

En caso de tener una temperatura inicial distinta de cero igual en todos los nodos solo basta introducir NNCIT=NPNOD y dar el valor en CITIM, si este no es el caso CITIM carece de significado.

El grupo de tarjetas 7.3 habrá tantas como NNCIT; si NNCIT=NPNOD no serán necesarias.

GRUPO DE TARJETAS 8. CONDICIONES DE FRONTERA DE FLUJO POR RADICACION / CONVECCION

TARJETA 8.1	1X,2I4 NFRON NNFRO	Número de elementos de frontera Número de nodos en cada lado.
TARJETA 8.2	1X,<1+NNFRO>I4 CONVE(IFRON,1) CONVE(IFRON,1+NNFRO)	Número de elemento de frontera. Número de nodos de frontera (numeración global)

NOTA:

Solo sera necesario introducir el grupo 8.2 cuando NFRON <> 0

GRUPO DE TARJETAS 9. PARAMETROS DE LAS CONDICIONES DE FRONTERA

TARJETA 9.1	A20 TITULO	Titulo del estado de carga.
TARJETA 9.2	4I5 IPUNT IPESO IDIST	Indicador de flujo puntual. =0 No existe el caso. =1 Existe el caso. Indicador de generación interna de calor. =0 No existe el caso. =1 Existe el caso. Indicador de flujo y temperatura externa aplicada por lado. =0 No existe el caso. =1 Existe el caso.

GRUPO DE TARJETAS 10. FLUJOS PUNTUALES

TARJETA 10.1	I5,F10.3 LODPT PNODT(1)	Nodo cargado. Flujo puntual nodal.
---------------------	-------------------------------	---------------------------------------

NOTAS:

Se repite la tarjeta 9.1 para cada nodo cargado. La última tarjeta 9.1 debe corresponder al nodo de mayor numeración, con independencia de que esté o no cargado (en este último caso PNODT=0.0).
Si IPUNT=0 se omite este grupo de tarjetas.

GRUPO DE TARJETAS 11. FLUJO Y TEMPERATURA EXTERNA ACTUANTES SOBRE UN LADO DEL ELEMENTO.

TARJETA 11.1	2I5 NEDGE NODEG	Número de lados expuestos. Número de nodos por lado expuesto.
TARJETA 11.2	I5,<NODEG>I5 NEASS NOPRS(INODEG)	Número del elemento cargado. Nodo cargado en numeración global.
TARJETA 11.3	2F10.3 FLUJO TEMPE	Flujo actuante sobre el lado. Temperatura externa sobre el lado.

NOTAS:

El flujo normal es positivo si va dirigido hacia el interior del elemento.
Si IDIST=0 se omite el grupo de tarjetas 11.
Tantas tarjetas 11.3 como nodos por lado y tantos grupos de tarjetas 11.2 y 11.3 como número de lados

I.II LISTADO DE VARIABLES DEL PROGRAMA CALTEP.

VARIABLES ESCALARES

ALFAT	Valor α del esquema de integración.
ALFAU	$1-\alpha$.
AUX3	Variable auxiliar de direccionamiento de ficheros.
B	Variable auxiliar operacional.
CERO	Variable Dummy (siempre igual a 0).
CITIM	Temperatura inicial generalizada.
DAREA	Diferencial de área del elemento.
DJACB	Determinante del Jacobiano.
DLONG	Diferencial de longitud.
DVOLU	Diferencial de volumen.
EGASP	Variable Dummy sobre EGISP (siempre igual a cero).
EGISP	Coordenada ζ del punto de Gauss.
ERROR	Variable de condición de error.
ETASP	Coordenada ξ del punto de Gauss.
EXISP	Coordenada η del punto de Gauss.
FACTR	Factor de multiplicación de la ecuación a reducir.
FLUJO	Flujo actuante sobre el lado.
ICOLS	Contador sobre el número de columnas de la matriz global.
IDIME	Contador sobre el número de dimensiones.
IDIST	Indicador de flujo y temperatura externa .
IELEM	Contador sobre el número de elementos.
IEON1	Contador sobre el número de ecuaciones.
IEONS	Contador sobre el número de ecuaciones.
IEVAB	Contador sobre el número de variables nodales.
IFRON	Contador sobre nodos de frontera NFRON .
IGASH	Contador sobre cuadraturas no triangulares de Gauss.
IGAUS	Contador sobre los puntos de Gauss.
IGDLN	Contador sobre los grados de libertad nodales.
IMATS	Contador sobre número de materiales.
INCIT	Contador sobre NNCIT .
INFRO	Contador sobre los nodos de frontera.
INODE	Contador sobre los nodos del elemento.
INPUT	Nombre del fichero de datos.
IODEG	Contador sobre los nodos por lado expuesto.
IPESO	indicador de generación interna de calor.
IPNOD	Contador sobre los nodos de la malla.
IPRES	Contador sobre los nodos prescritos.
IPROB	Contador sobre el Número de problemas.
IPROP	Contador sobre las propiedades de los materiales.
IPUNT	Indicador de temperatura puntual.
IROWS	Contador sobre el número de filas de la matriz global.
ITENS	Contador Dummy sobre NTENS .
ITIME	Contador sobre las iteraciones en tiempo.
ITOTV	Contador sobre el número de temperaturas nodales.
IVARI	Contador sobre el vector de temperaturas nodales.
IWRIT	Indicador para escritura de datos.
JDIME	Contador sobre el número de dimensiones.
JELEM	Contador sobre el número de elementos.

JEVAB	Contador sobre el número de variables nodales.
JGASH	Contador sobre cuadraturas no triangulares de Gauss.
JGAUS	Contador sobre los puntos de Gauss.
JGDLN	Contador sobre los grados de libertad nodales.
JNFRO	Contador sobre los nodos de frontera.
JNODE	Contador sobre los nodos del elemento.
JPNOD	Contador sobre los nodos de la malla.
JTENS	Contador Dummy sobre NTENS.
JTOTV	Contador sobre el número de temperaturas nodales.
KCONT	Contador sobre grados de libertad.
KGAUS	Contador sobre los puntos de Gauss.
KNODE	Contador sobre los nodos del elemento.
KPGAU	Contador sobre puntos de integración numérica.
LNODE	Variable auxiliar operacional.
LODPT	Nodo con temperatura puntual.
LPROP	Tipo de material para cada elemento.
NBAC1	Contador decreciente del número de ecuaciones.
NBACK	Contador decreciente del número de ecuaciones.
NCOLE	Columna de la matriz de rigidez elemental del nodo JNODE .
NCOLS	Columna de la matriz de rigidez global del nodo JNODE .
NCONT	Apuntador sobre los grados de libertad.
NDIME	Número de dimensiones en el problema.
NEASS	Número del elemento cargado.
NEDGE	Número de lados expuestos.
NELEM	Número de elementos total.
NEONS	Número de ecuaciones a resolver.
NEVAB	Número de variables por elemento.
NFRON	Número de fronteras de convección/radiación.
NGAUS	Número de puntos de integración de Gauss.
NGDLN	Grados de libertad nodales (siempre igual a 1).
NGISH	Número de grados de libertad prescritos.
NGUSH	Contador sobre los grados de libertad prescritos.
NLOCA	Localizador del nodo con temperatura puntual.
NMATS	Número de materiales distintos en la geometría.
NNCIT	Número de nodos con condición de temp. inicial distinta de cero.
NNFRO	Número de nodos de frontera convección/radiación.
NNOD1	Variable auxiliar para la lectura de la tarjeta 3.2.
NNOD2	Variable auxiliar para la lectura de la tarjeta 3.2.
NNODE	Número de nodos por elemento.
NODEG	Número de nodos por lado expuesto.
NODEI	INODE en numeración global.
NODEJ	JNODE en numeración global.
NPNOD	Número de nodos en toda la malla.
NPOSN	Localizador de nodo donde aplicar la temperatura puntual.
NPRES	Número de prescripciones nodales.
NPROB	Número de problemas a resolver.
NPROP	Número de propiedades por cada material.
NROWE	Fila de la matriz de rigidez elemental del nodo INODE .
NROWS	Fila de la matriz de rigidez global del nodo INODE .
NSTR1	Numero de flujos finales.
NTENS	Variable Dummy (siempre igual a NDIME).
NTIME	Número de iteraciones sobre el tiempo.
NTIPO	Variable Dummy (siempre igual a 1).
NTOTV	Número total de temperaturas nodales.
NTRAN	Indicador de problema transitorio o estacionario.

NUMAT	Número de materiales.
NUMEL	Número de elementos.
OUTPUT	Nombre del Fichero de resultados.
PATOP	Variable auxiliar operacional.
PESPP	Fuente de calor interna por densidad.
PIVOT	Elemento pivote de la matriz de rigidez, en la reducción.
Q	Coordenada ζ del punto de Gauss.
Q2	Valor de $2*\zeta$.
QQ	Valor $\zeta*\zeta$.
RESID	Reacción de la ecuación a reducir.
ROCEE	Densidad por calor por densidad entre incremento de tiempo.
S	Coordenada ξ del punto de Gauss.
S 1	Valor $\xi+1$.
S 2	Valor $2*\xi$.
S 9	Valor $\xi-1$.
SS	Valor $\xi*\xi$.
SST	Valor $\xi*\xi*\eta$.
ST	Valor $\xi*\eta$.
ST2	Valor $2*\xi*\eta$.
STT	Valor $\xi*\eta*\eta$.
T	Coordenada η del punto de Gauss.
T 1	Valor $\eta+1$.
T 2	Valor $2*\eta$.
T 9	Valor $\eta-1$.
TEMPE	Temperatura externa sobre el lado.
TITULO	Título del problema y/o estado de cargas.
TT	Valor $\eta*\eta$.
UNOVA	Variable Dummy (siempre igual a 1).

VARIABLES VECTORIALES

ASLOD(200)	Temperaturas nodales equivalentes globales.
BMATZ (3,81)	Matriz B
BSLOD(200)	Temperaturas nodales equivalentes globales.
CARGA (200,20)	Matriz para almacenar temperaturas nodales
CONVE (200,27)	Elementos y nodos de frontera por radiación convección.
COORD (200,3)	Coordenadas nodales
COREL (3,27)	Coordenadas nodales locales
CORPG (3,27)	Coordenadas en los puntos de gauss.
D(2)	Diferencial de x e y.
DBMAT (3,81)	Matriz del producto DB .
DCART (3,27)	Derivadas cartesianas de las funciones de forma.
DERIV(3,20)	Derivadas naturales de las funciones de forma.
DMATZ (3,3)	Matriz constitutiva D .
FFORM(20)	Funciones de forma.
FTFOR(20,20)	Matriz de rigidez por aportación de condiciones de frontera.
IFFIX(200,3)	Indicador de temperaturas prescritas globales.
INPRE (200,3)	Indicador de temperaturas prescritas elementales.
JFFIX(200)	Indicador de temperaturas prescritas globales.
LNODS (200,27)	Conectividades nodales.
LOCAL(8)	Numeración local de nodos de frontera.
MATNU (200)	Tipo de material del elemento.

NODPR (200)	Numero de nodos prescritos.
PESPG(14)	Pesos de los puntos de integración en cuadriláteros.
PNODT(3)	Conectividades para flujos repartidos.
POSGT(14)	Coordenadas de los pts. de integración de triángulos.
POSPG(14)	Coordenadas de pts. de integración en cuadriláteros.
PRESC (200,3)	Valores de las temperaturas prescritas.
PRESS(20,2)	Valor del flujo y temp. externa por lado.
PROPS (10,200)	Propiedades del material.
PXCOM(20)	Temperatura nodal equivalente por elemento.
RIGID(20,20)	Matriz de rigidez elemental.
TENSG(3)	Flujos en cada punto de integración.
TENSZ (3,81,27)	Flujos elementales en cada punto de integración.
XFORM(20)	Vector de funciones de forma de lado con convecc.Rad.
XJACI(3,3)	Matriz Jacobiana inversa.
XJACM(3,3)	Matriz Jacobiana.
XTIME (200)	Temperatura en el tiempo n-1.
ZJACC(3,3)	Matriz Jacobiana.
ZMASA(20,20)	Matriz de masa elemental.

APENDICE II

LISTADO DEL PROGRAMA CALTEP

```

1.          PROGRAM CALTEP
2.          C*****
3.          C
4.          C*** SUBROUTINA PRINCIPAL
5.          C
6.          C*****
7.          C*****COMMON STATEMENTS
8.          IMPLICIT NONE
9.          INCLUDE 'dtsgral.f'
10.         INCLUDE 'data.f'
11.         INCLUDE 'gaussdat.f'
12.         INCLUDE 'calculo.f'
13.         INCLUDE 'solu.f'
14.         INCLUDE 'solucuas.f'
15.         INTEGER*2 MPROB, IPROB, IVARI, ITIME
16.         CHARACTER*80 TITULO
17.         CHARACTER*10 INPUT, OUTPUT
18.         CHARACTER*13 AUX3
19.         DATA AUX3 /'aux3/zarate/'/
20.         C*****COMMON STATEMENTS
21.         OPEN (2, FILE='COMAN.DAT', FORM='FORMATTED')
22.         READ(2, 800) INPUT
23.         READ(2, 800) OUTPUT
24.         CLOSE(2, STATUS='KEEP')
25.         800 FORMAT(A10)
26.         OPEN (5, FILE=INPUT, FORM='FORMATTED')
27.         OPEN (6, FILE=OUTPUT, FORM='FORMATTED')
28.         OPEN (1, FILE=AUX3//'TEMP1.TMP', FORM='UNFORMATTED')
29.         OPEN (3, FILE=AUX3//'TEMP3.TMP', FORM='UNFORMATTED')
30.         OPEN (4, FILE=AUX3//'TEMP4.TMP', FORM='UNFORMATTED')
31.         OPEN (8, FILE=AUX3//'TEMP8.TMP', FORM='UNFORMATTED')
32.         OPEN (10, FILE=AUX3//'TEMP10.TMP', FORM='UNFORMATTED')
33.         OPEN (29, FILE=AUX3//'CALSEF.POS', FORM='UNFORMATTED')
34.         C
35.         C*** LEE NUMERO DE PROBLEMAS A ANALIZAR
36.         C
37.         READ(5, 900) NPROB
38.         900 FORMAT(I5)
39.         WRITE(6, 905) NPROB
40.         905 FORMAT(//10X, 'NUMERO DE PROBLEMAS= ', I5)
41.         C
42.         C*** BUCLE SOBRE NUMERO DE PROBLEMAS
43.         C
44.         MPROB=INT2(NPROB)
45.         ITIME=0
46.         WRITE(29) MPROB
47.         DO IPROB=1, NPROB
48.             REWIND 1
49.             REWIND 3
50.             REWIND 4
51.             WRITE(6, 910) IPROB
52.         910  FORMAT(///// , 6X, 'PROBLEMA NO. ', I3, ////)
53.             READ(5, 915) TITULO
54.         915  FORMAT(A80)

```

```

55.          WRITE(6,920) TITULO
56.      920  FORMAT(A80,////)
57.          CALL DATOS
58.          CALL GAUSSQ(NDIME,NNODE,NGAUS)
59.          CALL RIGIMAT
60.          CALL FUERZAS
61.          CALL ENSAMBLA
62.          IF (NTRAN.EQ.0) THEN
63.              CALL REDUCE
64.              CALL SUSTITUIR
65.              CALL TENSIONES
66.              CALL SUAV1
67.              CALL FEMV(ITIME,TITULO)
68.          ELSE
69.              IF (ALFAT.EQ.0) CALL DIAGONAL
70.              DO ITIME=1,NTIME
71.      C
72.      C***      APLICA LAS CONDICIONES DE TIEMPO T-1
73.      C
74.              CALL FUERZAS_T0
75.      C
76.      C***      RESUELVE POR ELIMINACION GAUSSIANA
77.      C
78.              IF (ITIME.EQ.1) THEN
79.                  CALL REDUCE
80.              ELSE
81.                  CALL REDUCE1
82.              ENDIF
83.              CALL SUSTITUIR
84.      C
85.      C***      CALCULA LAS TENSIONES EN LOS ELEMENTOS
86.      C
87.              CALL TENSIONES
88.      C
89.      C***      SUAVIZADO DE TENSIONES
90.      C
91.              CALL SUAV1
92.      C
93.      C***      EL TIEMPO T SE CONVIERTE EN T-1
94.      C
95.              DO IVARI=1,NGDLN*NPNOD
96.                  XTIME(IVARI)=DESPL(IVARI)
97.              ENDDO
98.          ENDDO
99.          ENDIF
100.         ENDDO
101.         CLOSE(1,STATUS='DELETE')
102.         CLOSE(3,STATUS='DELETE')
103.         CLOSE(4,STATUS='DELETE')
104.         CLOSE(8,STATUS='DELETE')
105.         CLOSE(10,STATUS='DELETE')
106.         CLOSE(29,STATUS='KEEP')
107.         CLOSE(5,STATUS='KEEP')
108.         CLOSE(6,STATUS='KEEP')
109.         STOP
110.        END

1.          SUBROUTINE BMAT(IELEM,KPGAU)
2.      C*****
3.      C
4.      C***      MATRIZ DE DEFORMACION B
5.      C

```



```

6.      C*****
7.      IMPLICIT NONE
8.      INTEGER*2      IELEM,KPGAU
9.      INTEGER*2      KCONT, ITENS, IEVAB, INODE, IDIME
10.     INCLUDE 'dtsgral.f'
11.     INCLUDE 'calculo.f'
12.     C*****
13.     KCONT=0
14.     DO ITENS=1,NTENS
15.       DO IEVAB=1,NEVAB
16.         BMATZ(ITENS,IEVAB)=0.0
17.       ENDDO
18.     ENDDO
19.     IF(NTIPO.EQ.1) THEN
20.       DO INODE=1,NNODE
21.     C
22.     C***      TRANSMISION DE CALOR
23.     C
24.       DO IDIME=1,NDIME
25.         BMATZ(IDIME,INODE)=DCART(IDIME,INODE)
26.       ENDDO
27.     ENDDO
28.     ELSE
29.       WRITE(6,900)
30.     900  FORMAT(5X,'ERROR EN SELECCION DE TIPOLOGIA DEL PROBLEMA')
31.       STOP
32.     ENDIF
33.     RETURN
34.     END

1.      SUBROUTINE CONVECC(IELEM,FTFOR,LPROP)
2.     C*****
3.     C
4.     C***      CALCULA MATRIZ DE RIGIDEZ DE CADA ELEMENTO CUANDO HAY
5.     C***      CONVECCION Y RADIACION
6.     C
7.     C*****
8.     IMPLICIT NONE
9.     INCLUDE 'dtsgral.f'
10.    INCLUDE 'data.f'
11.    INCLUDE 'gaussdat.f'
12.    INCLUDE 'calculo.f'
13.    INTEGER*2      IELEM, IGAUS, JNFRO, IFRON, IEVAB, JEVAB,
14.    .               INFRO, LNODE, IDIME, LPROP, LOCAL, INODE,
15.    .               UNOVA, KPGAU, JGAUS, JDIME
16.    REAL*4         XLONG, XFORM, ALPHA, EXISP, DLONG, FTFOR,
17.    .               D, DAREA, ZJACC, ETASP
18.    DIMENSION      XFORM(20), FTFOR(60,60), LOCAL(8), D(2),
19.    .               ZJACC(3,3)
20.    C*****
21.    C
22.    C*** CALCULO DE LA APORTACION DE RIGIDEZ POR CONVECCION/RADIACION
23.    C
24.    JNFRO=0
25.    DO IFRON=1,NFRON
26.      IF (CONVE(IFRON,1).EQ.IELEM) JNFRO=IFRON
27.    ENDDO
28.    DO IEVAB=1,NEVAB
29.      XFORM(IEVAB)=0.0
30.      DO JEVAB=1,NEVAB
31.        FTFOR(IEVAB,JEVAB)=0.0
32.      ENDDO

```

```

33.      ENDDO
34.      IF (JNFRO.NE.0) THEN
35.          DO INFRO=1,NNFRO
36.              LNODE=CONVE(JNFRO,INFRO+1)
37.              DO IDIME=1,NDIME
38.                  COREL(IDIME,INFRO)=COORD(LNODE,IDIME)
39.              ENDDO
40.          ENDDO
41.      C
42.      C*** CAMBIA DE NUMERACION GLOBAL A LOCAL
43.      C
44.          DO INFRO=1,NNFRO
45.              DO INODE=1,NNODE
46.                  IF (CONVE(JNFRO,INFRO+1).EQ.
47.                      LNODS(IELEM,INODE)) LOCAL(INFRO)=INODE
48.              ENDDO
49.          ENDDO
50.          ALPHA=PROPS(LPROP,NDIME+1)
51.          IF (NDIME.EQ.1) THEN
52.      C
53.      C*** BUCLE DE INTEGRACION NUMERICA
54.      C
55.          KPGAU=0
56.          DO IGAUS=1,NGAUS
57.              EXISP=POSPG(IGAUS)
58.              CALL FFORMA(EXISP,0,0,NDIME,NNODE,FFORM,DERIV)
59.              CALL JACOBM(IELEM,DLONG,KPGAU)
60.              DLONG=DLONG*PESPG(IGAUS)
61.              DO IEVAB=1,NEVAB
62.                  DO JEVAB=IEVAB,NEVAB
63.                      FTFOR(IEVAB,JEVAB)=FTFOR(IEVAB,JEVAB)+
64.                          FFORM(IEVAB)*FFORM(JEVAB)*DLONG*ALPHA
65.                  ENDDO
66.              ENDDO
67.          ENDDO
68.          ENDIF
69.          IF (NDIME.EQ.2) THEN
70.              UNOVA=1
71.      C
72.      C*** BUCLE DE INTEGRACION NUMERICA
73.      C
74.          DO IGAUS=1,NGAUS
75.              EXISP=POSPG(IGAUS)
76.      C
77.      C*** CALCULA FUNCIONES DE FORMA EN PUNTOS DE INTEGRACION
78.      C
79.          CALL FFORMA(EXISP,0,0,UNOVA,NNFRO,FFORM,DERIV)
80.      C
81.      C*** CALCULA EL JACOBIANO
82.      C
83.          D(1)=0.0
84.          D(2)=0.0
85.          DO INFRO=1,NNFRO
86.      C*** ARMA VECTOR DE LAS FUNCIONES DE FORMA CON CEROS EN LOS NODOS
87.      C*** NO INCLUIDOS EN EL LADO
88.          XFORM(LOCAL(INFRO))=FFORM(INFRO)
89.          DO IDIME=1,NDIME
90.              D(IDIME)=D(IDIME)+
91.                  DERIV(1,INFRO)*COREL(IDIME,LOCAL(INFRO))
92.          ENDDO
93.          ENDDO
94.          DLONG=SQRT(1+D(2)*D(2)/D(1)/D(1))*D(1)*PESPG(IGAUS)*ALPHA
95.      C

```

```

96.  C*** ARMA LA MATRIZ DE APORTACION
97.  C
98.      DO IEVAB=1,NEVAB
99.      DO JEVAB=IEVAB,NEVAB
100.      FTFOR (IEVAB,JEVAB)=FTFOR (IEVAB,JEVAB)+
101.      XFORM (IEVAB)*XFORM (JEVAB)*DLONG
102.      ENDDO
103.      ENDDO
104.      ENDDO
105.      ENDIF
106.      IF (NDIME.EQ.3) THEN
107.      UNOVA=2
108.      KPGAU=0
109.      DO IGAUS=1,NGAUS
110.      DO JGAUS=1,NGAUS
111.      KPGAU=KPGAU+1
112.      EXISP=POSPG (IGAUS)
113.      ETASP=POSPG (JGAUS)
114.  C
115.  C*** CALCULA FUNCIONES DE FORMA EN PUNTOS DE INTEGRACION
116.  C
117.      CALL FFORMA (EXISP,ETASP,0,UNOVA,NNFRO,FFORM,DERIV)
118.  C
119.  C*** CALCULA EL JACOBIANO
120.  C
121.      DO IDIME=1,3
122.      DO JDIME=1,3
123.      ZJACC (IDIME,JDIME)=0.0
124.      DO INFRO=1,NNFRO
125.      ZJACC (IDIME,JDIME)=ZJACC (IDIME,JDIME)+
126.      DERIV (IDIME,INFRO)*COREL (JDIME,LOCAL (INFRO))
127.      ENDDO
128.      ENDDO
129.      ENDDO
130.      DAREA=SQRT (
131.      (ZJACC (1,2)*ZJACC (2,3)-ZJACC (1,3)*ZJACC (2,2))**2+
132.      (ZJACC (1,3)*ZJACC (2,1)-ZJACC (1,1)*ZJACC (2,3))**2+
133.      (ZJACC (1,1)*ZJACC (2,2)-ZJACC (1,2)*ZJACC (2,1))**2)
134.      *ALPHA
135.  C*** ARMA VECTOR DE LAS FUNCIONES DE FORMA CON CEROS EN LOS NODOS
136.  C*** NO INCLUIDOS EN EL LADO
137.      DO INFRO=1,NNFRO
138.      XFORM (LOCAL (INFRO))=FFORM (INFRO)
139.      ENDDO
140.  C
141.  C*** ARMA LA MATRIZ DE APORTACION
142.  C
143.      DO IEVAB=1,NEVAB
144.      DO JEVAB=IEVAB,NEVAB
145.      FTFOR (IEVAB,JEVAB)=FTFOR (IEVAB,JEVAB)+
146.      XFORM (IEVAB)*XFORM (JEVAB)*DAREA
147.      ENDDO
148.      ENDDO
149.      ENDDO
150.      ENDDO
151.      ENDIF
152.      ENDIF
153.      RETURN
154.      END

```

```

1.          SUBROUTINE DATOS
2.          C*****
3.          C
4.          C***          LEE DATOS DE LA TIPOLOGIA DE LA MALLA, CONDICIONES DE
5.          C          CONTORNO Y PROPIEDADES DEL MATERIAL
6.          C
7.          C*****
8.          C*****data definition
9.          IMPLICIT NONE
10.         INTEGER*2          IWRTIT, IELEM, NUMEL, INODE, NNOD1, NNOD2, IPNOD,
11.         .                  IDIME, JPNOD, IPRES, IGDLN, IMATS, NUMAT, IPROP,
12.         .                  NNCIT, INCIT, IFRON, INFRO
13.         REAL*4             CITIM, X
14.         INTEGER*2          NN, MNODE, IIODR, NNON, ICONT, IIODE, IILEM, IX,
15.         .                  IINOD, IIIME
16.         INCLUDE 'dtsgral.f'
17.         INCLUDE 'data.f'
18.         DIMENSION          X(3,500), IX(24,200), NN(12)
19.         STRING FILENAME5, FILENAME
20.         INTEGER*2 ERROR
21.         C*****data definition
22.         C
23.         C
24.         C          TIPO DE PROBLEMA
25.         C
26.         C          1  TRANSMISION DE CALOR
27.         C
28.         C          TIPOS DE ELEMENTOS
29.         C
30.         C          +  ELEMENTO TRIANGULAR DE 3 NODOS
31.         C          +  ELEMENTO TRIANGULAR DE 6 NODOS
32.         C          +  ELEMENTO CUADRILATERO LAGRANGIANO DE 4 NODOS
33.         C          +  ELEMENTO CUADRILATERO LAGRANGIANO DE 9 NODOS
34.         C          +  ELEMENTO CUADRILATERO SERENDIPITO DE 8 NODOS
35.         C          +  ELEMENTO HEXAGONAL SERENDIPITO DE 20 NODOS
36.         C
37.         C          NTRAN=0 ESTACIONARIO
38.         C          NTRAN=1 TRANSITORIO (SI ALFAT=0 NO DIAGONALIZA)
39.         C          NTRAN=2 TRANSITORIO (SI ALFAT=0 DIAGONALIZA SUMA RENGLONES)
40.         C          NTRAN=3 TRANSITORIO (SI ALFAT=0 DIAGONALIZA MASA EQUIV.)
41.         C
42.         C
43.         C*** LEE PARAMETROS DE CONTROL
44.         C
45.         READ(5,900,ERR=4,IOSTAT=ERROR)
46.         .NPNOD, NELEM, NPRES, NTRAN, MNODE, NMATS, NGAUS, NDIME, IWRTIT
47.         C          ASIGNA VALORES DE DEFAULT
48.         NTIPO=1
49.         NGDLN=1
50.         NTENS=NDIME
51.         IF (NTRAN.GT.0) THEN
52.             NPROP=NDIME+3
53.         ELSE
54.             NPROP=NDIME+2
55.         ENDIF
56.         900 FORMAT(13I5)
57.         NEVAB=NGDLN*NNODE
58.         WRITE(6,905) NPNOD, NELEM, NPRES, NTRAN, NTIPO, MNODE, NGDLN, NMATS,
59.         .            NPROP, NGAUS, NDIME, NTENS, NEVAB, IWRTIT
60.         905 FORMAT(2X, ' NPNOD =', I4, 4X, ' NELEM =', I4, 4X, ' NPRES =', I4,
61.         .            4X, ' NTRAN =', I4, 4X, ' *NTIPO =', I4, //,
62.         .            2X, ' MNODE =', I4, 4X, ' *NGDLN =', I4, 4X, ' NMATS =', I4,
63.         .            4X, ' *NPROP =', I4, 4X, ' NGAUS =', I4, //,

```



```

64.          .      2X, ' NDIME =', I4, 4X, '*NTENS =', I4, 4X, '*NEVAB =', I4,
65.          .      4X, ' IWRT =', I4)
66.          C
67.          C*** LEE Y ESCRIBE CONEX. NODALES Y NUM DE PROPIEDADES DE MATERIAL
68.          C
69.          WRITE(6,910)
70.          910 FORMAT(// 'ELEMENTO', 2X, 'PROPIEDAD', 4X, 'NUMERO DE NODOS')
71.          IF(NNODE.LE.13) THEN
72.              DO IELEM=1, NELEM
73.                  READ(5,903, ERR=4, IOSTAT=ERROR)
74.                  .      NUMEL, MATNU(NUMEL), (LNODS(NUMEL, INODE), INODE=1, NNODE)
75.          903      FORMAT(15I5)
76.              ENDDO
77.          ELSE
78.              NNOD1=13
79.              NNOD2=14
80.              DO IELEM=1, NELEM
81.                  READ(5,903, ERR=4, IOSTAT=ERROR)
82.                  .      NUMEL, MATNU(NUMEL), (LNODS(NUMEL, INODE), INODE=1, NNOD1)
83.                  .      READ(5,904, ERR=4, IOSTAT=ERROR)
84.                  .      (LNODS(NUMEL, INODE), INODE=NNOD2, NNODE)
85.          904      FORMAT(10X, 15I5)
86.              ENDDO
87.          ENDIF
88.          IF(IWRIT.EQ.1) THEN
89.              IF(NNODE.LE.13) THEN
90.                  DO IELEM=1, NELEM
91.                      WRITE(6,915) IELEM, MATNU(IELEM), (LNODS(IELEM, INODE), INODE=1,
92.                          .      NNODE)
93.          915      FORMAT(1X, I5, I9, 5X, 10I5)
94.                  ENDDO
95.              ELSE
96.                  NNOD1=10
97.                  NNOD2=11
98.                  DO IELEM=1, NELEM
99.                      WRITE(6,915) IELEM, MATNU(IELEM), (LNODS(IELEM, INODE), INODE=1,
100.                          .      NNOD1)
101.                      .      WRITE(6,916) (LNODS(IELEM, INODE), INODE=NNOD2, NNODE)
102.          916      FORMAT(20X, 10I5)
103.                  ENDDO
104.              ENDIF
105.          ENDIF
106.          DO IPNOD=1, NPNOD
107.              DO IDIME=1, NDIME
108.                  COORD(IPNOD, IDIME)=0.0
109.              ENDDO
110.          ENDDO
111.          C
112.          C*** LEE Y ESCRIBE COORDENADAS NODALES
113.          C
114.              DO JPNOD=1, NPNOD
115.                  READ(5,930, ERR=4, IOSTAT=ERROR)
116.                  .      IPNOD, (COORD(IPNOD, IDIME), IDIME=1, NDIME)
117.              ENDDO
118.          930      FORMAT(I5, 5F10.5)
119.          IF(IWRIT.EQ.1) THEN
120.              WRITE(6,920)
121.          920      FORMAT(// 'COORDENADAS DE PUNTOS NODALES')
122.              WRITE(6,925)
123.          925      FORMAT(' NODO', 7X, 'X', 9X, 'Y', 9X, 'Z')
124.              DO IPNOD=1, NPNOD
125.                  WRITE(6,935) IPNOD, (COORD(IPNOD, IDIME), IDIME=1, NDIME)
126.              ENDDO

```

```

127.    935    FORMAT(1X,I5,3F10.3)
128.    ENDIF
129.    C
130.    C*** LEE Y ESCRIBE MOVIMIENTOS PRESCRITOS
131.    C
132.        DO IPRES=1,NPRES
133.            READ(5,950)
134.            .    NODPR(IPRES), (INPRE(IPRES,IGDLN),IGDLN=1,NGDLN),
135.            .    (PRESC(IPRES,IGDLN),IGDLN=1,NGDLN)
136.        ENDDO
137.    950    FORMAT(1X,I4,3X,<ngdln>I1,<ngdln>F10.5)
138.        IF(IWRIT.EQ.1) THEN
139.            WRITE(6,940)
140.    940    FORMAT(//,'NODOS RESTRINGIDOS Y PRESCRIPCIONES')
141.            WRITE(6,945)
142.    945    FORMAT('NODO',1X,'CODIGO',3X,'VALORES PRESCRITOS')
143.            DO IPRES=1,NPRES
144.                WRITE(6,950) NODPR(IPRES), (INPRE(IPRES,IGDLN),IGDLN=1,NGDLN),
145.                .    (PRESC(IPRES,IGDLN),IGDLN=1,NGDLN)
146.            ENDDO
147.        ENDIF
148.    C
149.    C*** LEE Y ESCRIBE PROPIEDADES DE LOS MATERIALES
150.    C
151.        DO IMATS=1,NMATS
152.            READ(5,990,ERR=4,IOSTAT=ERROR)
153.            .    NUMAT,(PROPS(NUMAT,IPROP),IPROP=1,NPROP)
154.        ENDDO
155.    990    FORMAT(1X,I4,5E15.5)
156.        IF(IWRIT.EQ.1) THEN
157.            WRITE(6,980)
158.    980    FORMAT(//'PROPIEDADES DE LOS MATERIALES')
159.            WRITE(6,985)
160.    985    FORMAT('NUMERO',7X,'PROPIEDADES')
161.            DO IMATS=1,NMATS
162.                WRITE(6,991) IMATS,(PROPS(IMATS,IPROP),IPROP=1,NPROP)
163.            ENDDO
164.    991    FORMAT(1X,I4,<NDIME>('    K=',E15.2),/,
165.            .    5x,' ALPHA=',E15.2,'    ROr=',E15.2,'    ROc=',E15.2,/)
166.        ENDIF
167.    C
168.    C*** LEE Y ESCRIBE DATOS DEL PROBLEMA TRANSITORIO
169.    C
170.        IF (NTRAN.GT.0) THEN
171.            READ(5,992,ERR=4,IOSTAT=ERROR) NTIME,DTIME,ALFAT
172.    992    FORMAT(I5,2E15.5)
173.            IF(IWRIT.EQ.1) THEN
174.                WRITE(6,993) NTIME,DTIME,ALFAT
175.    993    FORMAT(1X,' NUMERO DE PASOS DE TIEMPO.....',I5,/,
176.            .    1X,' INCREMENTO DE TIEMPO.....',E15.5,/,
177.            .    1X,' ALFA DE INTEGRACION TEMPORAL...',E15.5,/)
178.        ENDIF
179.    C
180.    C*** LECTURA DE LA CONDICION INICIAL DE TIEMPO
181.    C
182.        DO IPNOD=1,NPNOD
183.            XTIME(IPNOD)=0.0
184.        ENDDO
185.        READ(5,992,ERR=4,IOSTAT=ERROR) NNCIT,CITIM
186.        IF (NNCIT.EQ.NPNOD) THEN
187.            IF (CITIM.NE.0) THEN
188.                DO IPNOD=1,NPNOD
189.                    XTIME(IPNOD)=CITIM

```

```

190.         ENDDO
191.     ENDIF
192.     ELSE
193.         DO INCIT=1,NNCIT
194.             READ(5,992) IPNOD,XTIME(IPNOD)
195.         ENDDO
196.     ENDIF
197.     IF(IWRIT.EQ.1) THEN
198.         WRITE(6,994)
199.         DO IPNOD=1,NPNOD
200.             WRITE(6,995) IPNOD,XTIME(IPNOD)
201.         ENDDO
202.     994     FORMAT (1X,'CONDICIONES INICIALES DE TEMPERATURA',/,
203.                 1X,'NODO  TEMP.o')
204.     995     FORMAT (1X,I5,1X,E15.5)
205.     ENDIF
206. ENDIF
207. C
208. C*** LEE Y ESCRIBE CONDICIONES DE FRONTERA
209. C*** DE FLUJO POR CONVECCION RADIACION
210. C*** NFRON => NUMERO DE ELEMENTOS DE FRONTERA
211. C*** NNFRO => NUMERO DE NODOS DE FRONTERA
212. C*** CONVE(NFRON,1) ELEMENTO NUMERO
213. C*** CONVE(NFRON,<>1) NODOS DE FRONTERA
214. C
215.     READ(5,996,ERR=4,IOSTAT=ERROR) NFRON,NNFRO
216.     IF (IWRIT.EQ.1)
217.         WRITE(6,997) NFRON,NNFRO
218.     IF(NFRON.NE.0) THEN
219.         IF (IWRIT.EQ.1) WRITE (6,998)
220.         DO IFRON=1,NFRON
221.             READ(5,999,ERR=4,IOSTAT=ERROR)
222.             (CONVE(IFRON,INFRO),INFRO=1,NNFRO+1)
223.             IF (IWRIT.EQ.1)
224.             WRITE(6,1000) (CONVE(IFRON,INFRO),INFRO=1,NNFRO+1)
225.         ENDDO
226.     ENDIF
227.     996 FORMAT (1X,2I4)
228.     997 FORMAT(1X,'FRONTERAS CON FLUJO DE CONVECCION/RADIACION =',I4,
229.             /,'NUMERO DE NODOS POR FRONTERA =',I4)
230.     998 FORMAT (1X,'ELEMENTO  N O D O S')
231.     999 FORMAT (1X,<NNFRO+1>I4)
232.     1000 FORMAT (3X,I4,4X,<NNFRO>(I4,1X))
233. C
234. C*** ESCRITURA PARA POST-PROCESO
235. C
236.     MNODE=NNODE
237.     IF(NNODE.EQ.6) MNODE=8
238.     IIODR=(MNODE+4)
239.     IF((NNODE.EQ.6).OR.(NNODE.EQ.8).OR.(NNODE.EQ.9)) THEN
240.         NNON=MNODE
241.         IF(NNODE.EQ.6) THEN
242.             DO IELEM=1,NELEM
243.                 IX(1,IELEM)=(LNODS(IELEM,1))
244.                 IX(2,IELEM)=(LNODS(IELEM,3))
245.                 IX(3,IELEM)=(LNODS(IELEM,5))
246.                 IX(4,IELEM)=(LNODS(IELEM,1))
247.                 IX(5,IELEM)=(LNODS(IELEM,2))
248.                 IX(6,IELEM)=(LNODS(IELEM,4))
249.                 IX(7,IELEM)=(LNODS(IELEM,6))
250.                 IX(8,IELEM)=(LNODS(IELEM,1))
251.                 IX(12,IELEM)=(MATNU(IELEM))
252.             ENDDO

```

```

253.      ELSE
254.          IF (MNODE.EQ.9) NNON=8
255.          DO IELEM=1,NELEM
256.              ICONT=0
257.              DO INODE=1,NNON,2
258.                  ICONT=ICONT+1
259.                  IX(ICONT,IELEM)=(LNODS(IELEM,INODE))
260.              ENDDO
261.              DO INODE=2,NNON,2
262.                  ICONT=ICONT+1
263.                  IX(ICONT,IELEM)=(LNODS(IELEM,INODE))
264.              ENDDO
265.              IF (NNODE.EQ.9) IX(9,IELEM)=(LNODS(IELEM,9))
266.              IX(IIDR,IELEM)=(MATNU(IELEM))
267.          ENDDO
268.      ENDIF
269.      ELSE
270.          IF (NNODE.EQ.20) THEN
271.              DO IELEM=1,NELEM
272.                  IX(1,IELEM)=(LNODS(IELEM,1))
273.                  IX(2,IELEM)=(LNODS(IELEM,3))
274.                  IX(3,IELEM)=(LNODS(IELEM,5))
275.                  IX(4,IELEM)=(LNODS(IELEM,7))
276.                  IX(5,IELEM)=(LNODS(IELEM,13))
277.                  IX(6,IELEM)=(LNODS(IELEM,15))
278.                  IX(7,IELEM)=(LNODS(IELEM,17))
279.                  IX(8,IELEM)=(LNODS(IELEM,19))
280.                  IX(9,IELEM)=(LNODS(IELEM,2))
281.                  IX(10,IELEM)=(LNODS(IELEM,4))
282.                  IX(11,IELEM)=(LNODS(IELEM,6))
283.                  IX(12,IELEM)=(LNODS(IELEM,8))
284.                  IX(13,IELEM)=(LNODS(IELEM,9))
285.                  IX(14,IELEM)=(LNODS(IELEM,10))
286.                  IX(15,IELEM)=(LNODS(IELEM,11))
287.                  IX(16,IELEM)=(LNODS(IELEM,12))
288.                  IX(17,IELEM)=(LNODS(IELEM,14))
289.                  IX(18,IELEM)=(LNODS(IELEM,16))
290.                  IX(19,IELEM)=(LNODS(IELEM,18))
291.                  IX(20,IELEM)=(LNODS(IELEM,20))
292.                  IX(24,IELEM)=(MATNU(IELEM))
293.              ENDDO
294.          ELSE
295.              DO IELEM=1,NELEM
296.                  DO INODE=1,MNODE
297.                      IX(INODE,IELEM)=(LNODS(IELEM,INODE))
298.                  ENDDO
299.                  IX(IIDR,IELEM)=(MATNU(IELEM))
300.              ENDDO
301.          ENDIF
302.      ENDIF
303.      DO IPNOD=1,NPNOD
304.          DO IDIME=1,NDIME
305.              X(IDIME,IPNOD)=COORD(IPNOD,IDIME)
306.          ENDDO
307.      ENDDO
308.      NN(1)=(NPNOD)
309.      NN(2)=(NELEM)
310.      NN(3)=(NPRES)
311.      IF (NTRAN.EQ.0) THEN
312.          NN(4)=1
313.      ELSE
314.          NN(4)=NTIME
315.      ENDIF

```



```

316.      NN(5) = (NTIPO)
317.      NN(6) = (MNODE)
318.      NN(7) = (NGDLN)
319.      NN(8) = (NMATS)
320.      NN(9) = (NGAUS)
321.      NN(10) = (NDIME)
322.      NN(11) = (NTENS)
323.      NN(12) = (NEVAB)
324.      WRITE(29) (NN(IIODE), IIODE=1, 12)
325.      WRITE(29) (IILEM, (IX(IIODE, IILEM), IIODE=1, IIODR),
326.      .                               IILEM=1, NN(2))
327.      WRITE(29) (IINOD, (X(IIIME, IINOD), IIIME=1, NN(10)),
328.      .                               IINOD=1, NN(1))
329.      GO TO 444
330.      4 CLOSE(1, STATUS='DELETE')
331.      CLOSE(3, STATUS='DELETE')
332.      CLOSE(4, STATUS='DELETE')
333.      CLOSE(10, STATUS='DELETE')
334.      CLOSE(29, STATUS='KEEP')
335.      CLOSE(5, STATUS='KEEP')
336.      FILENAME='ERROR NO LOCALIZADO'
337.      IF (ERROR.EQ.-1) FILENAME='FIN PREMATUREO DEL FICHERO '
338.      IF (ERROR.EQ.41) FILENAME='NO HAY SUFICIENTE MEMORIA '
339.      IF (ERROR.EQ.62) FILENAME='CAMPO INVALIDO DE LECTURA '
340.      WRITE(6, *) 'ERROR EN LECTURA DE DATOS ', ERROR
341.      CLOSE(6, STATUS='KEEP')
342.      FILENAME5=FILENAME
343.      CALL ALERTBOX(FILENAME5)
344.      STOP
345.      444 RETURN
346.      END

1.      SUBROUTINE DBMATX
2.      C*****
3.      C
4.      C***      MULTIPLICA MATRICES D y B
5.      C
6.      C*****
7.      IMPLICIT NONE
8.      INTEGER*2      ITENS, IEVAB, JTENS
9.      INCLUDE 'dtsgral.f'
10.     INCLUDE 'calculo.f'
11.     C*****
12.     C
13.     C***      CALCULA D X B
14.     C
15.     DO ITENS=1, NTENS
16.     DO IEVAB=1, NEVAB
17.     DBMAT(ITENS, IEVAB)=0.0
18.     DO JTENS=1, NTENS
19.     DBMAT(ITENS, IEVAB)=DBMAT(ITENS, IEVAB)+
20.     .                               DMATZ(ITENS, JTENS)*BMATZ(JTENS, IEVAB)
21.     ENDDO
22.     ENDDO
23.     ENDDO
24.     RETURN
25.     END

```

```

1.      SUBROUTINE DIAGONAL
2.      C*****
3.      C
4.      C***      DIAGONALIZA LA MATRIZ DE RIGIDEZ PARA ALFAT=0
5.      C          (CASO EXPLICITO)
6.      C          NTRAN = 1 NO DIAGONALIZA
7.      C          NTRAN = 2 DIAGONALIZA POR SUMAS
8.      C          NTRAN = 3 DIAGONALIZA POR MASA EQUIVALENTE
9.      C*****
10.     IMPLICIT NONE
11.     INCLUDE 'dtsgral.f'
12.     INCLUDE 'solu.f'
13.     INTEGER*2      NEONS, IEONS, IROWS
14.     REAL*4         B, XMASS
15.     C*****
16.     NEONS=NSVAB
17.     C
18.     C*** DIAGONALIZACION POR SUMA DE RENGLONES
19.     C*
20.     IF (NTRAN.EQ.2) THEN
21.     DO IEONS=1, NEONS
22.     B=0.0
23.     DO IROWS=1, NEONS
24.     B=B+ASTIF(IEONS, IROWS)
25.     ASTIF(IEONS, IROWS)=0.0
26.     ENDDO
27.     ASTIF(IEONS, IEONS)=B
28.     ENDDO
29.     ENDIF
30.     C
31.     C*** DIAGONALIZACION POR MASA EQUIVALENTE
32.     C
33.     IF (NTRAN.EQ.3) THEN
34.     XMASS=0.0
35.     DO IEONS=1, NEONS
36.     XMASS=XMASS+ASTIF(IEONS, IEONS)
37.     ENDDO
38.     XLUMP=XLUMP/XMASS
39.     DO IEONS=1, NEONS
40.     XMASS=ASTIF(IEONS, IEONS)
41.     DO IROWS=1, NEONS
42.     ASTIF(IEONS, IROWS)=0.0
43.     ENDDO
44.     ASTIF(IEONS, IEONS)=XMASS*XLUMP
45.     ENDDO
46.     ENDIF
47.     RETURN
48.     END

1.      SUBROUTINE DMAT(LPROP)
2.      C*****
3.      C
4.      C***      MATRIZ CONSTITUTIVA D
5.      C
6.      C*****data definition
7.      IMPLICIT NONE
8.      INTEGER*2      ITENS, JTENS, LPROP
9.      INCLUDE 'dtsgral.f'
10.     INCLUDE 'data.f'
11.     INCLUDE 'calculo.f'
12.     C*****data definition
13.     DO ITENS=1, NTENS

```

```

14.         DO JTENS=1,NTENS
15.             DMATZ (ITENS ,JTENS)=0.0
16.         ENDDO
17.     ENDDO
18.     IF (NTIPO.EQ.1) THEN
19.     C
20.     C***   MATRIZ D PARA TRANSMISION DE CALOR
21.     C
22.         DO ITENS=1,NTENS
23.             DMATZ (ITENS, ITENS)=PROPS (LPROP, ITENS)
24.         ENDDO
25.     ELSE
26.         WRITE (6,900) NTIPO
27.     900   FORMAT (1X,'TIPO DE PROBLEMA ',I3,' NO IMPLEMENTADO')
28.     ENDIF
29.     RETURN
30.     END

1.         SUBROUTINE ENSAMBLA
2.     C*****
3.     C
4.     C*** ENSAMBLA LAS MATRICES DE RIGIDEZ Y LOS VECTORES DE FUERZAS
5.     C
6.     C*****
7.         IMPLICIT NONE
8.         INCLUDE 'dtsgral.f'
9.         INCLUDE 'data.f'
10.        INCLUDE 'calculo.f'
11.        INCLUDE 'solu.f'
12.        INTEGER*2  NTOTV, ITOTV, JTOTV, IPRES, NLOCA, IGDLN, NCONT, IELEM,
13.        .          INODE, NODEI, NROWS, NROWE, JNODE, NODEJ, JGDLN, NCOLS,
14.        .          NCOLE, ICARG
15.        INTEGER*2  MTOTV, JFFIX
16.        REAL*4     B, ZMASA
17.        DIMENSION JFFIX(200), ZMASA(60,60)
18.     C*****
19.        NTOTV=NPNOD*NGDLN
20.        MTOTV=(NTOTV)
21.        REWIND 1
22.        IF (NTRAN.GT.0) REWIND 4
23.        DO ITOTV=1,NTOTV
24.            ASLOD (ITOTV)=0.0
25.            DESPL (ITOTV)=0.0
26.            FIXED (ITOTV)=0
27.            IFFIX (ITOTV)=0
28.            DO JTOTV=1,NTOTV
29.                ASTIF (ITOTV, JTOTV)=0.0
30.            ENDDO
31.        ENDDO
32.     C
33.     C*** ENSAMBLA VECTOR DE FUERZAS ELEMENTALES
34.     C
35.        NSVAB=1
36.        DO IPRES=1, NPRES
37.            NLOCA=(NODPR (IPRES) -1) *NGDLN
38.            DO IGDLN=1, NGDLN
39.                NCONT=NLOCA+IGDLN
40.                IFFIX (NCONT) =INPRE (IPRES, IGDLN)
41.                FIXED (NCONT) =PRESC (IPRES, IGDLN)
42.            ENDDO
43.        ENDDO
44.        DO IELEM=1, NELEM

```

```

45.      READ(1) RIGID
46.      IF (NTRAN.GT.0) READ(4) ZMASA
47.      DO INODE=1, NNODE
48.          NODEI=LNODS (IELEM, INODE)
49.          DO IGDLN=1, NGDLN
50.              NROWS= ( (NODEI-1) *NGDLN) +IGDLN
51.              NROWE= ( (INODE-1) *NGDLN) +IGDLN
52.              ASLOD(NROWS)=ASLOD(NROWS)+CARGA (IELEM, NROWE)
53.              IF(NROWS.GT.NSVAB) NSVAB=NROWS
54.      C
55.      C***  ENSAMBLA LAS MATRICES DE RIGIDECES ELEMENTALES
56.      C
57.          DO JNODE=1, NNODE
58.              NODEJ=LNODS (IELEM, JNODE)
59.              DO JGDLN=1, NGDLN
60.                  NCOLS= (NODEJ-1) *NGDLN+JGDLN
61.                  NCOLE= (JNODE-1) *NGDLN+JGDLN
62.                  B=RIGID(NROWE, NCOLE)
63.                  IF (NTRAN.GT.0) B=B*ALFAT+ZMASA (NROWE, NCOLE)
64.                  ASTIF (NROWS, NCOLS) =ASTIF (NROWS, NCOLS) +B
65.              ENDDO
66.          ENDDO
67.      ENDDO
68.      ENDDO
69.      ENDDO
70.      DO ITOTV=1, NTOTV
71.          IF (IFFIX (ITOTV) .EQ. 0) THEN
72.              JFFIX (ITOTV) =1
73.          ELSE
74.              JFFIX (ITOTV) =0
75.          ENDIF
76.      ENDDO
77.      IF (NTRAN.GT.0) WRITE(4) ASLOD
78.      WRITE(29) MTOTV, (JFFIX (ITOTV) , ITOTV=1, NTOTV)
79.      WRITE(29) MTOTV, (ASLOD (ITOTV) , ITOTV=1, NTOTV)
80.      RETURN
81.      END

1.      SUBROUTINE FFORMA(S, T, Q, NDIME, NNODE, FFORM, DERIV)
2.      C*****
3.      C
4.      C***  FUNCIONES DEFORMA DE LOS ELEMENTOS
5.      C
6.      IMPLICIT NONE
7.      REAL*4 S, T, Q, S1, T1, S2, T2, Q2, SS, TT, QQ, ST, SST, STT, ST2,
8.          S9, T9
9.      INTEGER*2      NDIME, NNODE
10.     REAL*4          FFORM, DERIV
11.     DIMENSION      DERIV(3, 20), FFORM(20)
12.     C*****
13.     S1=S+1.0
14.     T1=T+1.0
15.     S2=S*2.0
16.     T2=T*2.0
17.     Q2=Q*2.0
18.     SS=S*S
19.     TT=T*T
20.     QQ=Q*Q
21.     ST=S*T
22.     SST=S*S*T
23.     STT=S*T*T
24.     ST2=S*T*2.0

```



```

25.      S9=S-1.0
26.      T9=T-1.0
27.      IF (NDIME.EQ.1) THEN
28.          IF (NNODE.EQ.2) THEN
29.              C
30.              C***      ELEMENTO UNIDIMENSIONAL DE DOS NODOS
31.              C
32.                  FFORM(1)=(1-S)/2.0
33.                  FFORM(2)=(1+S)/2.0
34.                  DERIV(1,1)=-1/2.0
35.                  DERIV(1,2)=1/2.0
36.              ENDIF
37.              IF (NNODE.EQ.3) THEN
38.                  C
39.                  C***      ELEMENTO UNIDIMENSIONAL DE TRES NODOS
40.                  C
41.                      FFORM(1)=(S-1)*S/2.0
42.                      FFORM(2)=(1+S)*(1-S)
43.                      FFORM(3)=(1+S)*S/2.0
44.                      DERIV(1,1)=S-1/2.0
45.                      DERIV(1,2)=-2.0*S
46.                      DERIV(1,3)=S+1/2.0
47.                  ENDIF
48.              ENDIF
49.              IF (NDIME.EQ.2) THEN
50.                  IF (NNODE.EQ.3) THEN
51.                      C
52.                      C***      ELEMENTO TRIANGULAR DE 3 NODOS
53.                      C
54.                          FFORM(1)=1.0-S-T
55.                          FFORM(2)=S
56.                          FFORM(3)=T
57.                          DERIV(1,1)=-1.0
58.                          DERIV(1,2)=1.0
59.                          DERIV(1,3)=0.0
60.                          DERIV(2,1)=-1.0
61.                          DERIV(2,2)=0.0
62.                          DERIV(2,3)=1.0
63.                      ENDIF
64.                      IF (NNODE.EQ.4) THEN
65.                          C
66.                          C***      ELEMENTO CUADRILATERO DE 4 NODOS
67.                          C
68.                              FFORM(1)=(1.0-S-T+ST)/4.0
69.                              FFORM(2)=(1.0+S-T-ST)/4.0
70.                              FFORM(3)=(1.0+S+T+ST)/4.0
71.                              FFORM(4)=(1.0-S+T-ST)/4.0
72.                              DERIV(1,1)=(-1+T)/4.0
73.                              DERIV(1,2)=(1-T)/4.0
74.                              DERIV(1,3)=(1+T)/4.0
75.                              DERIV(1,4)=(-1-T)/4.0
76.                              DERIV(2,1)=(-1+S)/4.0
77.                              DERIV(2,2)=(-1-S)/4.0
78.                              DERIV(2,3)=(1+S)/4.0
79.                              DERIV(2,4)=(1-S)/4.0
80.                          ENDIF
81.                          IF (NNODE.EQ.6) THEN
82.                              C
83.                              C***      ELEMENTO TRIANGULAR DE 6 NODOS
84.                              C
85.                                  FFORM(1)=1.0-(S+T)*3.0+(SS+TT)*2.0+ST*4.0
86.                                  FFORM(2)=(S-SS-ST)*4.0
87.                                  FFORM(3)=SS*2.0-S

```

```

88.          FFORM(4)=4.0*ST
89.          FFORM(5)=TT*2.0-T
90.          FFORM(6)=(T-TT-ST)*4.0
91.          DERIV(1,1)=-3.0+S*4.0+T*4.0
92.          DERIV(1,2)=(1.0-S*2.0-T)*4.0
93.          DERIV(1,3)=S*4.0-1.0
94.          DERIV(1,4)=T*4.0
95.          DERIV(1,5)=0.0
96.          DERIV(1,6)=-T*4.0
97.          DERIV(2,1)=-3.0+T*4.0+S*4.0
98.          DERIV(2,2)=-S*4.0
99.          DERIV(2,3)=0.0
100.         DERIV(2,4)=4.0*S
101.         DERIV(2,5)=T*4.0-1.0
102.         DERIV(2,6)=(1.0-T*2.0-S)*4.0
103.        ENDIF
104.        IF (NNODE.EQ.8) THEN
105.        C
106.        C***      ELEMENTO CUADRILATERO SERENDIPITO DE 8 NODOS
107.        C
108.          FFORM(1)=(-1.0+ST+SS+TT-SST-STT)/4.0
109.          FFORM(2)=(1.0-T-SS+SST)/2.0
110.          FFORM(3)=(-1.0-ST+SS+TT-SST+STT)/4.0
111.          FFORM(4)=(1.0+S-TT-STT)/2.0
112.          FFORM(5)=(-1.0+ST+SS+TT+SST+STT)/4.0
113.          FFORM(6)=(1.0+T-SS-SST)/2.0
114.          FFORM(7)=(-1.0-ST+SS+TT+SST-STT)/4.0
115.          FFORM(8)=(1.0-S-TT+STT)/2.0
116.          DERIV(1,1)=(T+S2-ST2-TT)/4.0
117.          DERIV(1,2)=-S+ST
118.          DERIV(1,3)=(-T+S2-ST2+TT)/4.0
119.          DERIV(1,4)=(1.0-TT)/2.0
120.          DERIV(1,5)=(T+S2+ST2+TT)/4.0
121.          DERIV(1,6)=-S-ST
122.          DERIV(1,7)=(-T+S2+ST2-TT)/4.0
123.          DERIV(1,8)=(-1.0+TT)/2.0
124.          DERIV(2,1)=(S+T2-SS-ST2)/4.0
125.          DERIV(2,2)=(-1.0+SS)/2.0
126.          DERIV(2,3)=(-S+T2-SS+ST2)/4.0
127.          DERIV(2,4)=-T-ST
128.          DERIV(2,5)=(S+T2+SS+ST2)/4.0
129.          DERIV(2,6)=(1.0-SS)/2.0
130.          DERIV(2,7)=(-S+T2+SS-ST2)/4.0
131.          DERIV(2,8)=-T+ST
132.        ENDIF
133.        IF (NNODE.EQ.9) THEN
134.        C
135.        C***      ELEMENTO CUADRILATERO LAGRANGIANO DE 9 NODOS
136.        C
137.          FFORM(1)=0.25*S9*ST*T9
138.          FFORM(2)=0.5*(1.0-SS)*T*T9
139.          FFORM(3)=0.25*S1*ST*T9
140.          FFORM(4)=0.5*S*S1*(1.0-TT)
141.          FFORM(5)=0.25*S1*ST*T1
142.          FFORM(6)=0.5*(1.0-SS)*T*T1
143.          FFORM(7)=0.25*S9*ST*T1
144.          FFORM(8)=0.5*S*S9*(1.0-TT)
145.          FFORM(9)=(1.0-SS)*(1.0-TT)
146.          DERIV(1,1)=0.25*T*T9*(-1.0+S2)
147.          DERIV(1,2)=-ST*T9
148.          DERIV(1,3)=0.25*(1.0+S2)*T*T9
149.          DERIV(1,4)=0.5*(1.0+S2)*(1.0-TT)
150.          DERIV(1,5)=0.25*(1.0+S2)*T*T1

```

```

151.          DERIV(1,6)=-ST*T1
152.          DERIV(1,7)=0.25*(-1.0+S2)*T*T1
153.          DERIV(1,8)=0.5*(-1.0+S2)*(1.0-TT)
154.          DERIV(1,9)=-S2*(1.0-TT)
155.          DERIV(2,1)=0.25*(-1.0+T2)*S*S9
156.          DERIV(2,2)=0.5*(1.0-SS)*(-1.0+T2)
157.          DERIV(2,3)=0.25*S*S1*(-1.0+T2)
158.          DERIV(2,4)=-ST*S1
159.          DERIV(2,5)=0.25*S*S1*(1.0+T2)
160.          DERIV(2,6)=0.5*(1.0-SS)*(1.0+T2)
161.          DERIV(2,7)=0.25*S*S9*(1.0+T2)
162.          DERIV(2,8)=-ST*S9
163.          DERIV(2,9)=-T2*(1.0-SS)
164.          ENDIF
165.          ENDIF
166.          IF (NDIME.EQ.3) THEN
167.              IF (NNODE.EQ.20) THEN
168.                  C
169.                  C*** ELEMENTO SOLIDO TRIDIM. HEXAGONAL SERENDIPITO DE 20 NODOS
170.                  C
171.                  FFORM(1)=(1+S)*(1-T)*(1-Q)*(S-T-Q-2)*.125
172.                  FFORM(2)=(1-TT)*(1+S)*(1-Q)*.25
173.                  FFORM(3)=(1+S)*(1+T)*(1-Q)*(S+T-Q-2)*.125
174.                  FFORM(4)=(1-SS)*(1+T)*(1-Q)*.25
175.                  FFORM(5)=(1-S)*(1+T)*(1-Q)*(-S+T-Q-2)*.125
176.                  FFORM(6)=(1-TT)*(1-S)*(1-Q)*.25
177.                  FFORM(7)=(1-S)*(1-T)*(1-Q)*(-S-T-Q-2)*.125
178.                  FFORM(8)=(1-SS)*(1-T)*(1-Q)*.25
179.                  FFORM(9)=(1-QQ)*(1+S)*(1-T)*.25
180.                  FFORM(10)=(1-QQ)*(1+S)*(1+T)*.25
181.                  FFORM(11)=(1-QQ)*(1-S)*(1+T)*.25
182.                  FFORM(12)=(1-QQ)*(1-S)*(1-T)*.25
183.                  FFORM(13)=(1+S)*(1-T)*(1+Q)*(S-T+Q-2)*.125
184.                  FFORM(14)=(1-TT)*(1+S)*(1+Q)*.25
185.                  FFORM(15)=(1+S)*(1+T)*(1+Q)*(S+T+Q-2)*.125
186.                  FFORM(16)=(1-SS)*(1+T)*(1+Q)*.25
187.                  FFORM(17)=(1-S)*(1+T)*(1+Q)*(-S+T+Q-2)*.125
188.                  FFORM(18)=(1-TT)*(1-S)*(1+Q)*.25
189.                  FFORM(19)=(1-S)*(1-T)*(1+Q)*(-S-T+Q-2)*.125
190.                  FFORM(20)=(1-SS)*(1-T)*(1+Q)*.25
191.                  DERIV(1,1)=.125*(1-T)*(1-Q)*(S2-T-Q-1)
192.                  DERIV(2,1)=.125*(1+S)*(1-Q)*(-S+T2+Q+1)
193.                  DERIV(3,1)=.125*(1+S)*(1-T)*(-S+T+Q2+1)
194.                  DERIV(1,2)=.25*(1-TT)*(1-Q)
195.                  DERIV(2,2)=.25*(-T2)*(1+S)*(1-Q)
196.                  DERIV(3,2)=.25*(TT-1)*(1+S)
197.                  DERIV(1,3)=.125*(1+T)*(1-Q)*(S2+T-Q-1)
198.                  DERIV(2,3)=.125*(1+S)*(1-Q)*(S+T2-Q-1)
199.                  DERIV(3,3)=.125*(1+S)*(1+T)*(-S-T+Q2+1)
200.                  DERIV(1,4)=.25*(-S2)*(1+T)*(1-Q)
201.                  DERIV(2,4)=.25*(1-SS)*(1-Q)
202.                  DERIV(3,4)=-.25*(1-SS)*(1+T)
203.                  DERIV(1,5)=.125*(1+T)*(1-Q)*(S2-T+Q+1)
204.                  DERIV(2,5)=.125*(1-S)*(1-Q)*(-S+T2-Q-1)
205.                  DERIV(3,5)=.125*(1-S)*(1+T)*(S-T+Q2+1)
206.                  DERIV(1,6)=-.25*(1-TT)*(1-Q)
207.                  DERIV(2,6)=-.25*(T2)*(1-S)*(1-Q)
208.                  DERIV(3,6)=-.25*(1-TT)*(1-S)
209.                  DERIV(1,7)=.125*(1-T)*(1-Q)*(S2+T+Q+1)
210.                  DERIV(2,7)=-.125*(1-S)*(1-Q)*(-S-T2-Q-1)
211.                  DERIV(3,7)=-.125*(1-S)*(1-T)*(-S-T-Q2-1)
212.                  DERIV(1,8)=-.25*(S2)*(1-T)*(1-Q)
213.                  DERIV(2,8)=-.25*(1-SS)*(1-Q)

```



```

214.      DERIV(3,8)=-.25*(1-SS)*(1-T)
215.      DERIV(1,9)=.25*(1-QQ)*(1-T)
216.      DERIV(2,9)=-.25*(1-QQ)*(1+S)
217.      DERIV(3,9)=-.25*(Q2)*(1+S)*(1-T)
218.      DERIV(1,10)=.25*(1-QQ)*(1+T)
219.      DERIV(2,10)=.25*(1-QQ)*(1+S)
220.      DERIV(3,10)=-.25*(Q2)*(1+S)*(1+T)
221.      DERIV(1,11)=-.25*(1-QQ)*(1+T)
222.      DERIV(2,11)=.25*(1-QQ)*(1-S)
223.      DERIV(3,11)=-.25*(Q2)*(1-S)*(1+T)
224.      DERIV(1,12)=-.25*(1-QQ)*(1-T)
225.      DERIV(2,12)=-.25*(1-QQ)*(1-S)
226.      DERIV(3,12)=-.25*(Q2)*(1-S)*(1-T)
227.      DERIV(1,13)=.125*(1-T)*(1+Q)*(S2-T+Q-1)
228.      DERIV(2,13)=-.125*(1+S)*(1+Q)*(S-T2+Q-1)
229.      DERIV(3,13)=.125*(1+S)*(1-T)*(S-T+Q2-1)
230.      DERIV(1,14)=.25*(1-TT)*(1+Q)
231.      DERIV(2,14)=-.25*(T2)*(1+S)*(1+Q)
232.      DERIV(3,14)=.25*(1-TT)*(1+S)
233.      DERIV(1,15)=.125*(1+T)*(1+Q)*(S2+T+Q-1)
234.      DERIV(2,15)=.125*(1+S)*(1+Q)*(S+T2+Q-1)
235.      DERIV(3,15)=.125*(1+S)*(1+T)*(S+T+Q2-1)
236.      DERIV(1,16)=-.25*(S2)*(1+T)*(1+Q)
237.      DERIV(2,16)=.25*(1-SS)*(1+Q)
238.      DERIV(3,16)=.25*(1-SS)*(1+T)
239.      DERIV(1,17)=-.125*(1+T)*(1+Q)*(-S2+T+Q-1)
240.      DERIV(2,17)=.125*(1-S)*(1+Q)*(-S+T2+Q-1)
241.      DERIV(3,17)=.125*(1-S)*(1+T)*(-S+T+Q2-1)
242.      DERIV(1,18)=-.25*(1-TT)*(1+Q)
243.      DERIV(2,18)=-.25*(T2)*(1-S)*(1+Q)
244.      DERIV(3,18)=.25*(1-TT)*(1-S)
245.      DERIV(1,19)=-.125*(1-T)*(1+Q)*(-S2-T+Q-1)
246.      DERIV(2,19)=-.125*(1-S)*(1+Q)*(-S-T2+Q-1)
247.      DERIV(3,19)=.125*(1-S)*(1-T)*(-S-T+Q2-1)
248.      DERIV(1,20)=-.25*(S2)*(1-T)*(1+Q)
249.      DERIV(2,20)=-.25*(1-SS)*(1+Q)
250.      DERIV(3,20)=.25*(1-SS)*(1-T)
251.      ENDIF
252.      ENDIF
253.      RETURN
254.      END

1.      SUBROUTINE FUERZAS
2.      C*****
3.      C
4.      C***      CALCULO DE FUERZAS NODALES EQUIVALENTES
5.      C
6.      C*****
7.      IMPLICIT NONE
8.      INCLUDE 'dtsgral.f'
9.      INCLUDE 'data.f'
10.     INTEGER*2      IELEM, IEVAB, IPUNT, IPESO, IDIST, LPROP,
11.     .              NEDGE, NODEG, NEASS, IODEG, NOPRS
12.     REAL*4         FLUJO, TEMPE, EGASP, PESPP, PRESS
13.     DIMENSION     PRESS(20,2), NOPRS(9)
14.     CHARACTER*20  TITULO
15.     C*****
16.     EGASP=0.0
17.     C
18.     C*** INICIALIZA VECTOR DE FUERZAS NODALES EQUIVALENTES
19.     C
20.     DO IELEM=1,NELEM

```



```

21.          DO IEVAB=1,NEVAB
22.            CARGA(IELEM,IEVAB)=0.0
23.          ENDDO
24.        ENDDO
25.          READ(5,900) TITULO
26.        900 FORMAT(A20)
27.          WRITE(6,905) TITULO
28.        905 FORMAT(/,1X,A20,/)
29.      C
30.      C*** LEE PARAMETROS DE TIPO DE CARGA
31.      C
32.          READ(5,910) IPUNT,IPESO,IDIST
33.          WRITE(6,911) IPUNT,IPESO,IDIST
34.        910 FORMAT(4I5)
35.        911 FORMAT (1X,'COND. DE CARGA PUNTUAL:.....',I5,/,
36.          .          1X,'COND. DE CARGA DE GENER. INTERNA:....',I5,/,
37.          .          1X,'COND. DE CALOR DISTRIBUIDO POR LADO..',I5)
38.      C
39.      C*** CARGAS PUNTUALES
40.      C
41.          IF(IPUNT.NE.0) CALL PUNTUAL
42.      C
43.      C*** CARGAS POR GENERACION INTERNA DE CALOR
44.      C
45.          IF (IPESO.NE.0) THEN
46.            DO IELEM=1,NELEM
47.              LPROP=MATNU(IELEM)
48.              PESPP=PROPS(LPROP,NPROP)
49.              IF (NTRAN.GT.0) PESPP=PROPS(LPROP,NPROP-1)
50.              IF(PESPP.NE.0.0) CALL PESOP(PESPP,IELEM)
51.            ENDDO
52.          ENDIF
53.      C
54.      C*** CARGAS POR FLUJO Y TEMPERATURA NORMAL REPARTIDAS SOBRE
55.      C      EL LADO DE UN ELEMENTO
56.      C
57.          IF(IDIST.NE.0) THEN
58.            READ(5,930) NEDGE,NODEG
59.            WRITE(6,935) NEDGE,NODEG
60.            WRITE(6,940)
61.          930  FORMAT(2I5)
62.          935  FORMAT(5X,'NUMERO DE LADOS EXPUESTOS=',I5,/,
63.          .          5X,'NUMERO DE NODOS EN LADO  ='I5,///)
64.          940  FORMAT(1X,5X,'ELEMENTO EXPUESTO (NODOS EN NUM. GLOBAL) ',/,
65.          .          'FLUJO ACTUANTE ,TEMPERATURA EXTERNA, ALFA*TEMP/FLUJO')
66.      C
67.      C*** BUCLE SOBRE CADA LADO CARGADO
68.      C
69.          DO IDIST=1,NEDGE
70.        C
71.        C*** LEE DATOS SOBRE EL LADO CARGADO Y EL VALOR DE LA CARGA
72.        C*** ELEMENTO NUMERO Y NODOS ACTUANTES
73.        C
74.            READ(5,945) NEASS,(NOPRS(IODEG),IODEG=1,NODEG)
75.            WRITE(6,950) NEASS,(NOPRS(IODEG),IODEG=1,NODEG)
76.          945  FORMAT(I5,<NODEG>I5)
77.          950  FORMAT(I10,5X,<NODEG>I5)
78.        C
79.        C*** POR CADA NODO DEL LADO UN VALOR DE FLUJO Y TEMPERATURA EXT
80.        C
81.            DO IODEG=1,NODEG
82.              READ(5,955) FLUJO,TEMPE
83.              PRESS(IODEG,1)=TEMPE*PROPS(MATNU(NEASS),NPROP-1)-FLUJO

```

```

84.          WRITE(6,956) FLUJO,TEMPE,PRESS(IODEG,1)
85.          ENDDO
86.    955    FORMAT(2F10.3)
87.    956    FORMAT(3F10.3)
88.          CALL LADO(PRESS,NEASS,NOPRS,NODEG)
89.          ENDDO
90.        ENDIF
91.        WRITE(6,970)
92.    970    FORMAT(/,1X,5X,'FUERZAS TOTALES NODALES EQUIVALENTES ',
93.          .      'ELEMENTALES',/)
94.        DO IELEM=1,NELEM
95.          WRITE(6,975) IELEM,(CARGA(IELEM,IEVAB),IEVAB=1,NEVAB)
96.        ENDDO
97.    975    FORMAT(1X,I4,5X,5E12.4,11(/,10X,5E12.4))
98.        RETURN
99.        END

1.          SUBROUTINE FUERZAS_T0(ITIME)
2.          C*****
3.          C
4.          C***          CALCULO DE FUERZAS INICIALES
5.          C
6.          C*****
7.          IMPLICIT NONE
8.          INCLUDE 'dtsgral.f'
9.          INCLUDE 'data.f'
10.         INCLUDE 'solu.f'
11.         INTEGER*2  NTOTV,ITOTV,IGDLN,NCONT,IELEM,NROWT,
12.         .          INODE,NODEI,NROWS,JNODE,NODEJ,JGDLN,NCOLS,
13.         .          NCOLE,ICARG,ITIME
14.         REAL*4     ZMASA,BSLOD,ALFAU
15.         DIMENSION ZMASA(60,60),BSLOD(200)
16.         C*****
17.         NTOTV=NPNOD*NGDLN
18.         ALFAU=1-ALFAT
19.         REWIND 4
20.         REWIND 1
21.         DO ITOTV=1,NTOTV
22.           ASLOD(ITOTV)=0.0
23.         ENDDO
24.         C
25.         C***  ENSAMBLA VECTOR DE FUERZAS ELEMENTALES
26.         C
27.         DO IELEM=1,NELEM
28.           READ(4) ZMASA
29.           READ(1) RIGID
30.           DO INODE=1,NNODE
31.             NODEI=LNODS(IELEM,INODE)
32.             DO JNODE=1,NNODE
33.               NODEJ=LNODS(IELEM,JNODE)
34.               ASLOD(NODEI)=ASLOD(NODEI)+(-RIGID(INODE,JNODE)*ALFAU+
35.             .          ZMASA(INODE,JNODE))*XTIME(NODEJ)
36.             ENDDO
37.           ENDDO
38.         ENDDO
39.         READ(4) BSLOD
40.         DO ITOTV=1,NTOTV
41.           ASLOD(ITOTV)=ASLOD(ITOTV)+BSLOD(ITOTV)
42.         ENDDO
43.         IF (ITIME.GT.1) THEN
44.           WRITE(29) NTOTV,(ASLOD(ITOTV),ITOTV=1,NTOTV)
45.         ENDIF

```

46. RETURN
47. END

```

1.      SUBROUTINE GAUSSQ (NDIME, NNODE, NGAUS)
2.      C*****
3.      C
4.      C***      CALCULA COORDENADAS Y PESOS DE LA CUADRATURA DE GAUSS
5.      C
6.      C*****data definition
7.      IMPLICIT NONE
8.      INTEGER*2      NDIME, NNODE, NGAUS
9.      INTEGER*2      KGAUS, IGASH, JGASH
10.     INCLUDE 'gaussdat.f'
11.     C*****data ,definiton
12.     C
13.     C***      ELEMENTOS TRIANGULARES
14.     C
15.     IF (NGAUS.EQ.1) THEN
16.     C
17.     C***      CUADRATURA DE 1 PUNTO
18.     C
19.         POSGT(1)=1.0/3.0
20.         POSGT(2)=1.0/3.0
21.         PESGT(1)=1.0/2.0
22.     ENDIF
23.     IF (NGAUS.EQ.3) THEN
24.     C
25.     C***      CUADRATURA DE 3 PUNTOS
26.     C
27.         POSGT(1)=1.0D0/6.0D0
28.         POSGT(2)=2.0D0/3.0D0
29.         POSGT(3)=POSGT(1)
30.         POSGT(4)=POSGT(1)
31.         POSGT(5)=POSGT(1)
32.         POSGT(6)=POSGT(2)
33.         PESGT(1)=1.0D0/6.0D0
34.         PESGT(2)=PESGT(1)
35.         PESGT(3)=PESGT(1)
36.     ENDIF
37.     IF (NGAUS.EQ.4) THEN
38.     C
39.     C***      CUADRATURA DE 4 PUNTOS
40.     C
41.         POSGT(1)=1.0D0/3.0D0
42.         POSGT(2)=0.6D0
43.         POSGT(3)=0.2D0
44.         POSGT(4)=0.2D0
45.         POSGT(5)=POSGT(1)
46.         POSGT(6)=POSGT(3)
47.         POSGT(7)=POSGT(2)
48.         POSGT(8)=POSGT(4)
49.         PESGT(1)=-27.0D0/96.0D0
50.         PESGT(2)= 25.0D0/96.0D0
51.         PESGT(3)=PESGT(2)
52.         PESGT(4)=PESGT(2)
53.     ENDIF
54.     IF (NGAUS.EQ.7) THEN
55.     C
56.     C***      CUADRATURA DE 7 PUNTOS
57.     C
58.         POSGT(1)=0.1012865073235D0
59.         POSGT(2)=0.7974269853531D0

```

```

60.          POSGT(3)=POSGT(1)
61.          POSGT(4)=0.4701420641051D0
62.          POSGT(5)=POSGT(4)
63.          POSGT(6)=0.0597158717898D0
64.          POSGT(7)=1.0D0/3.0D0
65.          POSGT(8)=POSGT(1)
66.          POSGT(9)=POSGT(1)
67.          POSGT(10)=POSGT(2)
68.          POSGT(11)=POSGT(6)
69.          POSGT(12)=POSGT(4)
70.          POSGT(13)=POSGT(4)
71.          POSGT(14)=POSGT(7)
72.          PESGT(1)=0.1259391805448D0/2.0D0
73.          PESGT(2)=PESGT(1)
74.          PESGT(3)=PESGT(1)
75.          PESGT(4)=0.1323941527885D0/2.0D0
76.          PESGT(5)=PESGT(4)
77.          PESGT(6)=PESGT(4)
78.          PESGT(7)=0.225D0/2.0D0
79.          ENDIF
80.          C
81.          C***  ELEMENTOS RECTANGULARES Y CUBICOS
82.          IF (NGAUS.EQ.1) THEN
83.          C
84.          C***  CUADRATURA DE 1 PUNTO  o 1X1 PUNTOS
85.          C
86.          POSPG(1)=0
87.          PESPG(1)=2
88.          ENDIF
89.          IF (NGAUS.EQ.2) THEN
90.          C
91.          C***  CUADRATURA DE 2 PUNTOS  o 2X2 PUNTOS  o 2X2X2 PUNTOS
92.          C
93.          POSPG(1)=-0.577350269189626
94.          PESPG(1)=1.0
95.          ENDIF
96.          IF (NGAUS.EQ.3) THEN
97.          C
98.          C***  CUADRATURA DE 3 PUNTOS  o 3X3 PUNTOS  o 3X3X3 PUNTOS
99.          C
100.         POSPG(1)=-0.774596669241483
101.         POSPG(2)=0.0
102.         PESPG(1)=0.5555555555555556
103.         PESPG(2)=0.8888888888888889
104.         ENDIF
105.         KGAUS=NGAUS/2
106.         DO IGASH=1,KGAUS
107.           JGASH=NGAUS+1-IGASH
108.           POSPG(JGASH)=-POSPG(IGASH)
109.           PESPG(JGASH)=PESPG(IGASH)
110.         ENDDO
111.         RETURN
112.         END

1.          SUBROUTINE INVER
2.          C*****
3.          C          SUBROUTINA PARA INVERSION DE MATRICES POR ELIMINACION
4.          C          GAUSIANA
5.          C          PARAMETROS:
6.          C
7.          C          A =====>  MATRIZ A INVERTIR
8.          C          B =====>  MATRIZ INVERTIDA

```



```

9.      C          N ===>  ORDEN DE LA MATRIZ
10.     C          IERROR  INDICADOR DE ERROR
11.     C*****
12.     IMPLICIT NONE
13.     INCLUDE 'solucuas.f'
14.     INTEGER*2      I, J, K
15.     REAL*4         TOLER, P
16.     DATA TOLER /1.0E-32/
17.     DO 10 I=1,NTNOD
18.         IF(XMASA(I, I).LT.TOLER) THEN
19.             DO 20 J=1,NTNOD
20.                 XMASA(I, J)=0.0
21.             20 IF(J.EQ.I) XMASA(I, I)=1.0
22.         ENDIF
23.     10 CONTINUE
24.     IERROR=0
25.     DO 30 I=1,NTNOD
26.         DO 30 J=1,NTNOD
27.             XINV(I, J)=0.0
28.             IF(I.EQ.J)THEN
29.                 XINV(I, J)=1.
30.             ENDIF
31.     30 CONTINUE
32.     DO 60 I=1,NTNOD
33.         P=XMASA(I, I)
34.         IF(ABS(P).LT.TOLER) THEN
35.             WRITE(6,900)
36.             IERROR=1
37.             RETURN
38.         ENDIF
39.         DO 50 J=1,NTNOD
40.             XINV(I, J)=XINV(I, J)/P
41.             XMASA(I, J)=XMASA(I, J)/P
42.             DO 60 K=1,NTNOD
43.                 IF(K.NE.I) THEN
44.                     P=XMASA(K, I)
45.                     DO 70 J=1,NTNOD
46.                         XMASA(K, J)=XMASA(K, J)-P*XMASA(I, J)
47.                     70 XINV(K, J)=XINV(K, J)-P*XINV(I, J)
48.                 ENDIF
49.             60 CONTINUE
50.         RETURN
51.     900 FORMAT('ERROR EN LA RUTINA DE INVERSION DE MATRICES')
52.     END

```

```

1.      SUBROUTINE JACOBM(IELEM,DJACB,KPGAU)
2.     C*****
3.     C
4.     C***      CALCULA EL JACOBIANO, SU DETERMINANTE E INVERSA,
5.     C          Y DERIVADAS CARTESIANAS DE LAS FUNCIONES DE FORMA
6.     C
7.     C*****
8.     IMPLICIT NONE
9.     INCLUDE 'dtsgral.f'
10.    INCLUDE 'gaussdat.f'
11.    INCLUDE 'calculo.f'
12.    INTEGER*2      KPGAU, IDIME, JDIME, INODE, IELEM
13.    REAL*4         XJACM, XJACI, DJACB
14.    DIMENSION      XJACM(3,3), XJACI(3,3)
15.    C*****
16.    C
17.    C***      CALCULA COORDENADAS DE LOS PUNTOS DE INTEGRACION

```

```

18. C
19.     IF (KPGAU.NE.0) THEN
20.         DO IDIME=1,NDIME
21.             CORPG (IDIME,KPGAU)=0.0
22.             DO INODE=1,NNODE
23.                 CORPG (IDIME,KPGAU)=CORPG (IDIME,KPGAU)+COREL (IDIME,INODE)
24.                 *FFORM (INODE)
25.             ENDDO
26.         ENDDO
27.     ENDIF
28.     IF (NDIME.EQ.1) THEN
29. C
30. C*** CALCULA EL JACOBIANO UNIDIMENSIONAL Y SU DETERMINANTE
31. C
32.         XJACM(1,1)=0.0
33.         XJACM(1,2)=0.0
34.         DO IDIME=1,2
35.             DO INODE=1,NNODE
36.                 XJACM(1, IDIME)=XJACM(1, IDIME)+
37.                 DERIV(1, INODE) *COREL (IDIME, INODE)
38.             ENDDO
39.         ENDDO
40.         DJACB=SQRT(1+XJACM(1,2)*XJACM(1,2)/XJACM(1,1)/XJACM(1,1))
41.         *XJACM(1,1)
42.         IF(DJACB.LE.0.0) THEN
43.             WRITE (6,900) IELEM
44.             STOP
45.         ENDIF
46.         XJACI(1,1)=1/XJACM(1,1)
47.     ELSE
48. C
49. C*** CALCULA MATRIZ JACOBIANO 1D 2D Y 3D
50. C
51.         DO IDIME=1,NDIME
52.             DO JDIME=1,NDIME
53.                 XJACM(IDIME,JDIME)=0.0
54.                 DO INODE=1,NNODE
55.                     XJACM(IDIME,JDIME)=XJACM(IDIME,JDIME)+
56.                     DERIV(IDIME,INODE) *COREL (JDIME, INODE)
57.                 ENDDO
58.             ENDDO
59.         ENDDO
60.     ENDIF
61.     IF (NDIME.EQ.2) THEN
62. C
63. C*** CALCULA EL DETERMINANTE Y LA INVERSA DEL JACOBIANO EN 2D
64. C
65.         DJACB=XJACM(1,1)*XJACM(2,2)-XJACM(1,2)*XJACM(2,1)
66.         IF(DJACB.LE.0.0) THEN
67.             WRITE (6,900) IELEM
68.             STOP
69.         ENDIF
70.         XJACI(1,1)=XJACM(2,2)/DJACB
71.         XJACI(2,2)=XJACM(1,1)/DJACB
72.         XJACI(1,2)=-XJACM(1,2)/DJACB
73.         XJACI(2,1)=-XJACM(2,1)/DJACB
74.     ENDIF
75.     IF (NDIME.EQ.3) THEN
76. C
77. C*** CALCULA EL DETERMINANTE Y LA INVERSA DEL JACOBIANO EN 3D
78. C
79.         DJACB=XJACM(1,1)*XJACM(2,2)*XJACM(3,3)+XJACM(1,3)*XJACM
80.         (2,1)*XJACM(3,2)+XJACM(3,1)*XJACM(1,2)*XJACM(2,3)-

```

```

81.      .      XJACM(3,1)*XJACM(2,2)*XJACM(1,3)-XJACM(3,3)*XJACM
82.      .      (1,2)*XJACM(2,1)-XJACM(1,1)*XJACM(2,3)*XJACM(3,2)
83.      IF(DJACB.LE.0.0) THEN
84.      WRITE (6,900) IELEM
85.      900    FORMAT(/, 'PROGRAMA DETENIDO EN SUBROUTINA JACOBM',/,11X,
86.      .      ' DETERMINANTE DEL JACOBIANO CERO O NEGATIVO',/,10X,
87.      .      ' EN ELEMENTO NUMERO',I5)
88.      STOP
89.      ENDIF
90.      XJACI(1,1) = (XJACM(2,2)*XJACM(3,3)-XJACM(3,2)*XJACM(2,3))/DJACB
91.      XJACI(1,2) = -(XJACM(1,2)*XJACM(3,3)-XJACM(1,3)*XJACM(3,2))/DJACB
92.      XJACI(1,3) = (XJACM(1,2)*XJACM(2,3)-XJACM(2,2)*XJACM(1,3))/DJACB
93.      XJACI(2,1) = -(XJACM(2,1)*XJACM(3,3)-XJACM(3,1)*XJACM(2,3))/DJACB
94.      XJACI(2,2) = (XJACM(1,1)*XJACM(3,3)-XJACM(1,3)*XJACM(3,1))/DJACB
95.      XJACI(2,3) = -(XJACM(1,1)*XJACM(2,3)-XJACM(2,1)*XJACM(1,3))/DJACB
96.      XJACI(3,1) = (XJACM(2,1)*XJACM(3,2)-XJACM(3,1)*XJACM(2,2))/DJACB
97.      XJACI(3,2) = -(XJACM(1,1)*XJACM(3,2)-XJACM(3,1)*XJACM(1,2))/DJACB
98.      XJACI(3,3) = (XJACM(1,1)*XJACM(2,2)-XJACM(1,2)*XJACM(2,1))/DJACB
99.      ENDIF
100.     C
101.     C***  CALCULA LAS DERIVADAS CARTESIANAS DE LAS FUNCIONES DE FORMA
102.     C
103.     DO IDIME=1,NDIME
104.     DO INODE=1,NNODE
105.     DCART(IDIME,INODE)=0.0
106.     DO JDIME=1,NDIME
107.     DCART(IDIME,INODE)=DCART(IDIME,INODE)+XJACI(IDIME,JDIME)*
108.     .      DERIV(JDIME,INODE)
109.     ENDDO
110.     ENDDO
111.     RETURN
112.     RETURN
113.     END

1.      SUBROUTINE LADO(PRESS,NEASS,NOPRS,NODEG)
2.      C*****
3.      C
4.      C***  CALCULO DE FUERZAS NODALES EQUIVALENTES
5.      C      SOBRE UN LADO DEL ELEMENTO
6.      C*****
7.      IMPLICIT NONE
8.      INCLUDE 'dtsgral.f'
9.      INCLUDE 'data.f'
10.     INCLUDE 'gaussdat.f'
11.     INCLUDE 'calculof.f'
12.     INTEGER*2      IODEG,NODEG,NEASS,NOPRS,LNODE,IDIME,IGAUS,
13.     .      IGDLN,JNODE,INODE,NLOCA,KOUNT,KNODE,NCONT,
14.     .      UNOVA,KPGAU,LPROP,JGAUS,JDIME
15.     REAL*4      PRESS,EXISP,CORD,PGASH,DGASH,DAREA,PXCOM,
16.     .      D,DLONG,ETASP,ZJACC
17.     DIMENSION   PRESS(20,2),NOPRS(9),D(2),ZJACC(3,3),
18.     .      PXCOM(20)
19.     C*****
20.     UNOVA=1
21.     DO IODEG=1,NODEG
22.     LNODE=NOPRS(IODEG)
23.     DO IDIME=1,NDIME
24.     COREL(IDIME,IODEG)=COORD(LNODE,IDIME)
25.     ENDDO
26.     ENDDO
27.     IF (NDIME.EQ.1) THEN
28.     C

```

```

29. C*** ASOCIA TEMPERATURA EXTERNA Y EL FLUJO ACTUANTE A UN NODO EN
30. C*** FORMA DIRECTA
31. C
32.         JNODE=0
33.         DO INODE=1,NNODE
34.             NLOCA=LNODS (NEASS, INODE)
35.             IF (NLOCA.EQ.NOPRS (1)) JNODE=INODE
36.         ENDDO
37.         CARGA (NEASS, JNODE) =CARGA (NEASS, JNODE) +PRESS (1, 1)
38.     ENDIF
39.     IF (NDIME.EQ.2) THEN
40. C
41. C*** BUCLE DE INTEGRACION NUMERICA
42. C
43.         DO IGAUS=1,NGAUS
44.             EXISP=POSPG (IG AUS)
45. C
46. C*** CALCULA FUNCIONES DE FORMA EN PUNTOS DE INTEGRACION
47. C
48.             CALL FFORMA (EXISP, 0, 0, UNOVA, NODEG, FFORM, DERIV)
49. C
50. C*** CALCULA EL JACOBIANO
51. C
52.             D (1) =0.0
53.             D (2) =0.0
54.             DO IO DEG=1, NODEG
55.                 DO IDIME=1, NDIME
56.                     D (IDIME) =D (IDIME) +
57.                         DERIV (1, IO DEG) *COREL (IDIME, IO DEG)
58.                 ENDDO
59.             ENDDO
60.             DLONG=SQRT (1+D (2) *D (2) /D (1) /D (1)) *D (1) *PESPG (IG AUS)
61. C
62. C*** CALCULA LAS FUERZAS NODALES EQUIVALENTES
63. C
64.             DO IO DEG=1, NODEG
65.                 PXC COM (IO DEG) = PRESS (IO DEG, 1) *FFORM (IO DEG) *DLONG
66.             ENDDO
67. C
68. C*** ASOCIA LAS FUERZAS NODALES EQUIVALENTES DE BORDE A UN ELEMENTO
69. C
70.             JNODE=0
71.             DO INODE=1,NNODE
72.                 NLOCA=LNODS (NEASS, INODE)
73.                 IF (NLOCA.EQ.NOPRS (1)) JNODE=INODE
74.             ENDDO
75.             INODE=JNODE
76.             JNODE=JNODE+NODEG-1
77.             KOUNT=0
78.             DO KNODE=INODE, JNODE
79.                 KOUNT=KOUNT+1
80.                 NCONT=(KNODE-1) *NGDLN+1
81.                 IF (KNODE.GT.NNODE.OR.(NNODE.EQ.9.AND.KNODE.EQ.9)) NCONT=1
82.                 CARGA (NEASS, NCONT) =CARGA (NEASS, NCONT) +PXC COM (KNODE-INODE+1)
83.             ENDDO
84.         ENDDO
85.     ENDIF
86.     IF (NDIME.EQ.3) THEN
87.         UNOVA=2
88.         KPGAU=0
89.         DO IGAUS=1,NGAUS
90.             DO JGAUS=1,NGAUS
91.                 KPGAU=KPGAU+1

```



```

92.             EXISP=POSPG(IGAUS)
93.             ETASP=POSPG(JGAUS)
94.     C
95.     C*** CALCULA FUNCIONES DE FORMA EN PUNTOS DE INTEGRACION
96.     C
97.             CALL FFORMA(EXISP,ETASP,0,UNOVA,NNFRO,FFORM,DERIV)
98.     C
99.     C*** CALCULA EL JACOBIANO
100.    C
101.            DO IDIME=1,3
102.            DO JDIME=1,3
103.            ZJACC(IDIME,JDIME)=0.0
104.            DO IODEG=1,NODEG
105.            ZJACC(IDIME,JDIME)=ZJACC(IDIME,JDIME)+
106.            .                DERIV(IDIME,IODEG)*COREL(JDIME,IODEG)
107.            ENDDO
108.            ENDDO
109.            ENDDO
110.            DAREA=SQRT(
111.            .                (ZJACC(1,2)*ZJACC(2,3)-ZJACC(1,3)*ZJACC(2,2))**2+
112.            .                (ZJACC(1,3)*ZJACC(2,1)-ZJACC(1,1)*ZJACC(2,3))**2+
113.            .                (ZJACC(1,1)*ZJACC(2,2)-ZJACC(1,2)*ZJACC(2,1))**2)
114.    C
115.    C*** CALCULA LAS FUERZAS NODALES EQUIVALENTES
116.    C
117.            DO IODEG=1,NODEG
118.            PXCOSM(IODEG)=PRESS(IODEG,1)*FFORM(IODEG)*DAREA
119.            ENDDO
120.    C
121.    C*** ASOCIA LAS FUERZAS NODALES EQUIVALENTES DE BORDE A UN ELEMENTO
122.    C
123.            JNODE=0
124.            DO INODE=1,NNODE
125.            NLOCA=LNODS(NEASS,INODE)
126.            IF(NLOCA.EQ.NOPRS(1)) JNODE=INODE
127.            ENDDO
128.            INODE=JNODE
129.            JNODE=JNODE+NODEG-1
130.            KOUNT=0
131.            DO KNODE=INODE,JNODE
132.            KOUNT=KOUNT+1
133.            NCONT=(KNODE-1)*NGDLN+1
134.            IF(KNODE.GT.NNODE.OR.(NNODE.EQ.9.AND.KNODE.EQ.9)) NCONT=1
135.            CARGA(NEASS,NCONT)=CARGA(NEASS,NCONT)+PXCOSM(KNODE-INODE+1)
136.            ENDDO
137.            ENDDO
138.            ENDDO
139.            ENDIF
140.            RETURN
141.            END

1.     SUBROUTINE PESOP(PESPP,IELEM)
2.     C*****
3.     C
4.     C***     CALCULO DE FUERZAS NODALES EQUIVALENTES
5.     C         POR GENERACION INTERNA DE CALOR (EQ. A PESO PROPIO)
6.     C*****
7.     IMPLICIT NONE
8.     INCLUDE 'dtsgral.f'
9.     INCLUDE 'data.f'
10.    INCLUDE 'gaussdat.f'
11.    INCLUDE 'calcul0.f'

```

```

12.      INTEGER*2          INODE, LNODE, IDIME, IELEM, IGAUS, JGAUS, KPGAU,
13.      .                  KGAUS, MCONT
14.      REAL*4            EXISP, ETASP, EGASP, DVOLU, PESPP, DJACB, EGISP
15.      C*****
16.      C
17.      C*** CARGAS POR GENERACION INTERNA DE CALOR
18.      C
19.      C*** CALCULA COORDENADAS DE LOS PUNTOS NODALES
20.      C
21.      DO INODE=1, NNODE
22.          LNODE=LNODS ( IELEM, INODE)
23.          DO IDIME=1, NDIME
24.              COREL ( IDIME, INODE) =COORD ( LNODE, IDIME)
25.          ENDDO
26.      ENDDO
27.      IF ( NDIME.EQ.1) THEN
28.          DO IGAUS=1, NGAUS
29.              EXISP=POSPG ( IGAUS)
30.      C
31.      C*** CALCULA FUNCIONES DE FORMA EN LOS PUNTOS DE INTEGRACION Y
32.      C EL VOLUMEN ELEMENTAL
33.      C
34.          CALL FFORMA ( EXISP, 0, 0, NDIME, NNODE, FFORM, DERIV)
35.          KPGAU=1
36.          CALL JACOBM ( IELEM, DJACB, KPGAU)
37.          DVOLU=DJACB*PESPP ( IGAUS)
38.      C
39.      C*** CALCULA LAS CARGAS NODALES EQUIVALENTES
40.      C
41.          DO INODE=1, NNODE
42.              MCONT=INODE
43.              CARGA ( IELEM, MCONT) =CARGA ( IELEM, MCONT) +
44.              .                  PESPP*FFORM ( INODE) *DVOLU
45.          ENDDO
46.      ENDDO
47.      ENDIF
48.      IF ( NDIME.EQ.2) THEN
49.          IF ( NNODE.EQ.4.OR.NNODE.EQ.8.OR.NNODE.EQ.9) THEN
50.      C
51.      C*** BUCLE DE INTEGRACION NUMERICA PARA ELEMENTOS CUADRILATEROS
52.      C
53.          DO IGAUS=1, NGAUS
54.              DO JGAUS=1, NGAUS
55.                  EXISP=POSPG ( IGAUS)
56.                  ETASP=POSPG ( JGAUS)
57.      C
58.      C*** CALCULA FUNCIONES DE FORMA EN LOS PUNTOS DE INTEGRACION Y
59.      C EL VOLUMEN ELEMENTAL
60.      C
61.          CALL FFORMA ( EXISP, ETASP, EGASP, NDIME, NNODE, FFORM, DERIV)
62.          KPGAU=1
63.          CALL JACOBM ( IELEM, DJACB, KPGAU)
64.          DVOLU=DJACB*PESPP ( IGAUS) *PESPP ( JGAUS)
65.      C
66.      C*** CALCULA LAS CARGAS NODALES EQUIV PARA CADA ELEMENTO CUADRILATERO
67.      C
68.          DO INODE=1, NNODE
69.              MCONT=INODE
70.              CARGA ( IELEM, MCONT) =CARGA ( IELEM, MCONT) +
71.              .                  PESPP*FFORM ( INODE) *DVOLU
72.          ENDDO
73.      ENDDO
74.      ENDDO

```

```
75.          ENDIF
76.          IF (NNODE.EQ.3.OR.NNODE.EQ.6) THEN
77.      C
78.      C***      BUCLE DE INTEGRACION NUMERICA PARA ELEMENTOS TRIANGULARES
79.      C
80.          DO IGAUS=1,NGAUS
81.              EXISP=POSGT(IGAUS)
82.              ETASP=POSGT(NGAUS+IGAUS)
83.      C
84.      C*** CALCULA FUNCIONES DE FORMA EN LOS PUNTOS DE INTEGRACION Y
85.      C      EL VOLUMEN ELEMENTAL
86.      C
87.          CALL FFORMA(EXISP,ETASP,EGASP,NDIME,NNODE,FFORM,DERIV)
88.          KPGAU=1
89.          CALL JACOBM(IELEM,DJACB,KPGAU)
90.          DVOLU=DJACB*PESGT(IGAUS)
91.      C
92.      C*** CALCULA LAS CARGAS NODALES EQUIV PARA CADA ELEMENTO TRIANGULAR
93.      C
94.          DO INODE=1,NNODE
95.              MCONT=INODE
96.              CARGA(IELEM,MCONT)=CARGA(IELEM,MCONT)+
97.                  PESPP*FFORM(INODE)*DVOLU
98.          ENDDO
99.          ENDDO
100.         ENDIF
101.     ENDIF
102.     C
103.     C*** BUCLE DE INTEGRACION NUMERICA PARA ELEMENTOS DE SOLIDO
104.     C
105.         IF(NDIME.EQ.3) THEN
106.             KPGAU=0
107.             DO IGAUS=1,NGAUS
108.                 DO JGAUS=1,NGAUS
109.                     DO KGAUS=1,NGAUS
110.                         KPGAU=KPGAU+1
111.                         EXISP=POSPG(IGAUS)
112.                         ETASP=POSPG(JGAUS)
113.                         EGISP=POSPG(KGAUS)
114.                     C
115.                     C*** CALCULA FUNCIONES DE FFORM EN LOS PUNTOS DE INTEGRACION Y
116.                     C      EL VOLUMEN ELEMENTAL
117.                     C
118.                         CALL FFORMA(EXISP,ETASP,EGISP,NDIME,NNODE,FFORM,DERIV)
119.                         CALL JACOBM(IELEM,DJACB,KPGAU)
120.                         DVOLU=DJACB*PESPG(IGAUS)*PESPG(JGAUS)*PESPG(KGAUS)
121.                     C
122.                     C*** CALCULA LAS CARGAS EQUIV NODALES PARA CADA ELEMENTO CUADRATICO
123.                     C
124.                         DO INODE=1,NNODE
125.                             MCONT=(INODE-1)*NGDLN+1
126.                             CARGA(IELEM,MCONT)=CARGA(IELEM,MCONT)+
127.                                 PESPP*FFORM(INODE)*DVOLU
128.                         ENDDO
129.                     ENDDO
130.                 ENDDO
131.             ENDDO
132.         ENDIF
133.     RETURN
134.     END
```

```

1.      SUBROUTINE PUNTUAL
2.      C*****
3.      C
4.      C***      CALCULO DE FUERZAS NODALES EQUIVALENTES PUNTUALES
5.      C
6.      C*****
7.      IMPLICIT NONE
8.      INTEGER*2      LODPT, IELEM, JELEM, INODE, JNODE, NLOCA, IGDLN,
9.      .              NCONT
10.     REAL*4      PNODT
11.     DIMENSION    PNODT(3)
12.     INCLUDE 'dtsgral.f'
13.     INCLUDE 'data.f'
14.     INCLUDE 'calculo.f'
15.     C*****
16.     C
17.     C*** LEE CARGAS PUNTUALES
18.     C
19.     20 READ(5,900) LODPT, (PNODT(IGDLN), IGDLN=1, NGDLN)
20.     WRITE(6,900) LODPT, (PNODT(IGDLN), IGDLN=1, NGDLN)
21.     900 FORMAT(I5,5F10.3)
22.     C
23.     C*** ASOCIA LAS CARGAS PUNTUALES NODALES CON UN ELEMENTO
24.     C
25.     DO IELEM=1,NELEM
26.     DO INODE=1,NNODE
27.     NLOCA=LNODS(IELEM,INODE)
28.     IF(LODPT.EQ.NLOCA) THEN
29.     JNODE=INODE
30.     JELEM=IELEM
31.     ENDIF
32.     ENDDO
33.     ENDDO
34.     DO IGDLN=1,NGDLN
35.     NCONT=(JNODE-1)*NGDLN+IGDLN
36.     CARGA(JELEM,NCONT)=PNODT(IGDLN)
37.     ENDDO
38.     IF(LODPT.LT.NPNOD) GO TO 20
39.     RETURN
40.     END

```

```

1.      SUBROUTINE REDUCE
2.      C*****
3.      C
4.      C***      REDUCE EL SISTEMA DE ECUACIONES GLOBALES POR ELIMINACION
5.      C              GAUSSIANA DIRECTA
6.      C
7.      C*****
8.      IMPLICIT NONE
9.      INCLUDE 'solu.f'
10.     INTEGER*2      NEONS, IEONS, IROWS, IEON1, ICOLS
11.     REAL*4      B, PIVOT, FACTR
12.     C*****
13.     NEONS=NSVAB
14.     DO 50 IEONS=1,NEONS
15.     IF(IFFIX(IEONS).EQ.1) THEN
16.     C
17.     C*** AJUSTA EL VECTOR DE CARGAS PARA DESPLAZAMIENTOS PRESCRITOS
18.     C
19.     DO IROWS=IEONS,NEONS
20.     B=ASTIF(IROWS,IEONS)*FIXED(IEONS)
21.     ASLOD(IROWS)=ASLOD(IROWS)-B

```



```

22.          ENDDO
23.          ELSE
24.      C
25.      C***  REDUCE ECUACIONES
26.      C
27.          PIVOT=ASTIF(IEONS, IEONS)
28.          IF(ABS(PIVOT).LT.1.0E-10) THEN
29.              WRITE(6,900) PIVOT, IEONS
30.              STOP
31.          ENDIF
32.          IF(IEONS.EQ.NEONS) GOTO 50
33.          IEON1=IEONS+1
34.          DO IROWS=IEON1, NEONS
35.              FACTR=ASTIF(IROWS, IEONS)/PIVOT
36.              IF(FACTR.NE.0.0) THEN
37.                  DO ICOLS=IEONS, NEONS
38.                      ASTIF(IROWS, ICOLS)=ASTIF(IROWS, ICOLS) -
39.                          FACTR*ASTIF(IEONS, ICOLS)
40.                  ENDDO
41.                  ASLOD(IROWS)=ASLOD(IROWS) -FACTR*ASLOD(IEONS)
42.                  ASTIF(IROWS, IEONS)=FACTR
43.              ENDIF
44.          ENDDO
45.      ENDIF
46.      50 CONTINUE
47.      900 FORMAT(5X, 'PIVOTE INCORRECTO=', E20.6, 5X, 'ECUACION No.=', I5)
48.      RETURN
49.      END

```

```

1.      SUBROUTINE REDUCE1
2.      C*****
3.      C
4.      C***  REDUCE EL SISTEMA DE ECUACIONES GLOBALES POR ELIMINACION
5.      C          GAUSSIANA DIRECTA
6.      C
7.      C*****
8.      IMPLICIT NONE
9.      INCLUDE 'solu.f'
10.     INTEGER*2      NEONS, IEONS, IROWS, IEON1, ICOLS
11.     REAL*4         B, PIVOT, FACTR
12.     C*****
13.     NEONS=NSVAB
14.     DO 50 IEONS=1, NEONS
15.         IF(IFFIX(IEONS).EQ.1) THEN
16.             C
17.             C***  AJUSTA EL VECTOR DE CARGAS PARA DESPLAZAMIENTOS PRESCRITOS
18.             C
19.                 DO IROWS=IEONS, NEONS
20.                     B=ASTIF(IROWS, IEONS)*FIXED(IEONS)
21.                     ASLOD(IROWS)=ASLOD(IROWS) -B
22.                 ENDDO
23.             ELSE
24.                 C
25.                 C***  REDUCE ECUACIONES
26.                 C
27.                     IF(IEONS.EQ.NEONS) GOTO 50
28.                     IEON1=IEONS+1
29.                     DO IROWS=IEON1, NEONS
30.                         FACTR=ASTIF(IROWS, IEONS)
31.                         IF(FACTR.NE.0.0) THEN
32.                             ASLOD(IROWS)=ASLOD(IROWS) -FACTR*ASLOD(IEONS)
33.                         ENDIF

```

```

34.          ENDDO
35.          ENDIF
36.    50 CONTINUE
37.          RETURN
38.          END

1.          SUBROUTINE RIGIMAT
2. C*****
3. C
4. C***          CALCULA MATRIZ DE RIGIDEZ DE CADA ELEMENTO
5. C
6. C*****
7.          IMPLICIT NONE
8.          INTEGER*2          IELEM, INODE, IDIME, IEVAB, JEVAB, KPGAU, IGAUS,
9.          .                  JGAUS, ITENS, KGAUS, LPROP, LNODE
10.         REAL*4            EGASP, RIGID, EXISP, ETASP, DJACB, DVOLU,
11.         .                  FTFOR, ZMASA, ROCEE, PATOP, XLONG
12.         INCLUDE 'dtsgral.f'
13.         INCLUDE 'data.f'
14.         INCLUDE 'calculo.f'
15.         INCLUDE 'gaussdat.f'
16.         DIMENSION RIGID(60,60), FTFOR(60,60), ZMASA(60,60)
17. C*****
18.         IF (NTIPO.NE.1) THEN
19.             WRITE(6,900)
20.     900    FORMAT(1X, 'ERROR EN LA SELECCION DE TIPO DE ANALISIS')
21.             STOP
22.         ENDIF
23.         EGASP=0.0
24.         XLUMP=0.0
25. C
26. C*** BUCLE SOBRE CADA ELEMENTO
27. C
28.         DO IELEM=1, NELEM
29.             LPROP=ABS (MATNU ( IELEM ) )
30. C
31. C***          CALCULA LAS COORDENADAS DE LOS PUNTOS NODALES DEL ELEMENTO
32. C
33.             DO INODE=1, NNODE
34.                 LNODE=LNODS ( IELEM, INODE )
35.                 DO IDIME=1, NDIME
36.                     COREL ( IDIME, INODE ) =COORD ( LNODE, IDIME )
37.                 ENDDO
38.             ENDDO
39. C
40. C*** CALCULA LA MATRIZ D
41. C
42.             CALL DMAT(LPROP)
43. C
44. C*** INICIALIZA LA MATRIZ DE RIGIDEZ ELEMENTAL
45. C
46.             DO IEVAB=1, NEVAB
47.                 DO JEVAB=1, NEVAB
48.                     RIGID ( IEVAB, JEVAB ) =0.0
49.                 ENDDO
50.             ENDDO
51. C
52. C*** INICIALIZA LA MATRIZ DE MASA ELEMENTAL
53. C
54.             IF (NTRAN.GT.0) THEN
55.                 DO IEVAB=1, NEVAB
56.                     DO JEVAB=1, NEVAB

```

```

57.             ZMASA ( IEVAB, JEVAB ) = 0.0
58.             ENDDO
59.             ENDDO
60.             ROCEE = PROPS ( LPROP, NPROP ) / DTIME
61.             ENDIF
62. C
63. C*** AQUI EMPIEZA
64.             KPGAU = 0
65.             IF ( NDIME.EQ.1 ) THEN
66.                 DO IGAUS = 1, NGAUS
67.                     KPGAU = KPGAU + 1
68.                     EXISP = POSPG ( IGAUS )
69.                     CALL FFORMA ( EXISP, 0, 0, NDIME, NNODE, FFORM, DERIV )
70.                     CALL JACOBM ( IELEM, DJACB, KPGAU )
71.                     DVOLU = DJACB * PESPG ( IGAUS )
72.                     CALL BMAT ( IELEM, KPGAU )
73.                     CALL DBMATX
74.                     DO IEVAB = 1, NEVAB
75.                         DO JEVAB = IEVAB, NEVAB
76.                             DO ITENS = 1, NTENS
77.                                 RIGID ( IEVAB, JEVAB ) = RIGID ( IEVAB, JEVAB ) +
78.                                     BMATZ ( ITENS, IEVAB ) * DBMAT ( ITENS, JEVAB ) * DVOLU
79.                             ENDDO
80.                             IF ( NTRAN.GT.0 ) THEN
81.                                 PATOP = FFORM ( IEVAB ) * FFORM ( JEVAB ) * DVOLU * ROCEE
82.                                 ZMASA ( IEVAB, JEVAB ) = ZMASA ( IEVAB, JEVAB ) + PATOP
83.                                 IF ( NTRAN.EQ.3 ) XLUMP = XLUMP + DVOLU * ROCEE
84.                             ENDIF
85.                         ENDDO
86.                     ENDDO
87.                 DO ITENS = 1, NTENS
88.                     DO IEVAB = 1, NEVAB
89.                         TENSZ ( ITENS, IEVAB, KPGAU ) = DBMAT ( ITENS, IEVAB )
90.                     ENDDO
91.                 ENDDO
92.             ENDDO
93.             ENDIF
94.             IF ( NDIME.EQ.2 ) THEN
95.                 IF ( ( NNODE.EQ.4 ) .OR. ( NNODE.EQ.8 ) .OR. ( NNODE.EQ.9 ) ) THEN
96. C
97. C*** ELEMENTOS CUADRILATEROS
98. C
99.                 DO IGAUS = 1, NGAUS
100.                    DO JGAUS = 1, NGAUS
101.                        KPGAU = KPGAU + 1
102.                        EXISP = POSPG ( IGAUS )
103.                        ETASP = POSPG ( JGAUS )
104.                        CALL FFORMA ( EXISP, ETASP, EGASP, NDIME, NNODE, FFORM, DERIV )
105.                        CALL JACOBM ( IELEM, DJACB, KPGAU )
106.                        DVOLU = DJACB * PESPG ( IGAUS ) * PESPG ( JGAUS )
107.                        CALL BMAT ( IELEM, KPGAU )
108.                        CALL DBMATX
109.                        DO IEVAB = 1, NEVAB
110.                            DO JEVAB = IEVAB, NEVAB
111.                                DO ITENS = 1, NTENS
112.                                    RIGID ( IEVAB, JEVAB ) = RIGID ( IEVAB, JEVAB ) +
113.                                        BMATZ ( ITENS, IEVAB ) * DBMAT ( ITENS, JEVAB ) * DVOLU
114.                                ENDDO
115.                                IF ( NTRAN.GT.0 ) THEN
116.                                    PATOP = FFORM ( IEVAB ) * FFORM ( JEVAB ) * DVOLU * ROCEE
117.                                    ZMASA ( IEVAB, JEVAB ) = ZMASA ( IEVAB, JEVAB ) + PATOP
118.                                    IF ( NTRAN.EQ.3 ) XLUMP = XLUMP + DVOLU * ROCEE
119.                                ENDIF

```

```

120.          ENDDO
121.          ENDDO
122.          DO ITENS=1,NTENS
123.            DO IEVAB=1,NEVAB
124.              TENSZ (ITENS,IEVAB,KPGAU) =DBMAT (ITENS,IEVAB)
125.            ENDDO
126.          ENDDO
127.        ENDDO
128.      ENDDO
129.    ENDIF
130.    IF ((NNODE.EQ.3).OR.(NNODE.EQ.6)) THEN
131.  C
132.  C*** ELEMENTOS TRIANGULARES
133.  C
134.        DO IGAUS=1,NGAUS
135.          KPGAU=KPGAU+1
136.          EXISP=POSGT(IGAUS)
137.          ETASP=POSGT(NGAUS+IGAUS)
138.          CALL FFORMA(EXISP,ETASP,EGASP,NDIME,NNODE,FFORM,DERIV)
139.          CALL JACOBM(IELEM,DJACB,KPGAU)
140.          DVOLU=DJACB*PESGT(IGAUS)
141.          CALL BMAT(IELEM,KPGAU)
142.          CALL DBMATX
143.          DO IEVAB=1,NEVAB
144.            DO JEVAB=IEVAB,NEVAB
145.              DO ITENS=1,NTENS
146.                RIGID(IEVAB,JEVAB)=RIGID(IEVAB,JEVAB)+
147.                BMATZ(ITENS,IEVAB)*DBMAT(ITENS,JEVAB)*DVOLU
148.              ENDDO
149.              IF (NTRAN.GT.0) THEN
150.                PATOP=FFORM(IEVAB)*FFORM(JEVAB)*DVOLU*ROCEE
151.                ZMASA(IEVAB,JEVAB)=ZMASA(IEVAB,JEVAB)+PATOP
152.                IF (NTRAN.EQ.3) XLUMP=XLUMP+DVOLU*ROCEE
153.              ENDIF
154.            ENDDO
155.          ENDDO
156.        DO ITENS=1,NTENS
157.          DO IEVAB=1,NEVAB
158.            TENSZ (ITENS,IEVAB,KPGAU) =DBMAT (ITENS,IEVAB)
159.          ENDDO
160.        ENDDO
161.      ENDDO
162.    ENDIF
163.  ENDIF
164.  C
165.  C*** BUCLE PARA INTEGRACION NUMERICA EN ELEMENTOS SOLIDOS
166.  C   TRIDIMENSIONALES HEXAGONALES
167.  C
168.        IF (NDIME.EQ.3) THEN
169.          DO IGAUS=1,NGAUS
170.            DO JGAUS=1,NGAUS
171.              DO KGAUS=1,NGAUS
172.                KPGAU=KPGAU+1
173.                EXISP=POSPG(IGAUS)
174.                ETASP=POSPG(JGAUS)
175.                EGASP=POSPG(KGAUS)
176.                CALL FFORMA(EXISP,ETASP,EGASP,NDIME,NNODE,FFORM,DERIV)
177.                CALL JACOBM(IELEM,DJACB,KPGAU)
178.                DVOLU=DJACB*PESPG(IGAUS)*PESPG(JGAUS)*PESPG(KGAUS)
179.                CALL BMAT(IELEM,KPGAU)
180.                CALL DBMATX
181.                DO IEVAB=1,NEVAB
182.                  DO JEVAB=IEVAB,NEVAB

```



```

183.          DO ITENS=1,NTENS
184.              RIGID(IEVAB,JEVAB)=RIGID(IEVAB,JEVAB)+
185.                  BMATZ(ITENS,IEVAB)*DBMAT(ITENS,JEVAB)*DVOLU
186.          ENDDO
187.          IF (NTRAN.GT.0) THEN
188.              PATOP=FFORM(IEVAB)*FFORM(JEVAB)*DVOLU*ROCEE
189.              ZMASA(IEVAB,JEVAB)=ZMASA(IEVAB,JEVAB)+PATOP
190.              IF (NTRAN.EQ.3) XLUMP=XLUMP+DVOLU*ROCEE
191.          ENDIF
192.      ENDDO
193.  ENDDO
194.      DO ITENS=1,NTENS
195.          DO IEVAB=1,NEVAB
196.              TENSZ(ITENS,IEVAB,KPGAU)=DBMAT(ITENS,IEVAB)
197.          ENDDO
198.      ENDDO
199.  ENDDO
200.  ENDDO
201.  ENDDO
202.  ENDDO
203.  C
204.  C*** CALCULO DE LA APORTACION DE RIGIDEZ POR CONVECCION/RADIACION
205.  C
206.      IF (NFRON.NE.0) THEN
207.          CALL CONVECC(IELEM,FTFOR,LPROP)
208.          DO IEVAB=1,NEVAB
209.              DO JEVAB=IEVAB,NEVAB
210.                  RIGID(IEVAB,JEVAB)=RIGID(IEVAB,JEVAB)+FTFOR(IEVAB,JEVAB)
211.              ENDDO
212.          ENDDO
213.      ENDIF
214.  C
215.  C*** CALCULA LA PARTE TRIANGULAR INFERIOR DE LA MATRIZ DE RIGIDEZ
216.  C
217.      DO IEVAB=1,NEVAB
218.          DO JEVAB=IEVAB,NEVAB
219.              RIGID(JEVAB,IEVAB)=RIGID(IEVAB,JEVAB)
220.              IF (NTRAN.GT.0) ZMASA(JEVAB,IEVAB)=ZMASA(IEVAB,JEVAB)
221.          ENDDO
222.      ENDDO
223.  C
224.  C*** ALMACENA MAT DE RIGIDEZ ELEMENTAL, LA MAT DB PARA EL CALCULO
225.  C DE TENSIONES Y LAS COORDENADAS DE LOS PUNTOS DE INTEGRACION
226.  C
227.      WRITE(1) RIGID
228.      WRITE(3) TENSZ,CORPG
229.      IF (NTRAN.GT.0) WRITE(4) ZMASA
230.  ENDDO
231.  RETURN
232.  END

```

```

1.      SUBROUTINE SOLUCION
2.  C*****
3.  C
4.  C***      ENSAMBLA Y RESUELVE EL SISTEMA DE ECUACIONES MATRICIALES
5.  C
6.  C*****
7.      IMPLICIT NONE
8.      INCLUDE 'dtsgral.f'
9.      INCLUDE 'data.f'
10.     INCLUDE 'calculo.f'
11.     INCLUDE 'solu.f'

```

```

12. C*****
13. C
14. C*** SOLUCION DEL SISTEMA DE ECUACIONES UTILIZANDO ELIMINACION
15. C GAUSSIANA DIRECTA
16. C
17. CALL ENSAMBLA
18. CALL REDUCE
19. CALL SUSTITUIR
20. RETURN
21. END

1. SUBROUTINE SUAV1
2. C*****
3. C
4. C*** CALCULA LAS TENSIONES SUAVIZADAS EN LOS NODOS
5. C
6. C*****
7. implicit none
8. INCLUDE 'dtsgral.f'
9. INCLUDE 'data.f'
10. INCLUDE 'gaussdat.f'
11. INCLUDE 'calculo.f'
12. INCLUDE 'solucuas.f'
13. INTEGER*2 NSTR1, MPOST, INDC, NNOD1, NNOD2, IELEM, ICONT, INODE,
14. . LPROP, LNODE, IGAUS, JGAUS, KPGAU, MTENS, LNOD1, LPOS1,
15. . MPOS1, IPOST, LNOD2, LNOD3, LPOSI, MPOSI, NPOSI, IPNOD,
16. . MTNOD, ITNOD, IDIME, JTNOD, ISTR1, JSTR1, KTNOD, KGAUS,
17. . INSTR1, MPNOD
18. REAL*4 EGASP, EXISP, ETASP, TENSG, DJACB,
19. . DVOLU, DAREA, ELONG
20. DIMENSION TENSG(3)
21. WRITE(29) IDIME
22. IF(NNODE.EQ.20) THEN
23. WRITE(29) (0)
24. RETURN
25. ENDIF
26. REWIND (8)
27. REWIND (10)
28. EGASP=0.0
29. NSTR1=NTENS
30. MPOST=0
31. C
32. C*** CALCULO DE TENSIONES
33. C
34. WRITE(29) NSTR1
35. MTNOD=NSTR1*MPNOD
36. DO ITNOD=1, MTNOD
37. NAUXI(ITNOD)=0
38. TENST(ITNOD)=0.0
39. SMOTI(ITNOD)=0.0
40. ENDDO
41. INDC=0
42. NNOD1=0
43. IF (NNODE.EQ.8.OR.NNODE.EQ.9) THEN
44. INDC=1
45. NNOD1=7
46. NNOD2=8
47. IF (NNODE.EQ.9) NNOD2=9
48. NNODE=4
49. ENDIF
50. IF (NNODE.EQ.6) THEN
51. INDC=1

```

```

52.          NNOD1=5
53.          NNOD2=5
54.          NNODE=3
55.        ENDIF
56.        IF (INDC.NE.0) THEN
57.          OPEN(11,FILE='TENS.TMP',FORM='UNFORMATTED')
58.          WRITE(11) LNODS
59.          DO IELEM=1,NELEM
60.            ICONT=0
61.            DO INODE=1,NNOD1,2
62.              ICONT=ICONT+1
63.              LNODS (IELEM, ICONT) =LNODS (IELEM, INODE)
64.            ENDDO
65.          ENDDO
66.        ENDIF
67.        NTNOD=NSTR1*NNODE
68.      C
69.      C*** BUCLE SOBRE CADA ELEMENTO
70.      C
71.        DO IELEM=1,NELEM
72.          LPROP=ABS (MATNU (IELEM) )
73.        C
74.        C*** CALCULA LAS COORDENADAS DE LOS PUNTOS NODALES DEL ELEMENTO
75.        C
76.          DO INODE=1,NNODE
77.            LNODE=LNODS (IELEM, INODE)
78.            DO IDIME=1,NDIME
79.              COREL (IDIME, INODE) =COORD (LNODE, IDIME)
80.            ENDDO
81.          ENDDO
82.        C
83.        C*** INICIALIZA LAS MATRIZ DE SUAVIZADO ELEMENTALES
84.        C
85.          DO ITNOD=1,NTNOD
86.            TENS1 (ITNOD) =0.0
87.            DO JTNOD=1,NTNOD
88.              XMASA (ITNOD, JTNOD) =0.0
89.            ENDDO
90.            DO ISTR1=1,NSTR1
91.              XFORT (ITNOD, ISTR1) =0.0
92.              XFORM (ISTR1, ITNOD) =0.0
93.            ENDDO
94.          ENDDO
95.          KPGAU=0
96.          IF (NDIME.EQ.1) THEN
97.            DO IGAUS=1,NGAUS
98.              KPGAU=KPGAU+1
99.              EXISP=POSPG (IGAUS)
100.             READ(10) TENSG
101.             CALL FFORMA (EXISP, 0, 0, NDIME, NNODE, FFORM, DERIV)
102.             CALL JACOBM (IELEM, DJACB, KPGAU)
103.             DVOLU=DJACB*PESPG (IGAUS)
104.             DO ITNOD=1,NTNOD
105.               DO ISTR1=1,NSTR1
106.                 XFORM (ISTR1, ITNOD) =0.0
107.                 XFORT (ITNOD, ISTR1) =0.0
108.               ENDDO
109.             ENDDO
110.             DO INODE=1,NNODE
111.               MTENS= (INODE-1) *NSTR1
112.               DO ISTR1=1,NSTR1
113.                 XFORM (ISTR1, MTENS+ISTR1) =FFORM (INODE)
114.                 XFORT (MTENS+ISTR1, ISTR1) =FFORM (INODE)

```

```

115.          ENDDO
116.      ENDDO
117.      DO ITNOD=1,NTNOD
118.          DO JSTR1=1,NSTR1
119.              TENS1(ITNOD)=TENS1(ITNOD)+
120.                  XFORT(ITNOD,JSTR1)*TENSG(JSTR1)*DVOLU
121.          DO KTNOD=1,NTNOD
122.              XMASA(ITNOD,KTNOD)=XMASA(ITNOD,KTNOD)+
123.                  XFORT(ITNOD,JSTR1)*XFORM(JSTR1,KTNOD)*DVOLU
124.          ENDDO
125.      ENDDO
126.  ENDDO
127.  ENDDO
128.  ENDIF
129.  IF (NDIME.EQ.2) THEN
130.  C
131.  C*** BUCLE PARA INTEGRACION NUMERICA EN ELEMENTOS CUADRILATEROS O
132.  C    TRIANGULARES
133.  C
134.          IF (NNODE.EQ.3.OR.NNODE.EQ.6) THEN
135.  C
136.  C*** ELEMENTOS TRIANGULARES
137.  C
138.          DO IGAUS=1,NGAUS
139.              KPGAU=KPGAU+1
140.              EXISP=POSGT(IGAUS)
141.              ETASP=POSGT(NGAUS+IGAUS)
142.              READ(10)TENSG
143.              CALL FFORMA(EXISP,ETASP,EGASP,NDIME,NNODE,FFORM,DERIV)
144.              CALL JACOBM(IELEM,DJACB,KPGAU)
145.              DVOLU=DJACB*PESGT(IGAUS)
146.              DO ITNOD=1,NTNOD
147.                  DO ISTR1=1,NSTR1
148.                      XFORM(ISTR1,ITNOD)=0.0
149.                      XFORT(ITNOD,ISTR1)=0.0
150.                  ENDDO
151.              ENDDO
152.              DO INODE=1,NNODE
153.                  MTENS=(INODE-1)*NSTR1
154.                  DO ISTR1=1,NSTR1
155.                      XFORM(ISTR1,MTENS+ISTR1)=FFORM(INODE)
156.                      XFORT(MTENS+ISTR1,ISTR1)=FFORM(INODE)
157.                  ENDDO
158.              ENDDO
159.              DO ITNOD=1,NTNOD
160.                  DO JSTR1=1,NSTR1
161.                      TENS1(ITNOD)=TENS1(ITNOD)+
162.                          XFORT(ITNOD,JSTR1)*TENSG(JSTR1)*DVOLU
163.                  DO KTNOD=1,NTNOD
164.                      XMASA(ITNOD,KTNOD)=XMASA(ITNOD,KTNOD)+
165.                          XFORT(ITNOD,JSTR1)*XFORM(JSTR1,KTNOD)*DVOLU
166.                  ENDDO
167.              ENDDO
168.          ENDDO
169.      ENDDO
170.  ELSE
171.  C
172.  C*** ELEMENTOS CUADRILATEROS
173.  C
174.          DO IGAUS=1,NGAUS
175.              DO JGAUS=1,NGAUS
176.                  KPGAU=KPGAU+1
177.                  EXISP=POSPG(IGAUS)

```



```

178.          ETASP=POSPG (JGAUS)
179.          READ (10) TENSG
180.          CALL FFORMA (EXISP, ETASP, EGASP, NDIME, NNODE, FFORM, DERIV)
181.          DO ITNOD=1, NTNOD
182.              DO ISTR1=1, NSTR1
183.                  XFORM (ISTR1, ITNOD)=0.0
184.                  XFORT (ITNOD, ISTR1)=0.0
185.              ENDDO
186.          ENDDO
187.          DO INODE=1, NNODE
188.              MTENS= (INODE-1) *NSTR1
189.              DO ISTR1=1, NSTR1
190.                  XFORM (ISTR1, MTENS+ISTR1) =FFORM (INODE)
191.                  XFORT (MTENS+ISTR1, ISTR1) =FFORM (INODE)
192.              ENDDO
193.          ENDDO
194.          CALL JACOBM (IELEM, DJACB, KPGAU)
195.          DVOLU=DJACB*PESPG (IGAUS) *PESPG (JGAUS)
196.          DO ITNOD=1, NTNOD
197.              DO JSTR1=1, NSTR1
198.                  TENS1 (ITNOD) =TENS1 (ITNOD) +
199.                      XFORT (ITNOD, JSTR1) *TENSG (JSTR1) *DVOLU
200.              DO KTNOD=1, NTNOD
201.                  XMASA (ITNOD, KTNOD) =XMASA (ITNOD, KTNOD) +
202.                      XFORT (ITNOD, JSTR1) *XFORM (JSTR1, KTNOD) *DVOLU
203.              ENDDO
204.          ENDDO
205.          ENDDO
206.          ENDDO
207.          ENDDO
208.          ENDIF
209.          ENDIF
210.          IF (NDIME.EQ.3) THEN
211. C
212. C*** BUCLE PARA INTEGRACION NUMERICA EN ELEMENTOS SOLIDOS
213. C   TRIDIMENSIONALES HEXAGONALES
214. C
215.          DO IGAUS=1, NGAUS
216.              DO JGAUS=1, NGAUS
217.                  DO KGAUS=1, NGAUS
218.                      KPGAU=KPGAU+1
219.                      EXISP=POSPG (IGAUS)
220.                      ETASP=POSPG (JGAUS)
221.                      EGASP=POSPG (KGAUS)
222.                      READ (10) TENSG
223.                      CALL FFORMA (EXISP, ETASP, EGASP, NDIME, NNODE, FFORM, DERIV)
224.                      CALL JACOBM (IELEM, DJACB, KPGAU)
225.                      DVOLU=DJACB*PESPG (IGAUS) *PESPG (JGAUS) *PESPG (KGAUS)
226.                      DO ITNOD=1, NTNOD
227.                          DO ISTR1=1, NSTR1
228.                              XFORM (ISTR1, ITNOD)=0.0
229.                              XFORT (ITNOD, ISTR1)=0.0
230.                          ENDDO
231.                      ENDDO
232.                      DO INODE=1, NNODE
233.                          MTENS= (INODE-1) *NSTR1
234.                          DO INSTR1=1, NSTR1
235.                              XFORM (ISTR1, MTENS+ISTR1) =FFORM (INODE)
236.                              XFORT (MTENS+ISTR1, ISTR1) =FFORM (INODE)
237.                          ENDDO
238.                      ENDDO
239.                      DO ITNOD=1, NTNOD
240.                          DO JSTR1=1, NSTR1

```

```

241.          TENS1 ( ITNOD ) = TENS1 ( ITNOD ) +
242.          .          XFORT ( ITNOD , JSTR1 ) * TENS1 ( JSTR1 ) * DVOLU
243.          DO KTNOD = 1 , NTNOD
244.          .          XMASA ( ITNOD , KTNOD ) = XMASA ( ITNOD , KTNOD ) +
245.          .          XFORT ( ITNOD , JSTR1 ) * XFORM ( JSTR1 , KTNOD ) * DVOLU
246.          .          ENDDO
247.          .          ENDDO
248.          .          ENDDO
249.          .          ENDDO
250.          .          ENDDO
251.          .          ENDDO
252.          .          ENDDO
253.          .          IF ( INDC . EQ . 0 ) THEN
254.          C
255.          C*** ENSAMBLAJE MATRICES DE SUAVIZADO
256.          C
257.          .          DO ITNOD = 1 , NTNOD
258.          .          DO JTNOD = 1 , NTNOD
259.          .          IF ( ITNOD . NE . JTNOD ) THEN
260.          .          .          XMASA ( ITNOD , ITNOD ) = XMASA ( ITNOD , ITNOD ) + XMASA ( ITNOD , JTNOD )
261.          .          .          XMASA ( ITNOD , JTNOD ) = 0 . 0
262.          .          .          ENDDO
263.          .          .          ENDDO
264.          .          .          ENDDO
265.          .          DO INODE = 1 , NNODE
266.          .          .          LNOD1 = ( LNODS ( IELEM , INODE ) - 1 ) * NSTR1
267.          .          .          DO ISTR1 = 1 , NSTR1
268.          .          .          .          LPOS1 = ( INODE - 1 ) * NSTR1 + ISTR1
269.          .          .          .          MPOS1 = LNOD1 + ISTR1
270.          .          .          .          IF ( MPOS1 . GT . MPOST ) MPOST = MPOS1
271.          .          .          .          TENST ( MPOS1 ) = TENST ( MPOS1 ) + TENS1 ( LPOS1 )
272.          .          .          .          SMOTI ( MPOS1 ) = SMOTI ( MPOS1 ) + XMASA ( LPOS1 , LPOS1 )
273.          .          .          .          ENDDO
274.          .          .          ENDDO
275.          .          ELSE
276.          .          .          CALL INVER
277.          .          .          IF ( IERROR . NE . 0 ) THEN
278.          .          .          .          WRITE ( 6 , 710 )
279.          .          .          .          710   FORMAT ( 'ERROR EN EL LA INVERSION DE LA MATRIZ DE ALISADO ' ,
280.          .          .          .          .          ' DE TENSIONES ' )
281.          .          .          .          STOP
282.          .          .          ENDDO
283.          .          .          DO ITNOD = 1 , NTNOD
284.          .          .          .          TEMPO ( ITNOD ) = 0 . 0
285.          .          .          .          DO JTNOD = 1 , NTNOD
286.          .          .          .          .          TEMPO ( ITNOD ) = TEMPO ( ITNOD ) + XINV ( ITNOD , JTNOD ) * TENS1 ( JTNOD )
287.          .          .          .          ENDDO
288.          .          .          ENDDO
289.          C
290.          C*** ENSAMBLAJE DE TENSIONES
291.          C
292.          .          DO INODE = 1 , NNODE
293.          .          .          LNOD1 = ( LNODS ( IELEM , INODE ) - 1 ) * NSTR1
294.          .          .          DO ISTR1 = 1 , NSTR1
295.          .          .          .          LPOS1 = ( INODE - 1 ) * NSTR1 + ISTR1
296.          .          .          .          MPOS1 = LNOD1 + ISTR1
297.          .          .          .          IF ( MPOS1 . GT . MPOST ) MPOST = MPOS1
298.          .          .          .          TENST ( MPOS1 ) = TENST ( MPOS1 ) + TEMPO ( LPOS1 )
299.          .          .          .          NAUXI ( MPOS1 ) = NAUXI ( MPOS1 ) + 1
300.          .          .          .          ENDDO
301.          .          .          ENDDO
302.          .          ENDDO
303.          ENDDO

```

```

304.     IF (INDC.EQ.0) THEN
305.         DO IPOST=1,MPOST
306.             TENST(IPOST)=-TENST(IPOST)/SMOTI(IPOST)
307.         ENDDO
308.     ELSE
309.         DO IPOST=1,MPOST
310.             IF(NAUXI(IPOST).NE.0)
311.                 TENST(IPOST)=TENST(IPOST)/FLOAT(NAUXI(IPOST))
312.         ENDDO
313.     ENDIF
314. C
315. C**** INTERPOLA RESULTADOS SI ES NECESARIO
316. C
317.     IF (INDC.NE.0) THEN
318.         REWIND(11)
319.         READ(11) LNODS
320.         DO IELEM=1,NELEM
321.             DO INODE=1,NNOD1,2
322.                 LNOD1=LNODS(IELEM,INODE)-1
323.                 LNOD2=LNODS(IELEM,INODE+1)-1
324.                 IF (INODE.EQ.NNOD1) THEN
325.                     LNOD3=LNODS(IELEM,1)-1
326.                 ELSE
327.                     LNOD3=LNODS(IELEM,INODE+2)-1
328.                 ENDIF
329.                 DO ISTR1=1,NSTR1
330.                     LPOSI=LNOD1*NSTR1+ISTR1
331.                     MPOSI=LNOD3*NSTR1+ISTR1
332.                     NPOSI=LNOD2*NSTR1+ISTR1
333.                     TENST(NPOSI)=(TENST(LPOSI)+TENST(MPOSI))/2.0
334.                 ENDDO
335.             ENDDO
336.         ENDDO
337.         CLOSE(11,STATUS='DELETE')
338.         IF(NNOD2.EQ.9) THEN
339.             NNODE=9
340.             DO IELEM=1,NELEM
341.                 DO ISTR1=1,NSTR1
342.                     LNODE=LNODS(IELEM,9)
343.                     NPOSI=(LNODE-1)*NSTR1+ISTR1
344.                     TENST(NPOSI)=0.0
345.                     DO INODE=1,NNOD1,2
346.                         LPOSI=(LNODS(IELEM,INODE)-1)*NSTR1+ISTR1
347.                         TENST(NPOSI)=TENST(NPOSI)+TENST(LPOSI)
348.                     ENDDO
349.                     TENST(NPOSI)=TENST(NPOSI)/4.0
350.                 ENDDO
351.             ENDDO
352.         ELSE
353.             NNODE=NNOD1+1
354.         ENDIF
355.     ENDIF
356. C
357. C
358. C**** IMPRESION DE FLUJOS SUAVIZADAS
359. C
360.     WRITE(6,904)
361.     904 FORMAT(/,10X,'FLUJOS ALISADAS NODALES',/)
362.     WRITE(6,905)
363.     905 FORMAT(1X,'NODO',2X,'X-FLUJ.',5X,'Y-FLUJ.',5X,'Z-FLUJ.')
```

```

367.          MPOSI=NPOSI+ISTR1
368.          TENSG(ISTR1)=TENST(MPOSI)
369.          ENDDO
370.          WRITE(6,920) IPNOD, (TENSG(ISTR1), ISTR1=1, NSTR1)
371.          MPNOD=(IPNOD)
372.          WRITE(29) MPNOD, (TENSG(ISTR1), ISTR1=1, NSTR1)
373.          ENDDO
374.          WRITE(8) TENST
375. 920  FORMAT(I5, 6E12.5)
376.          RETURN
377.          END

```

```

1.          SUBROUTINE SUSTITUIR
2.  C*****
3.  C
4.  C***          REALIZA LA SUSTITUCION HACIA ATRAS
5.  C
6.  C*****
7.          IMPLICIT NONE
8.          INCLUDE 'dtsgral.f'
9.          INCLUDE 'solu.f'
10.         INTEGER*2          MPNOD, NEONS, IEONS, NEON1, NBACK, NBAC1, ICOLS,
11.         .                  KOUNT, IPNOD, NGUSH, NCONT, NGISH, IGASH, MCONT,
12.         .                  NLOCA, IGDLN
13.         REAL*4             PIVOT, RESID
14.
15.  C*****
16.         NEONS=NSVAB
17.         DO 5 IEONS=1, NEONS
18.           REACT(IEONS)=0.0
19.         5 CONTINUE
20.         NEON1=NEONS+1
21.         DO 30 IEONS=1, NEONS
22.           NBACK=NEON1-IEONS
23.           PIVOT=ASTIF(NBACK, NBACK)
24.           RESID=ASLOD(NBACK)
25.           IF(NBACK.EQ.NEONS) GO TO 20
26.           NBAC1=NBACK+1
27.           DO 10 ICOLS=NBAC1, NEONS
28.             RESID=RESID-ASTIF(NBACK, ICOLS)*DESPL(ICOLS)
29.           10 CONTINUE
30.         20 IF(IFFIX(NBACK).EQ.0) DESPL(NBACK)=RESID/PIVOT
31.           IF(IFFIX(NBACK).EQ.1) DESPL(NBACK)=FIXED(NBACK)
32.           IF(IFFIX(NBACK).EQ.1) REACT(NBACK)=-RESID
33.         30 CONTINUE
34.         KOUNT=0
35.         WRITE(6,900)
36. 900  FORMAT(/, 5X, 'TEMPERATURAS')
37.         WRITE(6,905)
38. 905  FORMAT(6X, 'NODO', 2X, 'TEMPERATURA')
39.         DO 450 IPNOD=1, NPNOD
40.           NCONT=IPNOD*NGDLN
41.           NGISH=NCONT-NGDLN+1
42.           MPNOD=(IPNOD)
43.           WRITE(6,920) IPNOD, (DESPL(IGASH), IGASH=NGISH, NCONT)
44.           WRITE(29) MPNOD, (DESPL(IGASH), IGASH=NGISH, NCONT)
45.         450 CONTINUE
46. 920  FORMAT(I10, 3E14.6)
47.         WRITE(6,925)
48. 925  FORMAT(/, '0', 5X, 'REACCIONES')
49.         WRITE(6,930)
50. 930  FORMAT('0', 5X, 'NODO', 6X, 'REACCION')

```



```

51.          MCONT=0
52.          DO 481 IPNOD=1,NPNOD
53.          NLOCA=(IPNOD-1)*NGDLN
54.          DO 484 IGDLN=1,NGDLN
55.          NGUSH=NLOCA+IGDLN
56.          IF (IFFIX(NGUSH).GT.0) GO TO 483
57.      484 CONTINUE
58.          GO TO 481
59.      483 MCONT=MCONT+1
60.          DO 482 IGDLN=1,NGDLN
61.          NCONT=NLOCA+IGDLN
62.      482 TREAC (IGDLN)=REACT(NCONT)
63.          WRITE(6,945) IPNOD, (TREAC (IGDLN) , IGDLN=1,NGDLN)
64.      481 CONTINUE
65.      945 FORMAT(I10,3E14.6)
66.          RETURN
67.          END

1.          SUBROUTINE TENSIONES
2.      C*****
3.      C
4.      C***          CALCULA LAS TENSIONES Y ESFUERZOS EN LOS PUNTOS DE GAUSS
5.      C
6.      C*****
7.          IMPLICIT NONE
8.          INCLUDE 'dtsgral.f'
9.          INCLUDE 'data.f'
10.         INCLUDE 'calculo.f'
11.         INCLUDE 'solu.f'
12.         INTEGER*2          NSTR1, IELEM, INODE, LNODE, NPOSN, IGDLN,
13.         .                  KPGAU, IGAUS, JGAUS, KGAUS, ITENS, KCONT, IDIME
14.         REAL*4             TENSG, DESEL
15.         DIMENSION         TENSG(3), DESEL(1,20)
16.      C*****
17.         REWIND(3)
18.         REWIND(10)
19.         WRITE(6,900)
20.     900 FORMAT(/,10X,'TENSIONES',/)
21.      C
22.      C*** SELECCION DE TIPO DE ANALISIS
23.      C
24.         IF(NDIME.EQ.1) WRITE(6,905)
25.         IF(NDIME.EQ.2) WRITE(6,906)
26.         IF(NDIME.EQ.3) WRITE(6,907)
27.         NSTR1=NTENS
28.     905 FORMAT(' ', 'G.P.', 2X, 'X-COORD', 4X, 'X-FLUJ. ')
29.     906 FORMAT(' ', 'G.P.', 2X, 'X-COORD', 2X, 'Y-COORD', 4X, 'X-FLUJ.', 5X,
30.         .      'Y-FLUJ. ')
31.     907 FORMAT(' ', 'G.P.', 2X, 'X-COORD', 2X, 'Y-COORD', 2X, 'Z-COORD', 4X,
32.         .      'X-FLUJ.', 5X, 'Y-FLUJ.', 5X, 'Z-FLUJ. ')
33.      C
34.      C*** BUCLE SOBRE CADA ELEMENTO
35.      C
36.         DO IELEM=1,NELEM
37.      C
38.      C*** LEE LA MATRIZ DB, Y LAS COORD. DE LOS PUNTOS DE INTEGRACION
39.      C
40.         READ(3) TENSZ,CORPG
41.      C
42.         WRITE(6,910) IELEM
43.     910  FORMAT(/,5X,'ELEMENTO NO.=' , I5)
44.      C

```

```

45. C*** IDENTIFICA LAS TEMPERATURAS Y LOS PUNTOS NODALES DEL ELEMENTO
46. C
47.     DO INODE=1,NNODE
48.         LNODE=LNODS ( IELEM, INODE)
49.         NPOSN=( LNODE-1)*NGDLN
50.         DO IGDLN=1,NGDLN
51.             NPOSN=NPOSN+1
52.             DESEL (IGDLN, INODE) =DESPL (NPOSN)
53.         ENDDO
54.     ENDDO
55.     KPGAU=0
56.     IF (NDIME.EQ.1) THEN
57. C
58. C***         BUCLE SOBRE CADA PUNTO DE INTEGRACION
59. C
60.         DO IGAUS=1,NGAUS
61.             KPGAU=KPGAU+1
62. C
63. C*** CALCULO DE TENSIONES EN LOS PUNTOS DE INTEGRACION
64. C
65.         DO ITENS=1,NTENS
66.             TENSG (ITENS) =0.0
67.             KCONT=0
68.             DO INODE=1,NNODE
69.                 DO IGDLN=1,NGDLN
70.                     KCONT=KCONT+1
71.                     TENSG (ITENS) =TENSG (ITENS) +
72.                     TENSZ (ITENS, KCONT, KPGAU) *DESEL (IGDLN, INODE)
73.                 ENDDO
74.             ENDDO
75.         ENDDO
76. C
77. C*** SALIDA DE TENSIONES
78. C
79.         WRITE (6,920) KPGAU, (CORPG (IDIME, KPGAU) , IDIME=1, NDIME) ,
80.         .           (TENSG (ITENS) , ITENS=1, NTENS)
81.         WRITE (10) TENSZ
82.         ENDDO
83.     ENDIF
84.     IF (NDIME.EQ.2) THEN
85.         IF (NNODE.EQ.4 .OR. NNODE.EQ.8 .OR. NNODE.EQ.9) THEN
86. C
87. C***BUCLE SOBRE CADA PUNTO DE INTEGRACION EN ELEMENTOS CUADRILATEROS
88. C
89.         DO IGAUS=1,NGAUS
90.             DO JGAUS=1,NGAUS
91.                 KPGAU=KPGAU+1
92. C
93. C*** CALCULO DE TENSIONES EN LOS PUNTOS DE INTEGRACION
94. C
95.         DO ITENS=1,NTENS
96.             TENSG (ITENS) =0.0
97.             KCONT=0
98.             DO INODE=1,NNODE
99.                 DO IGDLN=1,NGDLN
100.                     KCONT=KCONT+1
101.                     TENSG (ITENS) =TENSG (ITENS) +
102.                     TENSZ (ITENS, KCONT, KPGAU) *DESEL (IGDLN, INODE)
103.                 ENDDO
104.             ENDDO
105.         ENDDO
106. C
107. C*** ESCRIBE RESULTADOS

```

```

108. C
109.          WRITE(6,920) KPGAU, (CORPG(IDIME,KPGAU), IDIME=1, NDIME),
110.          .          (TENSG(ITENS), ITENS=1, NTENS)
111.          WRITE(10)TENSG
112.          ENDDO
113.          ENDDO
114.          ENDIF
115.          IF(NNODE.EQ.3.OR.NNODE.EQ.6) THEN
116. C
117. C*** BUCLE SOBRE CADA PUNTO DE INTEGRACION EN ELEMENTOS TRIANGULARES
118. C
119.          DO IGAUS=1,NGAUS
120.          KPGAU=KPGAU+1
121. C
122. C*** CALCULO DE TENSIONES EN LOS PUNTOS DE INTEGRACION
123. C
124.          DO ITENS=1,NTENS
125.          TENSG(ITENS)=0.0
126.          KCONT=0
127.          DO INODE=1,NNODE
128.          DO IGDLN=1,NGDLN
129.          KCONT=KCONT+1
130.          TENSG(ITENS)=TENSG(ITENS)+
131.          .          TENSZ(ITENS,KCONT,KPGAU)*DESEL(IGDLN,INODE)
132.          ENDDO
133.          ENDDO
134.          ENDDO
135. C
136. C*** SALIDA DE TENSIONES
137. C
138.          WRITE(6,920) KPGAU, (CORPG(IDIME,KPGAU), IDIME=1, NDIME),
139.          .          (TENSG(ITENS), ITENS=1, NTENS)
140.          WRITE(10)TENSG
141.          ENDDO
142.          ENDIF
143.          ENDIF
144.          IF (NDIME.EQ.3) THEN
145. C
146. C*** BUCLE SOBRE CADA PUNTO DE INTEGRACION EN ELEMENTOS SOLIDOS
147. C TRIDIMENSIONALES
148. C
149.          DO IGAUS=1,NGAUS
150.          DO JGAUS=1,NGAUS
151.          DO KGAUS=1,NGAUS
152.          KPGAU=KPGAU+1
153. C
154. C*** CALCULO DE TENSIONES EN PUNTOS DE INTEGRACION
155. C
156.          DO ITENS=1,NTENS
157.          TENSG(ITENS)=0.0
158.          KCONT=0
159.          DO INODE=1,NNODE
160.          DO IGDLN=1,NGDLN
161.          KCONT=KCONT+1
162.          TENSG(ITENS)=TENSG(ITENS)+
163.          .          TENSZ(ITENS,KCONT,KPGAU)*DESEL(IGDLN,INODE)
164.          ENDDO
165.          ENDDO
166.          ENDDO
167. C
168. C*** ESCRIBE RESULTADOS
169. C
170.          WRITE(6,930) KPGAU, (CORPG(IDIME,KPGAU), IDIME=1, NDIME),

```

```
171. . (TENSG (ITENS) , ITENS=1 , NTENS)
172. WRITE (10) TENSG
173. ENDDO
174. ENDDO
175. ENDDO
176. ENDIF
177. ENDDO
178. 920 FORMAT (I5, 2F10.4, 6E12.5, F10.4)
179. 930 FORMAT (I5, 3F10.4, 6E12.5)
180. RETURN
181. END
```