

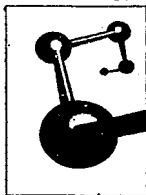
2Ej  
76



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**  
**FACULTAD DE CIENCIAS**

**METODOS NUMERICOS DE ELEMENTOS  
FINITOS PARA LA SOLUCION DE PROBLEMAS  
DE VALOR A LA FRONTERA**

**T E S I S**  
**QUE PARA OBTENER EL TITULO DE**  
**LICENCIADO EN ACTUARIA**  
**P R E S E N T A**  
**ALEJANDRO SAAVEDRA FLORES**



**MEXICO, D. F.**

**1987**



Universidad Nacional  
Autónoma de México



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

# I N D I C E

	PAG.
LISTA DE TABLAS	
LISTA DE FIGURAS	
INTRODUCCION .....	2
PRIMERA PARTE	
CAPITULO I: PROBLEMA DE VALOR A LA FRONTERA .....	6
METODOS DE ELEMENTOS FINITOS .....	6
1. PRELIMINARES .....	6
2. COLOCACION .....	8
3. GALERKIN .....	9
4. COLOCACION - GALERKIN .....	10
SEGUNDA PARTE	
CAPITULO II: SOLUCION AL PROBLEMA DE VALOR A LA FRONTERA POR EL METODO DE COLOCACION.....	12
1. PRELIMINARES .....	12
2. DESCRIPCION DEL METODO DE COLOCACION...	14
3. SOLUCION DE LA ECUACION DIFERENCIAL ORDINARIA LINEAL DE SEGUNDO ORDEN POR EL METODO DE COLOCACION .....	19

CAPITULO III: SOLUCION AL PROBLEMA DE VALOR A LA FRONTERA POR EL METODO DE GALERIN....	25
1. PRELIMINARES .....	25
2. APLICACION DE LA TEORIA DEL METODO DE ELEMENTOS FINITOS DE TIPO GALERKIN A NUESTRO PROBLEMA DE VALOR A LA FRONTERA .....	31

CAPITULO IV: SOLUCION AL PROBLEMA DE VALOR A LA FRONTERA POR EL METODO DE COLOCACION - GALERKIN .....	37
1. PRELIMINARES .....	37
2. DESCRIPCION DEL METODO DE COLOCACION - GALERKIN .....	38
3. APLICACION DE LA TEORIA DEL METODO DE ELEMENTOS FINITOS DE TIPO DE COLOCACION - GALERKIN A NUESTRO PROBLEMA DE VALOR A LA FRONTERA .....	41

TERCERA PARTE

CAPITULO V: DESCRIPCION DE LOS PROGRAMAS DE COMPUTO .....	45
1. INTRODUCCION .....	45

CUARTA PARTE

CAPITULO VI: TIPOS DE EJERCICIOS NUMERICOS ESTUDIADOS .....	54
---	----

	PAG.
1. INTRODUCCION .....	54
2. EJERCICIOS NUMERICOS .....	54
3. VALORES DE LOS PARAMETROS .....	55
4. ORDEN DE CONVERGENCIA .....	56
CONCLUSIONES .....	64
BIBLIOGRAFIA .....	68

## LISTA DE TABLAS

	PAG.
VI.3.1. Valores de los parámetros .....	55
VI.4.2. Orden de convergencia para los métodos .....	56
VI.4.2a. Ordenes de convergencia para el método de Co- locación .....	60
VI.4.2b. Ordenes de convergencia para el método de -- GALERKIN .....	60
VI.4.2c. Ordenes de convergencia para el método de -- COLOCACION - GALERKIN .....	60
VI.4.3a.3b.3c. Precisión de los métodos .....	61

## LISTA DE FIGURAS

	PAG.
II.2.1. <i>Coordenadas lineales y normalizadas del elemento finito <math>\Omega_i</math>.</i> .....	15
II.2.2 a. <i>Función base local en los nodos 1 y 2 y función base global en el nodo <math>i</math>.</i> .....	18
II.2.2 b. <i>Función base local asociada a la primera derivada en los nodos 1 y 2 y función base global asociada a la primera derivada en el nodo <math>i</math>.</i> .....	18
II.3.1. <i>Estructura de la matriz de la ecuación (1) - para el método de colocación.</i> .....	22
III.2.1. <i>Malla de elementos finitos de tipo cuadrática compuesta de polinomios interpolantes de LAGRANGE.</i> .....	31
III.2.2. <i>Valor de las funciones base globales para polinomios cuadráticos de LAGRANGE.</i> .....	31
V.1.1. <i>Programa Principal.</i> .....	50
V.1.2. <i>Subrutina INPSTA.</i> .....	51
V.1.3. <i>Subrutina STATIC.</i> .....	52

## INTRODUCCION



## INTRODUCCION

Los métodos de elementos finitos para problemas de valor a la frontera pueden tomar muchas formas, por ejemplo el método de elementos finitos de tipo COLOCACION ha sido aplicado en casos de problemas de valor a la frontera con condiciones a la frontera de tipo mixto: VER p. ej. [18].

Otros métodos pueden describirse basándose en aproximaciones de tipo  $L^2$  - GALERIKIN siendo éste método la forma débil más común, o bien, como una combinación de este método y el método de colocación.

Aplicar éstos métodos a un sistema de ecuaciones diferenciales no es inmediatamente claro. En este trabajo de tesis restringiremos nuestra atención al caso de una ecuación diferencial ordinaria lineal. Nuestro objetivo es investigar tres técnicas de elementos finitos, en particular consideremos

- 1) La descripción de los métodos.
- 2) El tratamiento de las condiciones a la frontera.
- 3) Los méritos relativos a cada uno de los métodos - en problemas con diferente grado de dificultad (tres problemas de prueba fueron elegidos).
- 4) La solución del sistema de ecuaciones algebraicas lineales por medio de algún método directo o iterativo conocido.

5) La determinación de las órdenes de convergencia - de cada uno de los métodos.

Algunas comparaciones sobre los métodos de elementos finitos para la solución de problemas de valor a la frontera - han aparecido con anterioridad en la literatura. Estas comparaciones están basadas principalmente sobre consideraciones teóricas, tal es el caso de la estimación del error; ver p. ej. - [19].

Cada uno de los métodos de elementos finitos tienen - características propias, pero sin embargo, hasta ahora no existen suficientes cálculos numéricos para establecer un Juicio - más claro para ver cual de estos métodos es el mejor.

El presente trabajo consta de cuatro partes distribuidas de la siguiente forma:

Primera parte. Contiene el capítulo uno, en el cual se plantea el problema de valor a la frontera de nuestro interés y se mencionan los métodos de elementos finitos usados para resolver dicho problema.

Segunda parte. Contiene los capítulos dos, tres y -- cuatro, en los cuales se describen y se aplican los métodos -- de:

- 1) Colocación
- 2) Galerkin
- 3) Colocación - Galerkin

Respectivamente y además se da la aplicación directa - a la solución al problema planteado con anterioridad.

Tercera parte. Contiene el capítulo cinco en el cual se describen los programas de cómputo siguientes:

- 1) COLFEM
- 2) GALFEM
- 3) COLGAL

Proporcionando posteriormente en el capítulo seis los resultados numéricos que se obtuvieron.

Cuarta parte. Contiene el capítulo seis, en el cual se aplican los problemas de valor propio a los que se les aplicó los métodos mencionados con anterioridad. Finalmente se dan las conclusiones pertinentes a los métodos y se anexan las versiones de los programas de cómputo, así como la bibliografía utilizada.

PRIMERA PARTE

CAPITULO UNO

PROBLEMA DE VALOR A LA FRONTERA  
METODOS DE ELEMENTOS FINITOS

PROBLEMA DE VALOR A LA FRONTERA  
METODOS DE ELEMENTOS FINITOS

1.1. PRELIMINARES

Muchos de los modelos de la ciencia y de la tecnología se expresan en la forma de ecuaciones diferenciales. En los modelos más sencillos, estas ecuaciones son ordinarias, es decir, que tienen una sola variable independiente, que llamaremos  $x$  cuando el problema es de valores a la frontera. En los más realistas, las ecuaciones diferenciales de interés serán ecuaciones parciales, con más de una variable independiente [17].

Estas ecuaciones en muchos casos no tienen solución analítica o si la tienen, esta solución es de una forma bastante complicada como para ser realmente útil (p. ej. una serie infinita). En otras palabras, se necesita una solución numérica que nos permita aproximar a la solución en varios puntos o hasta en cualquier punto del dominio considerado.

El problema de valor a la frontera que consideramos consiste en la siguiente ecuación diferencial ordinaria lineal de segundo orden.

$$- a(x) \frac{d^2 u(x)}{dx^2} + b(x) \frac{du(x)}{dx} + c(x) u(x) = f(x), \quad x \in [p, q] = \Omega \dots (1)$$

con condiciones a la frontera de la forma

$$\lambda_0 \frac{du(x)}{dx} + \mu_0 du(x) = v_0, \quad \dots (2)$$

$$\lambda_1 \frac{du(x)}{dx} + \mu_1 du(x) = v_1,$$

donde  $|\lambda_i| + |\mu_i| > 0, \quad i = 0, 1.$

El coeficiente  $a$  se supone continuo con  $a(x) > \kappa, \quad \kappa = \text{constante} > 0$  y  $b$  se supone continuo y no negativo para todo punto  $x \in [p, q].$

El coeficiente  $c$  y los datos  $F = f(x)$  se suponen suaves: todo lo anterior para asegurar la convergencia de los métodos: ver p. je. [16].

(1) es lineal porque las derivadas de la solución aparecen linealmente en la ecuación, es de segundo orden porque  $\frac{d^2 u(x)}{dx^2}$  es la derivada de más alto orden en la ecuación.

Con la aparición de las computadoras digitales al final de los 40, las técnicas numéricas se desarrollaron con gran rapidez, en particular en el campo de la solución numérica de ecuaciones diferenciales.

Un método general para construir soluciones aproximadas para problemas de valor a la frontera se le conoce con el nombre de métodos de elementos finitos (m.e.f) que apareció al final de la década de los 50 hasta alcanzar en la actualidad un alto grado de sofisticación. Estos métodos involucran dividir el dominio de solución  $\Omega$  en un número finito de subintervalos  $\Omega_i, \quad i = 1, N,$  los elementos finitos, y de esta forma construir la aproximación  $U_n(x)$  por tramos. Además usan conceptos variacionales para la construcción de la aproximación de

la solución sobre el conjunto de elementos finitos. En otras palabras, los m.e.f. obtienen un sistema de ecuaciones mediante la discretización del espacio de solución, la solución - - aproximada obtenida es una combinación finita de funciones conocidas previamente seleccionadas.

Examinaremos en el transcurso de este trabajo los siguientes m.e.f.:

## 1.2. COLOCACION

Una de las aproximaciones más directas para determinar una solución aproximada para un problema de valor a la - - frontera definido en un dominio simple como es el caso de un - intervalo es el método de colocación [16].

La idea básica de este método es introducir, como es usual, una solución aproximada  $u(x) \sim U_n(x) = \sum_{i=1}^n u_i \phi_i(x)$  - - que puede ser determinada de manera única seleccionando.

1) Funciones de ensayo suaves que son las funciones-base (localmente  $C^2$  para nuestro problema),

2) Funciones de prueba, y

3) Especificando un número de parámetros (coeficientes  $u_i$  que comúnmente son valores de la variable dependiente en puntos de la malla así como las correspondientes derivadas).

En este método, la solución aproximada se construye - para que satisfaga de antemano las condiciones a la frontera - [17] y los coeficientes en la aproximación son entonces determinados para que la aproximación satisfaga la ecuación diferente

cial en un número de puntos distintos en el dominio  $\Omega$ .

### 1.3. GALERKIN

Este método consiste en buscar una solución aproximada para el problema de valor a la frontera en un subespacio de dimensión finita  $H_0^{(N)}$  de funciones conocidas  $\phi_i$ ,  $i = \overline{1, N}$  del espacio  $H_0^1$ . Es decir, en vez de abordar el problema en un espacio de dimensión infinita, buscamos una solución aproximada  $u_N$  en  $H_0^{(N)}$  de la forma  $u_N(x) = \sum_{i=1}^N u_i \phi_i(x)$  la cual satisface el problema con  $H_0^1$  reemplazado por  $H_0^{(N)}$ , en donde las  $\phi_i$ ,  $i = \overline{1, N}$  son las funciones base globales linealmente independientes y las  $u_i$  son las coeficientes que se determinan sustituyendo la expansión  $u_N(x)$  en la formulación del problema y resolviendo el sistema algebraico lineal resultante.

Por otro lado, el método de Galerkin se basa en una formulación variacional del problema para el cual tenemos que integrar una vez por partes la formulación basada en el método de residuos pesados (M.R.P.).

La idea básica del M.R.P. consiste en buscar una aproximación  $u_N(x)$  a la solución  $u(x)$  de la ecuación (1) mediante una expresión analítica de la forma general

$$u(x) = \sum_{i=1}^N u_i \phi_i(x) \text{ donde las incógnitas son los coeficientes}$$

$u_i$ , mientras las funciones  $\phi_i(x)$ ,  $i = \overline{1, N}$  son funciones dadas, globales en el sentido que se extienden a todo el dominio  $\Omega$  donde se busca aproximar a la solución.

Luego entonces, volviendo al método que nos interesa, integrar por partes la formulación mencionada, significa, que una de las derivadas de las funciones de ensayo  $u$  se trans--



fiere mediante la integración a las funciones de prueba  $v$ .

#### 1.4. COLOCACION - GALERKIN

Las características de los dos métodos antes descritos pueden ser combinadas para formar un método híbrido llamado Colocación Galerkin para resolver el mismo problema - - (1). El método es intermedio entre los métodos de colocación y de Galerkin. Este método tiene la ventaja sobre el método de Galerkin sobre el mismo dominio en que las integrales involucran el producto de la solución aproximada y una función polinomial por tramos, por lo tanto, las integrales son más - - simples.

Las condiciones de continuidad en la solución aproximada son más débiles que aquellas requeridas en la aproximación para colocación: ver p. ej. [2].

En colocación - Galerkin tratamos otra variante del método de colocación en la cual la continuidad se trata en una forma semejante al método de Galerkin y con elementos de - - LAGRANGE de segundo grado. En este método solamente existe un punto de colocación que es precisamente el punto central de cada elemento  $\Omega_i$ ,  $i = \overline{1, N}$ .

SEGUNDA PARTE

CAPITULO DOS

SOLUCION AL PROBLEMA DE VALOR A LA FRONTERA POR  
EL METODO DE COLOCACION

## SOLUCION AL PROBLEMA DE VALOR A LA FRONTERA POR EL METODO DE COLOCACION

### 2.1. PRELIMINARES

En este capítulo se describe el método de colocación para la solución de la ecuación diferencial ordinaria lineal de segundo orden utilizada en el problema de valor a la frontera en una dimensión.

Las bases teóricas del método de colocación se proporcionaron desde 1930, [2]. El método fue aplicado en la solución de ecuaciones diferenciales por Slater [4] y Bartha [5].

Posteriormente fue desarrollado como un método general para resolver ecuaciones diferenciales ordinarias lineales. Frazen [6] usó muchas funciones de prueba diferentes, y empleó arbitrariamente los puntos de colocación. Lanczos [7] extendió la solución en términos de polinomios de Tchebychev, y usó las raíces de estos polinomios como los puntos de colocación.

Este método ha sido usado como un método de colocación ortogonal (C.O.), [8] el cual ha probado su efectividad en ciertos problemas no lineales concernientes al área de la ingeniería y química. Ha sido usado también para encontrar las soluciones aproximadas tanto para el flujo de neutrones como para las concentraciones de precursores de neutrones retardados [2;9].

No obstante, antes de 1972, todas las aplicaciones --

método de colocación traen consigo funciones de expansión definidas globalmente. Russell y Shampine [10] fueron los primeros en estudiar el método de colocación basado en funciones -- polinominales por tramos; ver p. ej. [9]. El uso de esta clase de funciones ofrece ciertas ventajas la teoría es mucho más -- simple y más poderosa y los cálculos involucran matrices las -- cuales son fácilmente resueltas mediante algún método de solución directa e interactivo.

Este método es algunas veces llamado colocación de -- elementos finitos, pero a lo largo de este trabajo de tesis -- nos referiremos, simplemente como método de colocación.

El método es bien conocido en ingeniería química [11] a pesar de que el método más simple para aplicar es el M.R.P.

Recientemente ha empezado a usarse en otras áreas de la ingeniería y en el área científica. El método fue usado -- por Meade [2] para resolver las ecuaciones cinéticas dependien -- tes del espacio.

La precisión del método de colocación es bastante sen -- sible a la localización de los puntos de colocación. Deboor y Schwartz [12] han probado que la localización óptima de estos -- puntos son las raíces de los polinomios de Legendre (puntos de Gauss).

Cuando los puntos de colocación son empleadas en esta forma al método se le conoce como método de colocación ortogo -- nal.

## 2.2. DESCRIPCIÓN DEL MÉTODO DE COLOCACIÓN

El problema de valor a la frontera que consideramos - en este trabajo de tesis consiste en la siguiente ecuación diferencial ordinaria lineal de segundo orden

$$Lu(x) \equiv -a(x) \frac{d^2 u(x)}{dx^2} + b(x) \frac{du(x)}{dx} + c(x)u(x) = F(x), \quad x \in [p, q] = \Omega \dots (1)$$

con condiciones a la frontera de la forma

$$\begin{aligned} \lambda_0 \frac{du(x)}{dx} + \mu_0 u(x) &= v_0 \\ &\dots (2) \\ \lambda_1 \frac{du(x)}{dx} + \mu_1 u(x) &= v_1 \end{aligned}$$

donde  $|\lambda_i| + |\mu_i| > 0$ ,  $i=0,1$ , y tal que las funciones  $a, b, c$  y  $f$  son suaves y  $\lambda_i, \mu_i$  son constantes. A esta forma de ecuación se le conoce como forma fuerte ya que derivadas de segundo orden actúan sobre la incógnita  $u$ .

Utilizaremos el método de colocación para aproximar - soluciones a (1), las cuales son una combinación lineal de un conjunto conveniente de funciones, i.e.,  $u(x) \sim u_n(x) =$

$\sum_{i=1}^N u_i \phi_i(x)$  en donde  $\phi_i(x)$  son funciones dadas globales en el sentido que se extienden a todo el dominio  $\Omega$  en donde se busca la solución.

Los coeficientes  $u_i$  de tales funciones se determinan para pedir que la combinación satisfaga la ecuación diferencial en ciertos puntos.

Para resolver esta ecuación por colocación, consideramos una partición que denotamos como  $\Delta$  tal que

$\Delta = \{z_i\}_{i=0}^{NINT} \geq 1$  con  $x_0 = z_0 < z_1 < \dots < z_{NINT} = x_1$  del intervalo  $[x_0, x_1]$ , y un conjunto  $I_i = [z_{i-1}, z_i]$  y  $h_i = z_i - z_{i-1}$  con  $i=1, NINT$  y  $NINT$  el número de subintervalos en  $\Delta$ .

En los tres métodos empleados en este trabajo, las soluciones aproximadas son polinomios por tramos expresados en términos de una base B-spline y se resuelven mediante un sistema de ecuaciones algebraicas lineales con una estructura en bloques casi diagonal. Procediendo con la descripción de la teoría consideraremos a  $k$  como el orden del espacio de prueba (el grado del polinomio más 1);  $NCOND$  como el número de condiciones de continuidad impuestas sobre el espacio de prueba, así si  $H_0^{(N)}$  es un elemento del espacio entonces  $H_0^{(N)} \in C^{NCOND-1}$ ;  $h$  el tamaño del intervalo ( $h = \max_i h_i$ ).

Hemos mencionado que una solución aproximada  $u(x)$  para (1) se define por tramos en cada subintervalo  $[x_{i-1}, x_i]$  de  $\Delta$ . Esto es, una función  $u(x)$  es seleccionada teniendo derivadas continuas hasta de grado  $K-1$  en  $[x_0, x_1]$ , i. e.,  $u(x) \in C^{(K-1)}[x_0, x_1]$  y las cuales consisten de  $n$  tramos (1 por cada subintervalo), cada uno de los cuales es un polinomio de grado menor que  $k + m$ . Consideremos un elemento finito  $\Omega_i$  y establezcamos un sistema de coordenados locales  $\xi$ , con origen en el centro del elemento y dividiéndolo para que  $\xi = -1$  en el punto izquierdo y  $\xi = +1$  en el punto derecho, como se muestra en la figura 2.2.1.

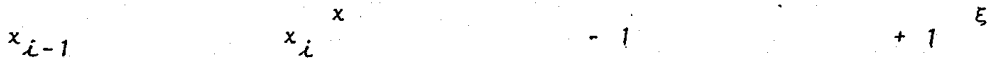


Figura 2.2.1 Coordenadas lineales y normalizadas del elemento finito  $\Omega_i$ .

Esto se realiza gracias a la transformación lineal.

$$\xi = \frac{2x - (x_i + x_{i-1})}{x_i - x_{i-1}} \quad \dots (3)$$

en donde los puntos  $x$  t.q.  $x_{i-1} \leq x \leq x_i$  se transforman en los puntos  $\xi$  t.q.  $-1 \leq \xi \leq +1$ .

Si sustituimos  $u(x)$ ,  $u(x) = \sum_{i=1}^n u_i \phi_i(x)$  en la ecuación original (1), el residuo  $r(u) \equiv Lu(x) - f(x)$  no sería idéntico a cero en el dominio  $\Omega$ , como lo sería si  $u(x)$  fuera la solución exacta de (1), tampoco cumpliría con las condiciones a la frontera  $(c_f)$ .

Supongamos entonces que  $u(x)$  satisface  $c_f$ , lo que es usualmente posible sobre dominios  $\Omega$  no demasiados complicados (como es el de nuestro caso, segmentos de recta), entonces la pregunta a plantear es cómo seleccionar  $\phi_i(x)$ ,  $i = \overline{1, N}$ .

Cuando el problema a tratar es de orden  $>1$ , tales como problemas sobre estructuras o de curvaturas y ciertos problemas hidráulicos, involucran segundas derivadas en las funciones convenientes de ensayo y prueba. Esto implica que las primeras y segundas derivadas deberán ser cuadradas integrables para que ambas función y derivada normal sean continuas. La necesidad de considerar la continuidad  $c^1$  de las funciones base globales se encuentra también en problemas de segundo orden cuando el método de elementos finitos colocación es usado. En esta situación, las familias de polinomios de Hermite de elementos en los cuales las funciones base globales son polinomios por tramos  $c^1$  se hacen presentes.

En la interpolación de Hermite ambos valores de funciones y derivadas de varios órdenes son interpoladas. Entonces la aproximación es suave en el elemento interior, la inter

polación en las derivadas es confinada en los nodos sobre las fronteras de los interelementos para asegurar que la derivada sea continua.

Para esto consideremos el ejemplo simple de interpolación cúbica sobre un elemento unidimensional [16].

Sea  $\Omega_i = [x_{i-1}, x_i]$  un elemento arbitrario en una partición de elemento finito del intervalo  $[x_0, x_1]$ . Un elemento cúbico  $C^1$  se obtiene interpolando la función y las derivadas en los extremos  $x_{i-1}$  y  $x_i$  de cada elemento.

Igual que como se hace en el tratamiento de la interpolación de Lagrange, desarrollaremos funciones base sobre un elemento  $\hat{\Omega}_i$ . El elemento  $\Omega_i$  se transforma linealmente en  $\hat{\Omega}_i = [-1, 1]$  ver figura 2.2.1., por la transformación definida en (3). En  $\hat{\Omega}_i$  con los elementos interpolados  $\hat{u}$  y  $\hat{u}' \equiv \frac{d\hat{u}}{d\xi}$  en los nodos extremos  $j=1, 2$ , la expansión tipo Hermite cúbica tiene la forma

$$\hat{u}(\xi) = \sum_{j=1}^2 \hat{u}_j \hat{\Psi}_j^0(\xi) + \sum_{j=1}^2 \hat{u}'_j \hat{\Psi}_j^1(\xi) \quad \dots (4)$$

donde las funciones base tipo Hermite  $\{\hat{\Psi}_j^0, \hat{\Psi}_j^1\}$  satisfacen las propiedades de interpolación en los nodos extremos  $\xi_1 = -1$ ,  $\xi_2 = 1$ :

$$\begin{aligned} \hat{\Psi}_j^0(\xi_k) &= \delta_{jk} & , & & \hat{\Psi}_j^1(\xi_k) &= 0 \\ \frac{d}{d\xi} \hat{\Psi}_j^0(\xi_k) &= 0 & , & & \frac{d}{d\xi} \hat{\Psi}_j^1(\xi_k) &= \delta_{jk} \end{aligned}$$

para nodos locales  $j=1, 2$  y  $k=1, 2$ .

Usando las propiedades anteriores podemos construir



Los polinomios cúbicos de Hermite directamente como

$$\hat{\psi}_j^0(\xi) = \frac{1}{4} (\xi-1)^2 (\xi+2), \quad \hat{\psi}_2^0(\xi) = \frac{1}{4} (\xi+1)^2 (2-\xi)$$

$$\hat{\psi}_j^1(\xi) = \frac{1}{4} (\xi-1)^2 (\xi+1), \quad \hat{\psi}_2^1(\xi) = \frac{1}{4} (\xi+1)^2 (\xi-1):$$

ver p. ej. [2],

Por medio de la transformación lineal  $\xi \rightarrow x$  podemos transformar las funciones base sobre  $\hat{\Omega}_i$  a las correspondientes funciones base  $\psi_j^{0i}$  y  $\psi_j^{1i}$  sobre  $\Omega_i$ , por lo que las aproximaciones de (4) toman la forma

$$u_h^i(x) = \sum_{j=1}^2 u_j^i \psi_j^{0i}(x) + \frac{h_i}{2} \sum_{j=1}^2 \left( \frac{du}{dx} \right)_j^i \psi_j^{1i}(x)$$

donde se hace uso de la regla de la cadena para expresar  $\left( \frac{d\hat{u}}{d\xi} \right)$  es  $\left( \frac{du}{dx} \right)^i \left( \frac{h_i}{2} \right)$  con  $h_i = x_i - x_{i-1}$ , la longitud del elemento.

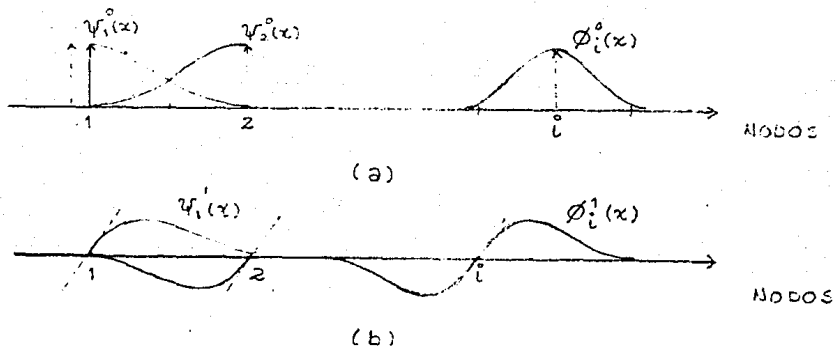


Figura 2.2.2 a) Función base local en los nodos 1 y 2 y función base global en el nodo i, b) Función base local asociada a la 1ra. derivada en los nodos 1 y 2 y función base-global asociada a la 1ra. derivada en el nodo i, valor y derivada de las funciones base globales y locales para cúbicos de Hermite.

### 2.3. SOLUCIÓN DE LA ECUACION DIFERENCIAL ORDINARIA LINEAL DE SEGUNDO ORDEN POR EL METODO DE COLOCACION.

Procedemos a construir un conjunto de funciones base globales usando el siguiente criterio:

1. Las funciones base globales  $\phi_i$  sean generadas por funciones simple definidas por tramos - elemento por elemento - sobre la malla de elementos finitos.

2. Las funciones base globales  $\phi_i$  sean suaves. Esto sugiere el uso de una interpolación mediante polinomios cúbicos de Hermite por tramos.

3. Las funciones base globales  $\phi_i$  sean elegidas de - tal manera que los coeficientes  $u_i$  definidos en la solución -- aproximada  $u$  sean precisamente los valores de  $u(x)$  en los puntos nodales [16].

Una muy buena adecuación de este criterio, es el siguiente conjunto de funciones base globales.

$u_j^{0+}(x)$ ,  $u_{j+1}^{0-}(x)$ ,  $u_j^{1+}(x)$  y  $u_{j+1}^{1-}(x)$  utilizadas - en [2] y que en este trabajo de tesis llamamos.

$$\begin{array}{l}
 u_j^{0+}(x) \longrightarrow \phi_{i-1}(x) \\
 u_{j+1}^{0-}(x) \longrightarrow \phi_i(x) \\
 u_j^{1+}(x) \longrightarrow \phi_{i-1}^1(x) \\
 u_{j+1}^{1-}(x) \longrightarrow \phi_i^1(x)
 \end{array}
 \left. \vphantom{\begin{array}{l} \\ \\ \\ \end{array}} \right\} \dots (5)$$

} Primeras derivadas

Por lo que,  $u(x) = \sum_{i=1}^n u_i \phi_i(x)$  (solución aproxima-

da por tramos sobre cada intervalo) se transforma en

$$u(x) = u_{-1}^{(0)} \phi_{i-1}(x) + u_1^{(0)} \phi_i(x) \\ + u_{-1}^{(1)} \phi_{i-1}^1(x) + u_1^{(1)} \phi_i^1(x) \quad \dots (6)$$

Sustituyendo (6) en (1) obtenemos

$$- a(x) \left\{ u_{-1}^{(0)} \frac{d^2}{dx^2} \phi_{i-1}(x) + u_1^{(0)} \frac{d^2}{dx^2} \phi_{i-1}^1(x) \right. \\ \left. + u_1^{(0)} \frac{d^2}{dx^2} \phi_i(x) + u_1^{(1)} \frac{d^2}{dx^2} \phi_i^1(x) \right\} + \\ + b(x) \left\{ u_{-1}^{(0)} \frac{d}{dx} \phi_{i-1}(x) + u_{-1}^{(1)} \frac{d}{dx} \phi_{i-1}^1(x) \right. \\ \left. + u_1^{(0)} \frac{d}{dx} \phi_i(x) + u_1^{(1)} \frac{d}{dx} \phi_i^1(x) \right\} + \\ + c(x) \left\{ u_{-1}^{(0)} \phi_{i-1}(x) + u_{-1}^{(1)} \phi_{i-1}^1(x) \right. \\ \left. + u_1^{(0)} \phi_i(x) + u_1^{(1)} \phi_i^1(x) \right\} = f(x)$$

Al colocar  $x$  en cada uno de los puntos de Gauss

$$\xi_{ij} = (x_{i-1} + x_i + p_j h_i) / 2 \quad \begin{matrix} i=1, 2, n+1; \\ j=1, 2, ; h_i = |x_i - x_{i-1}| \end{matrix}$$

donde  $p_j$  son las raíces del polinomio de Legendre de grado 2

$$p_1 = - \frac{1}{\sqrt{3}} \quad y \quad p_2 = + \frac{1}{\sqrt{3}}$$

Tendremos que  $x$  toma la siguiente forma  $x_i^j$  donde  $j$  es el índice del elemento e  $i$  es el índice de los puntos de Gauss, entonces volviendo a sustituir (6) en (1) pero ahora evaluando precisamente en  $x_i^j$  se tiene

$$\begin{aligned} - a(x_i^j) & \left\{ u_{-1}^{(0)} \frac{d^2}{dx^2} \phi_{i-1} \right|_{x_i^j} + u_{-1}^1 \frac{d^2}{dx^2} \phi_{i-1}^1 \Big|_{x_i^j} \\ & + u_1^{(0)} \frac{d^2}{dx^2} \phi_i \Big|_{x_i^j} + u_1^1 \frac{d^2}{dx^2} \phi_i^1 \Big|_{x_i^j} \Big\} + \\ + b(x_i^j) & \left\{ u_{-1}^{(0)} \frac{d}{dx} \phi_{i-1} \Big|_{x_i^j} + u_{-1}^1 \frac{d}{dx} \phi_{i-1}^1 \Big|_{x_i^j} \right. \\ & \left. + u_1^{(0)} \frac{d}{dx} \phi_i \Big|_{x_i^j} + u_1^1 \frac{d}{dx} \phi_i^1 \Big|_{x_i^j} \right\} + \\ + c(x_i^j) & \left\{ u_{-1}^{(0)} \phi_{i-1} \Big|_{x_i^j} + u_{-1}^1 \phi_{i-1}^1 \Big|_{x_i^j} \right. \\ & \left. + u_1^{(0)} \phi_i \Big|_{x_i^j} + u_1^1 \phi_i^1 \Big|_{x_i^j} \right\} = \delta(x_i^j) \end{aligned}$$

La ecuación (1) junto con las (6) satisfacen un sistema de  $\phi_2^k$  ecuaciones para  $\phi_i^j$  incógnitas, cuya matriz de coeficientes es casi diagonal (por bloque) como se muestra en la figura 2.3.1 donde  $\phi_2^k$  es igual a

$$\phi_2^k = (k-2) \times NINT + 2$$

Cada bloque es de  $(k-2) \times k$ , excepto para el primero y el último los cuales son de  $1 \times 2$ .

$$\begin{array}{l}
 \text{BLOQUE} \longrightarrow \\
 \\
 \text{BLOQUE} \longrightarrow
 \end{array}
 \left( \begin{array}{cccccccc}
 x & x & & & & & & \\
 x & x & x & x & & & & \\
 x & x & x & x & & & & \\
 & & x & x & x & x & & \\
 & & x & x & x & x & & \\
 & & & & x & x & x & x \\
 & & & & x & x & x & x \\
 & & & & & & x & x & x & x \\
 & & & & & & x & x & x & x \\
 & & & & & & & & x & x
 \end{array} \right)$$

Figura 2.3.1. Estructura de la matriz de la ecuación (1), para el método de colocación.

Observaciones:

1. Al trabajar con una base B-spline sabemos que:

1.1 Las funciones base globales que la forman pertenecen a la clase de funciones  $C^2$ , las cuales son muy útiles en problemas de interpolación y aproximación.

1.2 Además se conocen sus valores en los nodos.

1.3 y permiten

1.3.1 Trabajar con un soporte más local .

1.3.2 Emplear una interpolación de tipo Hermite.

2. Puesto que la aproximación pertenece globalmente a  $C^1$  y el residuo  $r(x)$  involucra segundas derivadas, los puntos -

colocación  $\xi_{ij}$  deben estar en el interior de un elemento  $\Omega_i$ ,  $i = \overline{1, N}$ .

3. Los cálculos se reducen simplemente a evaluar las funciones en los puntos de colocación.
4. Aunque el operador en la ecuación original sólo acopla incógnitas de un sólo elemento, la simetría se pierde en la formulación discreta.
5. Con funciones base globales, es muy frecuente que el sistema de ecuaciones algebraicas sea mal condicionado, es decir, que la solución  $x$  es extremadamente sensible a cualquier perturbación de  $A$  o de  $f$ , donde  $A = (a_{ij})$  es una matriz de orden  $N$  y  $F = (f_i)$  son vectores con  $N$  componentes.

CAPITULO TRES

SOLUCION AL PROBLEMA DE VALOR A LA FRONTERA POR  
EL METODO DE GALERKIN

## SOLUCION AL PROBLEMA DE VALOR A LA FRONTERA POR EL METODO DE GALERKIN

Los datos en nuestro problema deben ser suaves, para que exista una única función  $u$  tal que satisfaga tanto la ecuación diferencial para toda  $x$  en  $\Omega$  como las  $cf$ , de lo contrario se presentan dos problemas, tales como:

1. No existe ninguna solución al problema.

2. Aún existiendo una solución suave, no puede encontrarse una forma cerrada para la complejidad del dominio  $\Omega$ , los coeficientes y  $cf$ .

La dificultad es que nuestra condición de que la solución  $u$  satisfaga la ecuación (1) para todo punto  $x \in \Omega$ , es bastante fuerte.

Por lo que debemos de reformular nuestro problema de tal manera que admita condiciones menos fuertes, *i.e.*, que admita condiciones débiles sobre la solución y sus derivadas. Tales reformulaciones son llamadas formulaciones variacionales o débiles del problema y son utilizadas para acomodar datos y soluciones irregulares. Existiendo la solución suave para el problema, existe también la solución de la forma débil del problema. Por lo tanto, no perdemos nada en reformular el problema para la formulación débil. Por lo que, la formulación variacional o débil para problemas de valor a la frontera es precisamente la formulación que usaremos para construir las aproximaciones en los elementos  $\Omega_i$  finitos,  $i = \overline{1, N}$ .

El argumento esencial para la reformulación es como sigue:



Tomemos la formulación basada en el método de residuos pesados e integremos por partes, i.e.,

$$r(x) = - [a(x)u'(x)]' + b(x)u'(x) + c(x)u(x) - f(x) = 0$$

integrando para cualquier  $w \in H_0^1$  tenemos.

$$\int_{\Omega} r w \, dx = \int_{\Omega} ([-au']' w + b u' w + c u w - f w) \, dx = 0 \quad \dots (7)$$

e integrando por partes se tiene

$$\int_{\Omega} (a u' w' + b u' w + c u w - f w) \, dx = 0 \quad \forall w \in H_0^1 \quad \dots (8)$$

expresión que llamaremos formulación variacional o débil del problema. En (8) la función de peso  $w(x)$  o función de prueba, es cualquier función de  $x$  que se comporta bien.

La formulación débil de nuestro problema, aunque aparentemente menos directa que la formulación fuerte, es de gran ayuda en problemas físicos. En fenómenos físicos, es frecuentemente útil para medir (o por lo menos considerar la medida de) los datos y/o la solución de un problema de valor a la frontera. Por lo que cualquier medida real deseada (en mecánica, en electricidad y en termoelectrónica) tendrá una dimensión finita, estas cantidades pueden, ser determinadas solamente en algún sentido en forma de promedios sobre algunas regiones pequeñas y no para cualquier punto en particular.

En esta fase, existen dos puntos que deberían ser apreciados cuidadosamente:

### 1. La formulación débil

$$\int_{\Omega} (aw)' u' + bwu' + cwu - f(w) \, dx = 0, \quad w \in H_0^1$$

es tan válida y significativa como la formulación fuerte (1) - desde luego, la solución de (1) es también solución de (8), y - de hecho, es la única solución de (8).

2. La especificación del conjunto  $H$  de funciones de prueba es una característica esencial de la formulación débil.

Por que existe otro conjunto  $\bar{H}$  de funciones de prueba (ensayo) diferente al conjunto  $H$  de funciones de prueba. Nuestros requisitos de suavidad demandan que consideremos ambos - conjuntos  $H$  y  $\bar{H}$ , p. ej.  $u \in \bar{H}$  la cual tiene la propiedad de que su segunda derivada, cuando se multiplica por una función de prueba  $w$ , produce una función  $\frac{d^2 u}{dx^2} w$  la cual es integrable sobre  $p < x < q$ . De otra manera,  $\frac{d^2 u}{dx^2}$  ninguna derivada de estas funciones de prueba aparecen en (8).

Obtenemos una formulación débil de (1) observando que si  $u$  y  $w$  son funciones suficientemente suaves, entonces emplean la integración por partes se tiene que:

$$\int_{\Omega} a(x) w(x) \frac{d^2 u}{dx^2} dx = - \int_{\Omega} a(x) w(x) \frac{d^2 u}{dx^2} dx$$

$$= - \left\{ a w u' \Big|_{\Omega} - \int_{\Omega} (a w)' u' dx \right\}$$

Si continuamos demandando que las funciones de prueba sean igual a cero en los puntos extremos, entonces.

$$- \int_{\Omega} a(x) w(x) \frac{d^2 u}{dx^2} dx = \int_{\Omega} (a w)' u' dx \quad \dots (9)$$

para toda función de prueba  $w$ , y por lo tanto (8) puede ser -- reemplazado por el siguiente problema variacional:

encontrar  $u \in H_0^1$  tal que

$$\int_{\Omega} (aw)'u' + bwu' + cwu - f(w) dx = 0, \quad \forall w \in H_0^1$$

donde el mismo orden de derivación de ambos conjuntos aparecen y por consiguiente tomamos  $H = \bar{H} = H_0^1$

En vista de estas observaciones es natural plantearse la siguiente pregunta cómo determinar la solución aproximada para (8)?

Para esto, a continuación daremos la definición del espacio  $H_0^1$ :

Es el espacio lineal de funciones de dimensión infinita, que valen cero en los puntos extremos (i.e.,  $w(p) = w(q) = 0$ ) y cuya función y sus primeras derivadas son de cuadrado integrable sobre  $\Omega$ .

Y precisamente por esta definición y considerando la formulación débil, supondremos que cualquier función en  $H_0^1$  y en particular  $w(x)$ , se puede expresar en términos de

$$w(x) = \sum_{i=1}^{\infty} u_i \phi_i(x)$$

Supongamos que damos un conjunto infinito de funciones

$$\{ \phi_1(x), \phi_2(x), \phi_3(x), \dots \text{ en } H_0^1 \}$$

las cuales tienen la propiedad de que cada función de prueba  $w \in H_0^1$  puede ser representada como una combinación lineal de las  $\phi_i(x)$ , por una serie del tipo (10). Es claro que al tomar

un número finito  $N$  de la serie, obtenemos una aproximación  $w_n$  de  $w$ ,  $i, e.,$

$$w_n(x) = \sum_{i=1}^N u_i \phi_i(x_i) \quad \dots (11)$$

en donde  $[\phi_i]$  es el conjunto de funciones base globales y define un subespacio de dimensión finita  $N$ .

$H_0^{(N)}$  es un subespacio de  $H_0^1$  porque cada  $\phi_i$ ,  $i=1, 2, N$ , es por definición, un elemento de  $H_0^1$ .

El subespacio  $H_0^{(n)}$  es de dimensión finita  $N$  porque cada función  $w_n$  en  $H_0^{(N)}$  está determinada por una combinación lineal de sólo  $N$  funciones  $\phi_1, \dots, \phi_n$ .

(Suponemos que las  $N$  funciones  $\phi_1, \dots, \phi_N$  son linealmente independientes, es decir que es imposible encontrar  $N$  coeficientes  $u_1, \dots, u_n$ , no todos igual a cero, tales que  $-\sum_{i=1}^N u_i \phi_i(x_i) = 0$ ,  $x$ ).

Por consiguiente, nuestro problema en términos de esta descripción finalmente queda expresado de la siguiente forma

$$\int_{\Omega} (a w_n)' u_n' + b w_n u_n' + c w_n u_n dx = \int_{\Omega} f w_n dx, \quad w_n \in H_0^{(N)} \quad \dots (12)$$

Entonces las  $\phi_i$  son conocidas,  $u_n'$  serán determinadas completamente una vez que las  $N$  coeficientes  $u_i$  se determinen. Los  $u_i$  en (11) se refieren a los grados de libertad de la aproximación.

Observemos que todas las funciones de prueba  $w_n$  son combinación lineal de las funciones base globales  $\phi_i$  de la forma (11) y los  $u_i$  son constantes en principio arbitrarios.

Además  $w_n$  en (11) puede tomar los valores de cualquier función en  $H_0^{(N)}$  a través de una elección apropiada de las constantes  $u_i$ . Para determinar los valores específicos  $u_i$  en la aproximación (11) desarrollaremos (12) y factorizaremos los coeficientes  $u_i$  teniendo

$$\sum_{j=1}^N \left\{ \int_{\Omega} [a(x) \phi_i'(x) \phi_j'(x) + (b(x) \phi_i'(x) \phi_j'(x) + c(x) \phi_i(x) \phi_j(x))] u_j dx = \int_{\Omega} f(x) \phi_i(x) dx ; \right. \quad \dots(13)$$

$$p. \text{t. } i = \overline{1, N} \quad \text{y donde } \phi_j'(x) = \frac{d}{dx} \phi_j(x).$$

Pasando (13) a una forma compacta tenemos:

$$\sum_{j=1}^N a_{ij} u_j = f_i ; \quad i = \overline{1, N} \quad \dots(14)$$

El arreglo rectangular  $A = (a_{ij})$  de  $N \times N$  se refiere a la matriz del problema (12) y el vector columna  $F = (f_i)$  de  $N \times 1$  se refiere al vector empleado en dicha ecuación.

Por lo que tendremos el sistema de  $N$  ecuaciones lineales con  $N$  incógnitas  $u_i$

$$\sum_{j=1}^N a_{ij} u_j = f_i \quad i = \overline{1, N} \quad \dots(15)$$

Las funciones  $\phi_i$  han sido elegidas para ser independientes, por lo que las  $N$  ecuaciones serán independientes, y por lo tanto la matriz  $A$  será invertible. Los coeficientes  $u_i$  son determinados únicamente por (15) y son de la forma

$$u_j = \sum_{i=1}^N (a_{ij}^{-1} b_i) \quad \dots (16)$$

donde  $a_{ij}^{-1}$  son los elementos de la inversa de A. Por lo que la solución aproximada  $w_n$  se determina sustituyendo (16) en (11).

Es importante notar que la calidad de la aproximación está completamente determinada por la elección de nuestras funciones  $\phi_i$ .

### 3.1. APLICACION DE LA TEORIA DEL METODO DE ELEMENTOS FINITOS DE TIPO GALERKIN A NUESTRO PROBLEMA DE VALOR A LA FRONTERA.

Consideraremos el siguiente conjunto de elementos finitos  $\Omega_i$

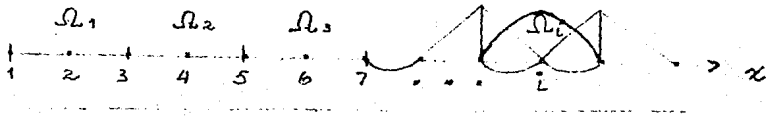


Figura 3.2.1 Malla de elementos finitos de tipo cuadrática compuesta de polinomios interpolantes de LAGRANGE.

Y tomemos el elemento finito  $\Omega_i$  en coordenadas normalizadas tal que

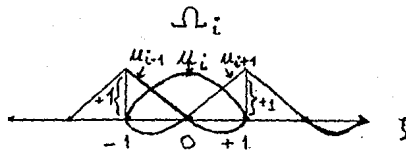


Figura 3.2.2 Valor de las funciones base globales para polinomios cuadráticos de LAGRANGE.

entonces  $u$  puede escribirse como

$u = u_{i-1}(\xi) u_{-1} + u_i(\xi) u_0 + u_{i+1}(\xi) u_{+1}$  en donde  $u_{i-1}$ ,  $u_i$  y  $u_{i+1}$  son las funciones base globales en  $\Omega_i$ , por lo que  $u(x)$  se comporta como

$$u(x) \sim u_h(x) = \sum_{i=1}^N u_i \mu_i(x)$$

entonces por definición de residuo

$$R[u_h(x)] := -a(x) \frac{d^2}{dx^2} u_h + b(x) \frac{du_h}{dx} + c(x) u_h - f$$

y aplicando el método de residuos pesados tenemos que

$$\int_{\Omega} w(x) R[u_h(x)] dx = 0$$

sea  $w(x) = u_i(x)$ ,  $i = \overline{1, N}$  y de la ecuación (7) tenemos que

$$\int_{\Omega} u_i(x) \left\{ -a(x) \frac{d^2}{dx^2} \sum_{j=1}^N u_i u_j(x) + b(x) \frac{d}{dx} \sum_{j=1}^N u_j u_j(x) + c(x) \sum_{j=1}^N u_j u_j(x) - f(x) \right\} dx = 0, \quad i = \overline{1, N}$$

factorizando  $u_j$  tenemos

$$\sum_{j=1}^N \int_{\Omega} \left\{ -a(x) u_i(x) \frac{d^2}{dx^2} u_j(x) + b(x) u_i(x) \frac{d}{dx} u_j(x) + \right.$$

$$\left. c(x) u_i(x) u_j(x) \right\} u_j dx$$

$$= \int_{\Omega} u_i(x) f(x) dx, \quad i = \overline{1, N}$$

La integral sobre todo el dominio  $\Omega$  es igual a la suma de las integrales en cada elemento  $\Omega_i$  de  $\Omega$ , entonces -

$$\int_{\Omega} a(x) u_i(x) \frac{d^2}{dx^2} u_i(x) dx = \left( a u_i' u_j' \right)_{\Omega} - \int_{\Omega} (a u_i)' u_j' dx$$

integrando por partes y por (9) se tiene

$$-\int_{\Omega} a(x) u_i(x) \frac{d^2}{dx^2} u_j(x) dx = \int_{\Omega} (a u_i)' u_j' dx \quad y,$$

la integral del lado derecho se evalúa en  $x_{i-1}$ ,  $x_i$ , es decir, -

$$\int_{\Omega} (a u_i)' u_j' dx = \int_{x_{i-1}}^{x_i} a (u_i)' u_j' dx = \int_{x_{i-1}}^{x_i} \left[ \frac{d}{dx} (a u_i) \right] \left[ \frac{d}{dx} u_j \right] dx$$

Considerando nuevamente la transformación lineal

$$\xi = \frac{2x - (x_i + x_{i-1})}{\Delta x}$$

y sabiendo que  $dx = \frac{\Delta x}{2} d\xi$  se tiene sustituyendo

$$\int_{x_{i-1}}^{x_i} \left[ \frac{d}{dx} (a u_i) \right] \left[ \frac{d}{dx} u_j \right] dx = \frac{2}{\Delta x} \int_{-1}^{+1} \left[ \frac{d}{d\xi} (a u_i) \right] \left[ \frac{d}{d\xi} u_j \right] d\xi$$

De  $(a u_i)' = a u_i' + a' u_i$  se tiene

$$\int_{-1}^{+1} \left[ \frac{d}{d\xi} (a u_i) \right] \left[ \frac{d}{d\xi} u_j \right] d\xi = \int_{-1}^{+1} a \left[ \frac{d}{d\xi} u_i \right] \left[ \frac{d}{d\xi} u_j \right] d\xi +$$

$$\int_{-1}^{+1} a' u_j \left[ \frac{d}{d\xi} u_j \right] d\xi$$

Para evaluar las integrales para los coeficientes  $a$ ,  $b$ ,  $c$  y  $f$ , empleamos una integración numérica de Simpson para la cual



$$\int_{-1}^{+1} a \left[ \frac{d}{d\xi} u_i \right] \frac{d}{d\xi} u_j \, d\xi = \sum_{k=1}^3 w_k a(x(\xi_k)) u_i'(x(\xi_k)) u_j'(x(\xi_k)) \frac{2}{\Delta x}$$

donde  $w_k$  y  $\xi_k$  son los pesos y abscisas respectivamente.

$$\begin{aligned} \int_{\Omega_i} b(x) u_i(x) \frac{d}{dx} u_j(x) dx &= \int_{x_{i-1}}^{x_i} b(x) u_i(x) \frac{d}{dx} u_j(x) dx \\ &= \int_{-1}^{+1} b(\xi(x)) u_i(\xi(x)) \frac{d}{d\xi} u_j(\xi(x)) d\xi \\ &= \sum_{k=1}^3 w_k b(x(\xi_k)) u_i(\xi_k) \frac{d}{d\xi_k} u_j(\xi_k) \\ &= \sum_{k=1}^3 w_k b(x(\xi_k)) \delta_{ik} \frac{d}{d\xi_k} u_j(\xi_k) = w_i b(x(\xi_i)) \frac{d}{d\xi_i} u_j(\xi_i), \quad i=1,3, \end{aligned}$$

para el coeficiente  $c$  tenemos

$$\begin{aligned} \int_{\Omega_i} c(x) u_i(x) u_j(x) dx &= \int_{x_{i-1}}^{x_i} c(x) u_i(x) u_j(x) dx \\ &= \Delta \frac{x}{2} \int_{-1}^{+1} c(\xi(x)) u_i(\xi(x)) u_j(\xi(x)) d\xi, \\ &= \sum_{k=1}^3 w_k c(x(\xi_k)) u_i(\xi_k) u_j(\xi_k) \\ &= \sum_{k=1}^3 w_k c(x(\xi_k)) \delta_{ik} \delta_{jk} = w_i c(x(\xi_i)) \delta_{ij}, \quad i=1,3, \quad j=1,3, \end{aligned}$$

$$y \int_{\Omega_i} u_i(x) f(x) dx = \int_{x_{i-1}}^{x_i} u_i(x) f(x) dx$$

$$= \Delta \frac{x}{2} \int_{x_{i-1}}^{x_{i+1}} u_i(s(x)) \delta(s(x)) ds$$

$$= \sum_{k=1}^3 w_k u_i(s_k) \delta(x(s_k))$$

$$= \sum_{k=1}^3 w_k \delta_{ik} \delta(x(s_k)) = w_i \delta(x(s_i)) , \quad i=1,3,$$

Por último de (14) se tiene  $A \underline{u} = \underline{F}$  en donde

$A$  es la matriz cuya dimensión es  $N \times N$  ( $a_{ij}$ )  $\begin{matrix} i=1, N \\ j=1, N \end{matrix}$

$\underline{u}$  es el vector incógnita cuya dimensión es  $N \times 1$  ( $[u_1, u_2, \dots, u_N]^T$ )

$\underline{F}$  es el vector cuya dimensión es  $N \times 1$  ( $[\delta_1, \delta_2, \dots, \delta_N]^T$ )

CAPITULO CUATRO  
SOLUCION AL PROBLEMA DE VALOR A LA FRONTERA POR EL  
METODO DE COLOCACION - GALERKIN

SOLUCIÓN AL PROBLEMA DE VALOR A LA FRONTERA POR  
EL MÉTODO DE COLOCACION - GALERKIN

#### 4.1. PRELIMINARES

La conexión entre los métodos de elementos finitos de tipo colocación y Galerkin está en considerar el método de residuos pesados. Como se vio en el capítulo anterior, el método de Galerkin se obtiene cuando las funciones de peso se eligen como las funciones base.

Cuando la función Delta de Dirac se elige en lugar de las funciones de peso, el método que se obtiene es el de COLOCACION. Por otra parte, tenemos que en ciertos problemas los coeficientes en las ecuaciones tienen discontinuidades de primer orden en varios puntos en el dominio  $\Omega$ . Provocando éstas a su vez discontinuidades en las segundas derivadas de la solución en ciertos puntos.

Eligiendo funciones base de tipo Hermite, en general, aseguramos la continuidad en estos puntos. Por consiguiente, las funciones base de tipo Hermite se eligen muy bien en problemas con discontinuidades en los coeficientes, de aquí que, ellas permitan saltos en las segundas derivadas. En cálculos prácticos es posible usar funciones base cúbicas colocando el punto de discontinuidad en el soporte entre los elementos de cualquier tamaño y usando puntos de colocación en todos los elementos.

Es importante por supuesto evitar localizar los puntos de colocación correctos en los puntos donde hay discontinuidad en los coeficientes para cualquier método (Ver p. ej. [2]).

Dicho lo anterior, debemos recordar que en la búsqueda de una solución aproximada tenemos la posibilidad de usar diferente tipo de interpolación, p. ej., interpolación cúbica mediante polinomios interpolantes de Hermite si el método a usarse es el de colocación.

En este capítulo trataremos otra variante del método de colocación en el cual la continuidad se trata en una forma semejante al método de Galerkin y con elementos cuadráticos de LAGRANGE, se trata del método híbrido COLOCACION-GALERKIN.

#### 4.2. DESCRIPCION DEL METODO DE COLOCACION - GALERKIN

Comenzaremos desarrollando el método retomando el problema de nuestro interés. La formulación variacional del problema es buscar  $u \in H_0^1$  tal que

$$\int_{\Omega} ((a w)'' u' + b w u' + c w u - f w) dx = 0, \quad w \in H_0^1$$

Definimos una discretización del dominio consistiendo de  $N$  elementos y construyendo un subespacio de elementos finitos  $H_0^{(N)}$  de funciones de prueba usando funciones base de tipo LAGRANGE.

La aproximación de elementos finitos  $w_h \in H_0^{(N)}$  es de la forma usual

$$w_h(x) = \sum_{i=1}^N u_i \varphi_i(x) \dots \dots \dots (17)$$

donde  $\varphi_i(x)$ ,  $i=1, N$  son las funciones base globales

Sea  $\{X_k\}$  con  $k=1, N$  la serie correspondiente a los  $N-1$  nodos entre los elementos en la discretización y sea  $\{\varphi_k\}$  la familia de funciones base tal que

$$\phi_k(x_m) = \begin{cases} 0 & \text{si } m \neq k \\ 1 & \text{si } m = k \end{cases}$$

Las funciones de prueba en la formulación variacional pertenecen a  $H_0^1$ . Para el problema discreto con  $w_h$  dada por (17) y seleccionando

$$u_h = \sum_{k=1}^N u_k \phi_k$$

para cada  $R$  obtenemos un conjunto de  $N$  ecuaciones asociadas con los  $N$  nodos, i. e.,

$$\sum_{k=1}^N \left\{ \int_{\Omega} [a(x) \phi_i(x) \phi_k'(x) + (b(x) \phi_i(x) \phi_k'(x)) + (c(x) \phi_i(x) \phi_k(x))] dx \right\} u_k$$

$$= \int_{\Omega} f(x) \phi_i(x) dx \quad \text{p.t. } u_i, \quad i = 1, N$$

y separando tenemos

$$\sum_{k=1}^N \left\{ \int_{\Omega} [a(x) \phi_i'(x) \phi_k'(x) + (b(x) \phi_i(x) \phi_k'(x)) + (c(x) \phi_i(x) \phi_k(x))] dx \right\} u_k$$

$$= \int_{\Omega} f(x) \phi_i(x) dx; \quad i = 1, \dots, N-1 \quad \dots \dots (18)$$

Las restricciones de  $\phi_i$  y  $\phi_k$  para  $\Omega_i$ , son las correspondientes funciones lineales  $\psi_i^i$  y  $\psi_k^i$ , por lo que (18) toma la forma

$$\sum_{k=1}^N \left\{ \int_{\Omega} [a(x) \psi_i^i(x) \psi_k^i(x) + (b(x) \psi_i^i(x) \psi_k^i(x)) + (c(x) \psi_k^i(x))] dx \right\} u_k$$

$$= \int_{\Omega} f(x) \psi_i^i(x) dx; \quad i = \overline{1, N}$$

dando la matriz de coeficientes del problema (12).

Aquí  $i = \overline{1, N_e}$  son los números de los nodos locales - por elemento  $\Omega_i$  y  $k = 1, N_i$  identifican los nodos extremos.

Para la parte de COLOCACION de la forma débil:

Sea  $H_0^{(N)}$  el subespacio de funciones  $u$  de prueba tal - que  $u \in C^\infty(\Omega)$  y  $u(p) = 0 = u(q)$  &  $u(x_k) = 0$  para el  $k$ -ésimo nodo.

Entonces de la expresión (8) tenemos

$$\begin{aligned} & \int_{\Omega} [ -a(x)w(x) \frac{d^2 u}{dx^2} + b(x)w(x) \frac{d u}{dx} + c(x)w(x) u ] dx \\ &= \sum_{i=1}^N \int_{\Omega_i} [ -a(x)w(x) \frac{d^2 u}{dx^2} + b(x)w(x) \frac{d u}{dx} + c(x)w(x) u ] dx \end{aligned}$$

Como en el método de colocación, supongamos que el residuo  $R(u_n)$  es una función suave y que  $u$  es una funcional lineal en  $R$  tal que

$$R(u_n)(x_k) = 0$$

siendo  $x_k$  un punto central del elemento  $k$ -ésimo.

Por lo tanto, las ecuaciones de colocación resultantes para un elemento  $\Omega_i$  son simplemente.

$$\sum_{i=1}^{N_e} \{ [ a(x_k) \psi_i'(x_k) ] + b(x_k) \psi_i'(x_k) + c(x_k) \psi_i(x_k) \} u_i = f(x_k)$$

p.t.  $k = \overline{1, N_{ELE}}$

... (19)

Por lo que las ecuaciones (18) y (19), juntas con los valores a la frontera, constituyen el sistema algebraico final. Las contribuciones del elemento para estas ecuaciones pueden -- ser desarrolladas directamente como en el método de Galerkin.

#### 4.3. APLICACION DE LA TEORIA DEL METODO DE ELEMENTOS FINITOS DE TIPO COLOCACION-GALERKIN A NUESTRO PROBLEMA DE VALOR A LA FRONTERA.

$u(x)$  es una función de interpolación que en nuestro caso particular, al nivel de cada elemento, se expresa como sigue:

$$u(x) = u_{i-1} u_{i-1}(x) + u_i u_i(x) + u_{i+1} u_{i+1}(x)$$

en donde  $u_{i-1}$ ,  $u_i$ ,  $u_{i+1}$  son las funciones base globales en  $\Omega_i$ , por lo que  $u(x)$  se comporta como

$$u(x) \sim u_N(x) = \sum_{i=1}^N u_i u_i(x)$$

entonces por definición de residuo tenemos

$$R[u_N(x)] = -a(x) \frac{d^2 u_N}{dx^2} + b(x) \frac{d u_N}{dx} + c(x) u_N - F(x)$$

Para la parte de colocación:

$R[u_N(x_k)] = 0$ ,  $R = 1$ , NELE siendo  $x_k$  el punto central del elemento  $k$ -ésimo, sustituyendo  $u_N(x_k)$  en el problema de -- nuestro interés tenemos

$$-a(x_k) \left. \frac{d^2 u_N}{dx^2} \right|_{x_k} + b(x_k) \left. \frac{d u_N}{dx} \right|_{x_k} + c(x_k) u_N(x_k) = f(x_k)$$

$$-a(x_k) \sum_{i=1}^N u_i \left\{ \left. \frac{d^2 \phi_i}{dx^2} \right|_{x_k} \right\} + b(x_k) \sum_{i=1}^N u_i \left\{ \left. \frac{d \phi_i}{dx} \right|_{x_k} \right\} + c(x_k) \sum_{i=1}^N u_i \phi_i(x_k) = f(x_k)$$



Para la parte de Galerkin:

Aplicando el método de residuos pesados tenemos que

$$\int_{\Omega} w(x) R[u_N(x)] dx = 0$$

sea  $w(x) = v_i$ ,  $i = \overline{1, N}$  pero sin considerar la asociada al punto central del elemento y de (7) tenemos que

$$\int_{\Omega} v_i(x) \left\{ -a(x) \frac{d^2}{dx^2} \sum_{i=1}^N u_i u_i(x) + b(x) \frac{d}{dx} \sum_{i=1}^N u_i u_i(x) + c(x) \sum_{i=1}^N u_i u_i(x) - f(x) \right\} dx = 0$$

Factorizando  $u_i$

$$\begin{aligned} & \sum_{i=1}^N \int_{\Omega} v_i \left\{ -a(x) v_i(x) \frac{d^2}{dx^2} u_i(x) + b(x) u_i(x) \frac{d}{dx} u_i(x) + c(x) v_i(x) u_i(x) \right\} u_i dx \\ & = \int_{\Omega} v_i(x) f(x) dx \quad i = \overline{1, N} \end{aligned}$$

Dado que la integral sobre todo el dominio  $\Omega$  es igual a la suma de las integrales en cada elemento  $\Omega_i$  de  $\Omega$ .

$$-\int_{\Omega} a(x) v_i(x) \frac{d^2}{dx^2} u_i(x) = - \left\{ a v_i u_i' \Big|_{\Omega} - \int_{\Omega} (a v_i)' u_i' dx \right\}$$

integrando por partes y continuando demandando que las funciones de prueba sean igual a cero en los puntos extremos, entonces

$$= \int_{\Omega} (a v_i)' u_i' dx$$

y la integral del lado derecho se evalúa en  $x_{i-1}$ ,  $x_i$ ,

$$\begin{aligned}
 \int_{\Omega} (av_i)' u_i' dx &= \int_{x_{i-1}}^{x_i} (av_i)' u_i' dx \\
 &= \int_{x_{i-1}}^{x_i} \left[ \frac{d}{dx} (a v_i) \right] \left[ \frac{d}{dx} u_i \right] dx \\
 &= \frac{2}{\Delta x} \int_{-1}^{+1} \left[ \frac{d}{d\xi} (a v_i) \right] \left[ \frac{d}{d\xi} u_i \right] d\xi
 \end{aligned}$$

esto

Por usar nuevamente la transformación  $\xi$  y sabiendo que

$$dx = \frac{\Delta x}{2} d\xi$$

De  $(av)' = av' + a'v$  se tiene

$$\int_{-1}^{+1} \left[ \frac{d}{d\xi} av_i \right] \left[ \frac{d}{d\xi} v_i \right] d\xi = \int_{-1}^{+1} a \left[ \frac{d v_i}{d\xi} \right] \left[ \frac{d u_i}{d\xi} \right] d\xi + \int_{-1}^{+1} a' \left[ \frac{d u_i}{d\xi} \right] d\xi$$

Para evaluar la integral del primer sumando empleamos nuevamente una integración numérica de Simpson para la cual tenemos

$$\sum_{k=1}^3 w_k a(x(\xi_k)) v_i'(x(\xi_k)) u_i'(x(\xi_k)) \frac{2}{\Delta x}$$

donde  $w_k$  y  $\xi_k$  los pesos y abscisas respectivamente.

TERCERA PARTE

CAPITULO CINCO

DESCRIPCION DE LOS PROGRAMAS DE COMPUTO

## DESCRIPCION DE LOS PROGRAMAS DE COMPUTO

## 5.1. INTRODUCCION

Para resolver el problema de valor a la frontera planteado en este trabajo de tesis por los métodos de elementos finitos de tipo COLOCACION, GALERKIN Y COLOCACION - GALERKIN, se desarrollaron los siguientes programas de cómputo:

1. COLFEM
2. GOLFEM
3. COLGAL

Estos programas están escritos en lenguaje FORTRAN IV y los cálculos fueron realizados en una computadora B7800 con - 11 dígitos en precisión sencilla.

Los programas mencionados contienen un programa principal, un subprograma BLOCK DATA para las declaraciones, 12 subrutinas y 2 funciones. El diagrama de flujo del programa principal se muestra en la figura 5.1.1., se tiene un subprograma - BLOCK DATA para cada uno de los métodos de elementos finitos.

BLOCK DATA COLOCACION contiene las evaluaciones numéricas siguientes: XG son las raíces del polinomio de LEGENDRE - de grado 2. U son las integrales de las funciones base locales en coordenadas normalizadas.

$U_i(I, J)$  son las primeras derivadas de las funciones base locales evaluadas en los puntos de Gauss.

$U0(I, J)$  son valores de las funciones base locales evaluadas en los puntos de Gauss.

$U2(I, J)$  son las segundas derivadas de las funciones base locales evaluadas en los puntos de Gauss.

BLOCK DATA GALERKIN contiene las evaluaciones numéricas siguientes:

$UE$  son los nodos de los elementos.

$UIS$  son las integrales de las funciones base por la regla de Simpson.

$UIJS$  Son las integrales de los productos de las funciones base.

$UXIJS$  son las integrales de los productos de las primeras derivadas de las funciones base.

BLOCK DATA COLOCACION - GALERKIN contiene las evaluaciones numéricas siguientes:

$UE$  son los nodos de los elementos.

$UIS$  son las integrales de las funciones base por la regla de Simpson.

$UXIJS$  son las integrales de los productos de las primeras derivadas de las funciones base.

$UX2IJS$  son las integrales de los productos de las segundas derivadas de las funciones base.

$UIJS$  son las integrales de los productos de las funciones base.

Los programas de cómputo COLFEM, GOLFEM y COLGAL han sido estructurados con las subrutinas y funciones siguientes:

#### SUBRUTINAS

##### BDSOLN

Resuelve el sistema de ecuaciones lineales dando la banda de la matriz no simétrica, el vector, el orden y el ancho de la banda.

##### BDFAC

Calcula los elementos de la matriz L y de la matriz-u.

##### BDSOL

Comprende las sustituciones hacia adelante y hacia atrás.

##### CALDIF

Calcula la matriz A representativa del método.

##### CALSOU

Calcula el vector fuente.

##### CMCTRL

Controla el almacenaje disponible para la memoria.

##### DATIT

Provee la fecha corriente (mes / día / año)

**GRAFOX**

Permite graficar la solución numérica de la ecuación -  
diferencial.

**INPSTA**

Lee los datos de entrada (el diagrama de flujo corres-  
pondiente se muestra en la figura 5.1.2.)

**MINMAX**

Define las cotas para la subrutina GRAFOX.

**OUTPUT**

Imprime los resultados de la solución numérica para la  
malla seleccionada.

**STATIC**

Resuelve el sistema algebraico para el método particu-  
lar seleccionado (el diagrama de flujo correspondiente  
se muestra en la figura 5.1.3)

**FUNCIONES****DT**

Permite estimar el tiempo de procesamiento.

**UIJ**

Evalda las funciones base globales para la variable -  
dependiente en cada posición.

## 5.2. FORMATO DE ENTRADA PARA EL CALCULO ESTATICO

Los datos siguientes se requieren y las características en paréntesis describen el formato en FORTRAN IV del número de campos.

HED (12A6) Encabezado.

Título del problema

NLBC, NRBC (Formato libre de entrada)

NLBC condición a la frontera a la izquierda

NRBC condición a la frontera a la derecha.

$x(1), x(2), \dots, x(NPT)$  (formato libre de entrada)

Cada una de las coordenadas en los puntos de la malla de la condición a la frontera a la izquierda a la condición a la frontera a la derecha.

$x(1), x(NPT)$  (Formato libre de entrada)

Son las coordenadas en los puntos de la malla para las fronteras izquierda y derecha respectivamente.



Figura 5.1.1. (Programa principal)

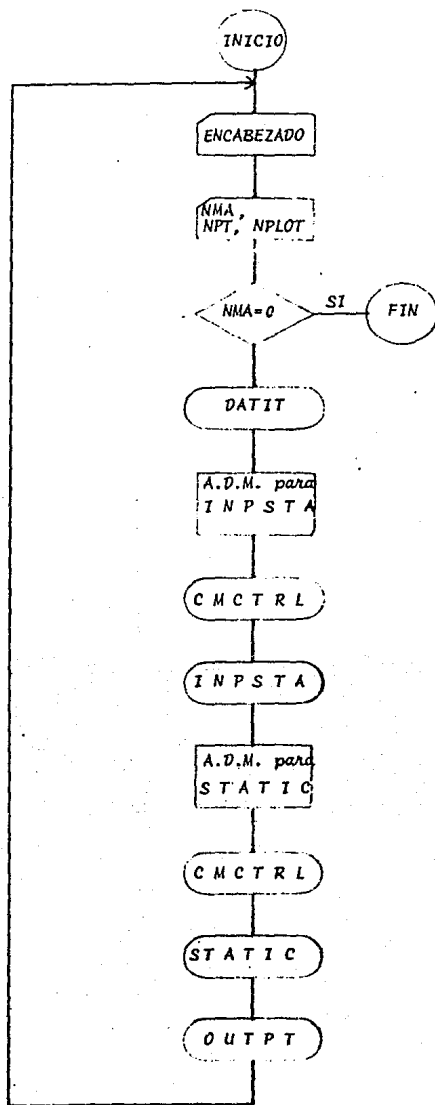


Figura 5.1.2. (Subrutina INPSTA)

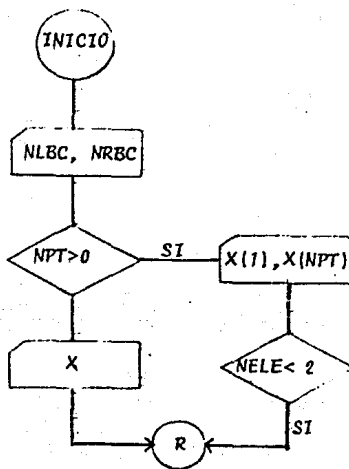
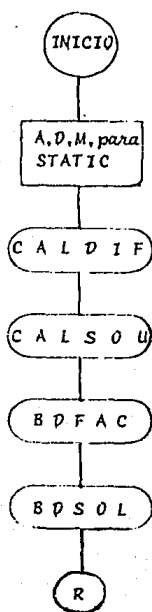


Figura 5.1.3. (Subrutina STATIC)



CUARTA PARTE

CAPITULO SEIS

TIPOS DE EJERCICIOS NUMERICOS ESTUDIADOS

## TIPOS DE EJERCICIOS NUMERICOS ESTUDIADOS

## 6.1. INTRODUCCION

Los ejercicios numéricos fueron elegidos para tratar - los siguientes puntos:

1) Determinar si el orden de convergencia se confirma en la práctica al menos para ejercicios numéricos con solución suave.

2) Comparar la precisión de los métodos usando el mismo número de subintervalos para un problema dado.

## 6.2. EJERCICIOS NUMERICOS

$$6.2.I. - (a u^1)^1 = 2[1 + \alpha (x-\bar{x}) (\arctan \alpha (x-\bar{x}) + \arctan \alpha \bar{x})]$$

dominio:  $\Omega = (0,1)$   $x \in \Omega$

$$c.f. \quad u(0) = u(1) = 0$$

donde:  $a(x) = \frac{1}{\alpha} + \alpha (x-\bar{x})^2$ , y  $\alpha$ ,  $\bar{x}$  son parámetros

$$\alpha > 0, \quad 0 < \bar{x} < 1.$$

Solución exacta:  $u(x) = (1-x) (\arctan \alpha (x-\bar{x}) + \arctan \alpha \bar{x})$ .

$$6.2.II. \quad u'' + a^2 u = -a^2 \cos^2 \pi x - 2\pi^2 \cos 2\pi x$$

dominio:  $\Omega = (0,1)$   $x \in \Omega$

$$c.f. \quad u(0) = u(1) = 0$$

donde  $a$  es un parámetro

Solución exacta:  $u(x) = \frac{[\exp\{a(1-x)\} + \exp\{a x\}]}{(1 + \exp\{a\})} - \cos^2 \pi x$

Este problema con  $a = 20$  fue introducido por STOER y -  
BULIRSCH (Ver. p. ej. [3]).

$$6.2.III. \quad - \frac{d^2 u}{d x^2} + u = (1 + \pi^2) \cos \pi x$$

dominio:  $\Omega = (0, 1)$   $x \in \Omega$

c.f.:  $u(0) = u(1) = 0$

Solución exacta:  $u(x) = a \exp\{-x\} + b \exp\{x\} + \cos \pi x$

donde

$$a = -(1. + \exp(1.)) / \Theta$$

$$b = (1. + \exp(-1.)) / \Theta$$

$$\Theta = \exp(1.) - \exp(-1.)$$

6.3. LA SIGUIENTE TABLA NOS MUESTRA LOS VALORES DE LOS PARAMETROS UTILIZADOS EN EL TRABAJO.

TABLA 1. VALORES DE LOS PARAMETROS

	I		II
	$\alpha$	$\bar{x}$	$a$
1	1.	.5	1.
2	5.	.2	20.
3	100.	.36388	30.

P.ej. I-2 significa problema I con parámetro 2,  $\alpha = 5$ ,  $\bar{x} = .2$  cada método fue corrido sobre el conjunto de estos ejercicios, con una serie de valores de NINT,  $k$  y NCOND.

Los errores inferidos dependen del método (ver, sección 6.4.1). Para facilitar el uso de estos métodos, los programas fueron llamados como sigue:

1. COLFEM : COLOCACION
2. GOLFEM : GALERKIN
3. COLGAL : COLOCACION-GALERKIN

6.4. ORDEN DE CONVERGENCIA, LO PRIMERO DE INTERES FUE VER SI - EL ORDEN DE CONVERGENCIA SE CONFIRMABA EN CADA METODO. ES TO SOLAMENTE SE MENCIONARA CON REFERENCIAS PROPIAS (VER.- p. ej. [2] Y TABLA 2)

Para poder realizar esto, una serie de cálculos con incremento en NINT fueron realizados, y las aproximaciones para el orden de cada método fueron calculadas de la siguiente forma:

Sean  $E(N_1)$  y  $E(N_2)$  los errores correspondientes a dos valores consecutivos de NINT,  $N_1$  y  $N_2$  con  $N_1 < N_2$  entonces el orden de convergencia, está dado por

$$\log | E(N_1) / E(N_2) | / \log (N_2/N_1)$$

La siguiente tabla nos muestra los órdenes de convergencia para los métodos.

TABLA 2. ORDEN DE CONVERGENCIA  
PARA LOS METODOS

COLOCACION	$L^2$ GALERKIN	COLOCACION $L^2$ - GALERKIN
4	3	3
4	3	3
4	3	3

6.4.1. En Esta Subsección Definiremos los Errores EI, EL1, EL2

Nuestra misión es calcular aproximaciones de las soluciones a nuestro-problema de valor a la frontera. Por lo que - debemos de estar concientes de la aproximación y de cómo el - - error en las soluciones afecta conforme el número de elementos - en la malla crece.

El error en la aproximación de elemento finito es la función  $e$ , definida como la diferencia en valor absoluto entre las soluciones exacta y aproximada.

$$e(x) = | u(x) - u_h(x) |$$

Es claro que el error real nunca puede ser calculado, - a no ser que la solución exacta sea conocida. Sin embargo, adn cuando  $u$  sea conocida es posible construir una estimación del -- error y determinar si el error decrece conforme  $h$  decrece y el número de elementos comienza a crecer. Por lo que, la información de este tipo es de gran utilidad en cálculos de elementos-finitos, ya que sirve como regla de aceptabilidad de elección - para varios elementos y para estimar la precisión en la aproximación si el número de elementos son duplicados o triplicados.

Más aún, tales estimaciones del error podrlan también- indicarnos qué elementos son aceptables para usarse en nuestro-problema.

Puesto que los errores son funciones, debemos tener un criterio de medida para el tamaño de estos, ya que estamos analizando la precisión de nuestras soluciones. Un criterio natural y universal usado para lo anterior es un número no negativo llamado la norma de la función y escrita como  $\|F\|$ .



Por consiguiente, deseamos calcular o estimar el error con respecto a alguna norma apropiada.

Existen tres opciones principales para elegir el error que son comunmente usadas en los métodos de elementos finitos:

1. Error norma potencia  $\|e\|_E$ ,
2. Error norma raíz cuadrada  $\|e\|_0$ ,
3. Error norma máxima e infinita  $\|e\|_\infty$ .

El error norma potencia depende de la forma del problema de valor a la frontera bajo consideración. Para nuestro problema esta definido

$$\|e\|_E = \left[ \int_{\Omega} ((e')^2 + e^2) dx \right]^{\frac{1}{2}}$$

Este error es realmente una de las formas más naturales y significantes para medir el error en nuestra aproximación. Si demandamos que nuestra aproximación sea tal que el error se comporte bien con respecto a  $\|e\|_E$  conforme la malla se refiere, entonces podemos confiar en el método de aproximación.

El error norma raíz cuadrada ( $L^2$  o L-dos) mide la raíz cuadrada de una función sobre su dominio y está definido como,

$$\|e\|_0 = \left[ \int_{\Omega} e^2 dx \right]^{\frac{1}{2}}$$

mientras que la norma máxima e infinita mide el máximo valor absoluto de una función sobre su dominio

$$\|e\|_\infty = \max_{0 \leq x \leq 1} |e(x)|$$

Cuando se llega a un sistema algebraico de ecuaciones en el proceso de aproximar una función cuya segunda derivada es diferente de cero en los puntos de la frontera ( $x = p$ ,  $x = q$ ) - se llega a un buen ajuste cerca del centro del intervalo (error de orden  $h^4$  si  $\nabla_{h_i}$  se tiene que  $h_i = h$ ), pero fuera de estas partes el error es comunmente grande.

Formalizando diremos que puede uno obtener un orden de magnitud de  $O(h^4)$ .

Ver tabla 2(a), 2(b), 2(c) para los órdenes de convergencia para los métodos.

T A B L A 2(a)

METODO	PROBLEMA NUMERICO	$N_1, N_2$	EI	EL 1	EL 2
COLOCACION	II - 1	6,7	4.56	5.33	5.11
		7,8	4.49	4.52	4.87
		8,9	4.49	4.76	4.71
COLOCACION	III	6,7	4.89	2.59	4.29
		7,8	5.07	3.85	4.13
		8,9	4.83	2.91	4.04

T A B L A 2(b)

METODO	PROBLEMA NUMERICO	$N_1, N_2$	EI	EL 1	EL 2
GALERKIN	I - 2	6,7	3.93	3.92	3.89
		7,8	3.75	3.71	3.70
		8,9	3.44	3.58	3.58
GALERKIN	II - 1	6,7	4.06	4.43	4.04
		7,8	3.11	3.47	3.79
		8,9	3.65	3.88	3.65
GALERKIN	II - 3	14,15	2.28	2.86	2.80
		15,16	2.43	3.40	2.85
		16,17	2.58	2.77	2.92

T A B L A 2(c)

METODO	PROBLEMA NUMERICO	$N_1, N_2$	EI	EL 1	EL 2
COLGAL	II - 2	14,15	2.56	3.18	2.88
		15,16	2.70	2.74	2.94
		16,17	2.85	3.07	2.89
COLGAL	II - 3	14,15	2.27	3.18	2.66
		15,16	2.42	2.62	2.75
		16,17	2.58	2.95	2.73

TABLA 2(a) Ordenes de convergencia para el método de colocación

TABLA 2(b) Ordenes de convergencia para el método de Galerkin

TABLA 2(c) Ordenes de convergencia para el método de colocación Galerkin.

6.5 PRECISION DE LOS METODOS. DE LA TABLA 3(A) A LA TABLA 3(C) DAMOS ALGUNOS RESULTADOS TÍPICOS OBTENIDOS DE LOS TRES CODIGOS USADOS EN ESTE TRABAJO. CADA TABLA SE REFIERE A UN PROBLEMA Y A UN VALOR DE NINT.

T A B L A 3(a)

METODO	PROBLEMA NUMERICO I-2 NINT. = 10		
	E I	E L 1	E L 2
C O L F E M	5.42E-4	0.02	0.0022
G A L F E M	0.0023	0.04	0.01
C O L G A L O	0.02	0.771	0.1

T A B L A 3(b)

METODO	PROBLEMA NUMERICO I-2 NINT= 24		
	E I	E L 1	E L 2
C O L F E M	1.58 E-5	3.32E-4	4.81E-5
G A L F E M	1.33E-4	0.002	3.49E-4
C O L G A L	0.003	0.121	0.0151

T A B L A 3(c)

METODO	PROBLEMA NUMERICO I-2 NINT = 34		
	E I	E L 1	E L 2
C O L F E M	3.49E-6	7.95E-5	1.13E-5
G A L F E M	4.03E-5	7.01E-4	1.14E-4
C O L G A L	0.0013	0.06	0.0073

**CUARTA PARTE**  
**CONCLUSIONES**

## CONCLUSIONES

Es indudable que un aspecto indispensable, para tener una buena base en la solución de problemas de valor a la frontera, es el conocimiento teórico de los conceptos, principios y leyes de la física.

Sin embargo, la práctica que tenga el ingeniero, físico o matemático en la aplicación de dichos conceptos, principios y leyes, será fundamental para determinar el camino o método que se debe emplear en la solución de un problema específico, así como en la obtención del resultado y en la confianza de éstos en la veracidad del mismo.

Por ello, al principio de este trabajo, comenzamos a definir las áreas en las cuales, los métodos de elementos finitos han sido de gran utilidad. Posteriormente definimos el problema de nuestro interés y procedimos a dar una breve idea de los métodos utilizados en el mismo. Dada la amplia variedad de métodos numéricos existentes cabe hacer mención de lo siguiente:

El método de elementos finitos es un método para el tratamiento aproximado de problemas físicos, definidos,  $pej$ , por una ecuación diferencial con c. f. o por principios variacionales. Comparado con algunos métodos,  $p. ej.$ , con el método variacional de Ritz, ha sido muy útil cuando el dominio del problema tiene una forma complicada o cuando las funciones involucradas tienen un comportamiento muy diferente en diferentes partes de  $\Omega$ .

El método de elementos finitos y el método Ritz son equivalentes (ver  $p. ej.$  (19)). Cada método usa un conjunto de

funciones llamadas de prueba como el punto de partida para obtener una solución aproximada; ambos métodos toman combinaciones lineales de estas funciones de prueba y ambos métodos buscan la combinación de dichas funciones para establecer una funcional.

La mejor diferencia entre estos dos métodos es que el supuesto de estas funciones en el método de elementos finitos no es definido sobre todo el dominio  $\Omega$  y tiene que satisfacer ciertas condiciones de continuidad. El método Ritz usa funciones de prueba sobre todo el dominio  $\Omega$  de solución, en donde  $\Omega$  tiene solamente una forma geométrica simple.

En el método de elementos finitos las mismas limitaciones geométricas existen, pero solamente para los elementos  $\Omega_i$ .

En muchos casos, el método de elementos finitos puede ser visto como una aplicación del método Ritz o del método Galerkin.

El método Galerkin es el método más popular en la actualidad, aunque no es más que un caso particular del método de proyección. En el que  $H_0^{(N)}$  es un subespacio generado por  $\phi_i(x)$  con  $i = 1, N$  las funciones en término de las cuales se busca la solución.

Por otra parte, el método de elementos finitos de tipo COLOCACION es inmediatamente aplicable a sistemas de ecuaciones diferenciales mientras que los otros m.e.f. no tienen una interpretación simple para más de una ecuación. Esto pone en ventaja al método de colocación antes de hacer cualquier comparación.

Tenemos como conclusiones, después de haber efectuado los cálculos, las siguientes:



1. El método de colocación es el mejor y el más fácil de aplicar.

2. Hablando de formas débiles en general, podemos decir que el método de colocación en su versión m. e. f. corresponde a la más fuerte de las formas débiles.

La precisión de cada método depende del orden de la aproximación y el número de subintervalos.

A través de este trabajo el método de colocación usando funciones cúbicas de Hermite se compara con el método Galerkin que usa funciones cuadráticas de LAGRANGE.

Los resultados corroboran la teoría para cada uno de los métodos y se confirma aproximadamente de esta manera el orden de los mismos. A manera de comparación a lo referente a estos órdenes de convergencia y de la precisión de los métodos, mencionaremos que fue necesario hacer algunas pruebas incrementando la malla para confirmar efectivamente que se cumpliera el orden teórico.

Los cuadros 3(a) - 3(b) presentan una mejor comparación entre los métodos con respecto a la precisión producida por cada uno de ellos.

## BIBLIOGRAFIA

## BIBLIOGRAFIA

1. Gilbert Strang & George F. Fix, *An Analysis of the Finite Element Method*, Prentice - Hall series in automatic computation. 1973 page 4.
2. Meade, D., *Collocation Methods for Space Time Nuclear - - Reactor Dynamics*, Ph.D. Dissertation, University of California, Berkeley (1982); páginas 81-82.
3. P. Keast, G. Fairweather & J. C. Díaz, *A computational - study of Finite Element Methods for second order linear--two - point boundary value problems*. *Mathematics of computation*, V. 40 number 162. April 1983, pp. 501-505
4. Slater, J.C., "Electronic Energy Bands in Metals", *Phys. - Rev.*, 45 (1934) 794.
5. Bartha, J., "Über die Näherungsweise Lösung einiger Zweidimensionaler Elastizitäts-Aufgaben", *Z. Angew. Math. -- Mech.*, 17 (1937) 184.
6. Frazer, R. A., Jones, W. P., & Skan, S. W., "Approxima--tions to Function and to the Solutions of Diferential -- Equations", Great Britain Air Ministry Aero, Res. Comm. - tech. Rep. 1 (1937) 517.
7. Lanczos, C., "Trigonometric Interpolation of Empirical - and Analytical Functions", *J. Math. Phys.*, 17 (1938) 123.
8. Dixon, A. G., Finlayson, B. A., "Solution of Packed-Bed - Heat - Exchanger Models by Orthogonal Collocation using -

- Piecewise Cubic Hermite Functions, MRC TSR #2116. Madison, Wisconsin, 1980.
9. Meade, D., "Solution of the Neutron Group - Diffusion Equations by Orthogonal Collocation with cubic Hermite Interpolants", *transport theory and Statistical Physics*, - 12(3), 271-284. (1983).
  10. Russell, R.D., Shampine, L. F., "A collocation Method for Boundary Value Problems", *Number. Math.*, 19, 1 (1972).
  11. Finlayson, B. A., *The Method of Weighted Residuals and Variational Principles*, Academic Press, New York, 1972.
  12. Deboor, C., Swartz, B., "Collocation at Gaussian Points", *SIAM J. Numer. Anal.*, 10, 57 (1973).
  13. Deboor, C., *A practical Guide to Splines* Springer - Verlag. New York, 1978.
  14. Cenetti, J. H., *Collocation for Systems of ordinary Differential Equations*. Comp. Sci. tech. Rep. 230, Univ. Wisconsin - Madison, 1974.
  15. Douglas, J., Dupont, t., *Collocation Methods for Parabolic Equations in a single space variable*, Springer-Verlag, Berlin (1974).
  16. Graham F. Carey, J. Tinsley oden, *Finite Elements. a Second course*. Vol. II. pages 34-41, 168 Vol. I pages 16-17.
  17. Hennart B. Jean Pierre, *Recent Methods for Numerical Solution of Partial Differential Equations*, ITMAS-UNAM.

18. U. Ascher, J. Christiansen & R.D. Russell, "A collocation solver for mixed order systems of boundary value problems", *Math. Comp.*, V. 33, 1979, pp. 659-679.
19. R.D. Russell, "A comparison of collocation and finite differences for two - point boundary value problems", *SIAMJ, Numer. Anal.*, V 14, 1979. p.p. 19-39.

ALEZ (04/03/87)

12:02 PM FRIDAY, APRIL 3, 1987.

```
1000 FILE 1=MST/AUX1,UNIT=DISK, RECORD=1020, AREA=1*200
2000 FILE 2=MST/AUX2,UNIT=DISK,RECORD=1020, AREA= 1*200
3000 FILE 5=MST/DATA, UNIT=DISK, RECORD=14, BLOCKING=420
4000 FILE 6=IMP, UNIT=PRINTER, RECORD=22
5000 FILE 7=MSG,UNIT=REMOTE
6100 METODOS NUMERICOS DE ELEMENTOS FINITOS PARA LA SO-
LUCION DE PROBLEMAS DE VALOR A LA FRONTERA
6200
6300 TESIS BASADA EN EL ARTICULO:
6400 "A COMPUTATIONAL STUDY OF FINITE ELEMENT METHODS -
FOR SECOND ORDER LINEAR TWO-POINT BOUNDARY VALUE PROBLEMS"
6500
6600 BY P.KEAST, G.FAIRWEATHER & J.C. DIAZ.
7300 METODO DE: COLOCACION
10000 DIMENSION A (50000), B(26), HED(12), DATA(2)
11000 COMMON NLBC, NRBC,
12000 * NOUT, EPSL, IMPX, IMPT, XG (2).
13000 * U(4),U0(4,2),U1(4,2),U2(4,2)
16000 C* NMA = NUMBER OF MATERIALS
17000 C* NGR = NUMBER OF GROUPS
19000 C* NPT = NUMBER OF MESH POINTS
19100 C* NPLOT = NUMBER TO USE PLOTTER
20000 C*
```

21000	C*		00021000
22000	C* STATIC CALCULATION		00022000
23000	C*****		00023000
23100		TIME=0.	00023100
23200		TSTAT=.	00023200
23400		TOUTP=C.	00023400
24700		LD=26	00024700
25000		IFIMAX=50000	00025000
26000	10	READ(5,1000) HLD	00026000
27000		READ(5,7) NMA,NGR,NPT,HPL0T	00027000
28000		IF(NMA.EQ.0) CALL EXIT	00028000
29000		CALL DATIT(DATA)	00029000
30000		WRITE(6,1100) HLD,DATA	00030000
31000	C*		00031000
32000		NLE=IABS(NPT)-1	00032000
33000		LG=NGR+NMA	00033000
34000		LEG=LG**NCR	00034000
35000	C*		00035000
35500		INPX=NPT	00035500
36000		ID=1	00036000
37000		ISR=ID+LG	00037000
38000		ISF=ISR+LG	00038000
39000		IXI=ISF+LG	00039000
40000		ISS=IXI+LG	00040000
41000		IX=ISS+LG	00041000
42000		INC=IX+IABS(NPT)	00042000
43000		IVT=INC+HLD-	00043000
44000		CALL CHCDEL("ENP61",IVT,1,IFIMAX)	00044000
45000	C*		00045000
46000	C*		00046000
47000			00047000
49000		CALL INFO7(1,1,0,0,0,NPT,NLE,A(10),A(1SR),A(1SF),A(1XI),A(1SS),	00049000
50000		A(1X),A(1C))	00050000

```

54300      T=HPD+T*HPD+DT(CI)
54400
54500      C+
54600      NFN=FM*NP1
54700      C+
54800      IFO=LVT
54900      IFN=IFO+NP1F*IGB
55000      IVT=IFO+NP1F*IGB
55100      CALL COUTL('STATIC',IV1,IFFMAX)
55200      LVT=IFN*YML*YCM+I
55300      IFF=IHL(CI)
55400      CALL STATDC('N',NDC,HPD,NLE,HPMF,A(10),A(15R),A(15F),A(15I),
55500      * A(15S),A(15D),I(100)+A(15O),A(15N),A(15T),LVT)
55600      TSTAT=STAT+DT(CI)
55700      IFF=IBBY
55800      IFF=IPLDT*(1+I)*IBBY
55900      IVT=IVT+HPD
56000      IVT1=IVT+HPD*IGB
56100      IVT2=IVT1+140
56200      T=T+IHL(CI)
56300      C*
56400      CALL COUTF('N',IGB,IBT,NLE,HPMF,HPLOT,1HP,A(15),
56500      * A(15D),A(15N),A(15O),A(15T),A(15T1),L)
56600      TOUTP=TOUTP+DT(CI)
56700      NHTICS=1300T*HPD,TSTAT,TOUTP
156800      12LU FORMAT(12A6)
156900      11LU FORMAT('HPD=12A6.9Y+DATE : ",2X,70(1H*)')
157000      13LU FORMAT('HTICS=OVERALL TIME LOG FOR STATIC CALCULATION//
157100      *41X,39(1H*)//
157200      *3X,"SUBROUTINE",IX,"TIME(CPU SECS)",///
157300      *26X,"INPSTACK INPUT OF STATIC DATA).....",1PE12.5//
157400      *20X,"STATIC SOURCE ITERATION FOR STEADY STATE CALC.)",1PE12.5//
157500      *20X,"OUTPT (PRINT THE CRITICAL NEUTRON FLUXES).....",1PE12.5//

```



```

169000      GO TO 14
169100      END
169200      (*****
169300      C*
169400      BLOCK DATA
169500      C*
169600      C*****
169700      COMMON /BLK/ BLOC,
169800      *      BLOC, BLOC, BLOC, BLOC,
169900      *      BLOC, BLOC, BLOC, BLOC, BLOC,
170000      C*
170100      DATA X0/
170200      *      -0.715717E+01, 3.7735127E+01, -0.17
170300      DATA U1/
170400      *      1.000000E+01, 3.333333E+01,
170500      *      1.000000E+01, -3.333333E+01,
170600      DATA U2/
170700      *      3.3490072E+01, -2.5299927E+01,
170800      *      1.1512281E+01, -7.1441621E-02,
170900      *      1.1512281E+01, 7.1441621E-02,
171000      *      2.8490072E+01, -2.5299927E+01,
171100      DATA U3/
171200      *      -1.000000E+01, +2.8667518E+01,
171300      *      5.000000E+01, -2.8667518E+01,
171400      *      -5.000000E+01, -2.8667518E+01,
171500      *      5.000000E+01, +2.8667518E+01,
171600      DATA U4/
171700      *      -8.66625403734E-01, -1.36662540373E+01,
171800      *      8.66625403734E-01, -3.66625403734E-01,
171900      *      8.66625403734E-01, 3.66625403734E-01,
172000      *      -8.66625403734E-01, -1.36662540373E+01,
181000      C*
182000      END
00150000
00151000
00152000
00153000
00154000
00155000
00156000
00157000
00158000
00159000
00160000
00161000
00162000
00163000
00164000
00165000
00166000
00167000
00168000
00169000
00170000
00171000
00172000
00173000
00174000
00175000
00176000
00177000
00178000
00179000
00180000
00181000
00182000

```

183000	C*****		00183000
184000	C*		00184000
185000		SUBROUTINE BVSOLN(B,C,N,M,V)	00185000
186000	C*		00186000
187000	C*****		00187000
188000		DIENSION C(N,1),C(N,M),V(N)	00188000
189000	C*		00189000
190000		DO 5 J=1,M	00190000
191000		DO 5 I=1,N	00191000
192000	5	C(I,J)=3(C,I,J)	00192000
193000	C*		00193000
194000		LR=(N+1)/2	00194000
195000		LRH1=LR-1	00195000
196000		DO 15 IR=1,LR-1	00196000
197000		LRHR=LR-IR	00197000
198000		DO 15 I=1,LRHR	00198000
199000		DO 10 J=2,M	00199000
200000		C(IR,J-1)=C(IR,J)	00200000
201000	10	CONTINUE	00201000
202000		C(N+1-IR,I+1-1)=C(I	00202000
203000	15	CONTINUE	00203000
204000	C*		00204000
205000			00205000
206000		NH1=N-1	00206000
207000		DO 20 I=1,NH1	00207000
208000		IP1V=1	00208000
209000		IP1=1+1	00209000
210000		DO 20 IR=IP1,LR	00210000
211000		F=ABS(C(-IR,1))-IT*/30(C(IP1V,1)) ; IP1V=IR	00211000
212000	20	CONTINUE	00212000
213000		IF(C(IP1V,1) > 1) GOTO 40	00213000
214000		T=V(I)	00214000
215000		V(I)=V(IP1V)	00215000
216000		V(IP1V)=T	00216000

217000  
218000  
219000  
220000  
221000  
222000  
223000  
224000  
225000  
226000  
227000  
228000  
229000  
230000  
231000  
232000  
233000  
234000  
235000  
236000  
237000  
238000  
239000  
240000  
241000  
242000  
243000  
244000  
245000  
246000  
247000  
248000  
249000

```
DO 70 J=1,M
  T=C(1,J)
  C(X,J)=C(PIV,J)
  C(PIV,J)=T
30 CONTINUE
40 V(I)=V(I)/C(1,1)
  DO 50 J=2,M
    C(I,J)=C(I,J)/C(1,1)
50 CONTINUE
  DO 70 IR=1,LR
    T=C(IR,1)
    V(IR)=V(IR)-T*V(1)
    DO 60 J=2,M
      C(IR,J-T)=C(IR,J)-T*C(1,J)
60 CONTINUE
    C(IR,M)=1.
70 CONTINUE
  IF(LR.NE.N) LR=LR+1
80 CONTINUE
C*
  V(N)=V(N)/C(N,1)
  JM=2
  DO 100 IRR=1,NM1
    IR=N-IRR
    DO 90 J=2,JM
      V(IR)=V(IR)-C(IR,J)*V(1+J)
90 CONTINUE
    IF(JM.NE.N) JM=JM+1
100 CONTINUE
C*
  RETURN
  END
```

00217000  
00218000  
00219000  
00220000  
00221000  
00222000  
00223000  
00224000  
00225000  
00226000  
00227000  
00228000  
00229000  
00230000  
00231000  
00232000  
00233000  
00234000  
00235000  
00236000  
00237000  
00238000  
00239000  
00240000  
00241000  
00242000  
00243000  
00244000  
00245000  
00246000  
00247000  
00248000  
00249000

\*\*\*\*\*

248020	C*		00248020
248030		SUBROUTINE BDFAC(N,M,L,A)	00248030
248040	C*		00248040
248050	C*	*****	00248050
248051		DIMENSION A(N,L)	00248051
248052	C*		00248052
248053	C*	CALCULATION OF THE ELEMENTS OF MATRIX L	00248053
248054	C*		00248054
248055		IF(M.EQ.1)RETURN	00248055
248056		DO 60 I=1,N	00248056
248057		IF(I.LE.1)GO TO 30	00248057
248058		JL=1	00248058
248059		IF(I.GT.M)JL=I*M+1	00248059
248060		JR=I-1	00248060
248061		DO 20 J=JL,JR	00248061
248062		A(I,M+J-1)=A(I,M+J-1)/A(J,M)	00248062
248063		IF(I.LE.2)GO TO 20	00248063
248064		IF(J.LE.JL)GO TO 20	00248064
248065		KL=JL	00248065
248066		KR=J-1	00248066
248067		DO 10 K=KL,KR	00248067
248068		A(I,M+J-1)=A(I,M+J-1)-A(I,M+K-1)*A(K,M+J-K)/A(K,M)	00248068
248069		10 CONTINUE	00248069
248070		20 CONTINUE	00248070
248071	C*		00248071
248072	C*	CALCULATION OF THE ELEMENTS OF MATRIX U	00248072
248073	C*		00248073
248074		50 JR=N	00248074
248075		IF(I.LE.N-M+1)JR=I+M-2	00248075
248076		JL=I	00248076
248077		DO 51 J=JL,JR	00248077
248078		IF(I.LE.1)GO TO 51	00248078
248079		KI=I-M+1	00248079

248080	KJ=J-M+1	00248080
248081	KL=KJ	00248081
248082	IF(KJ.GT.M)KL=KJ	00248082
248083	IF(KL.LT.1)KL=1	00248083
248084	KI=I-1	00248084
248085	DO 41 K=KL,KR	00248085
248086	A(I,M+J-I)=A(I,M+J-1)-A(I,M+K-I)*A(K,M+J-K)	00248086
248087	41 CONTINUE	00248087
248088	51 CONTINUE	00248088
248089	60 CONTINUE	00248089
248090	RETURN	00248090
248091	END	00248091
248092	C*****	00248092
248093	C*	00248093
248094	SUBROUTINE QDSOL(N,M,L,A,X)	00248094
248095	C*	00248095
248096	C*****	00248096
248097	DIMENSION A(N,L),X(N)	00248097
248098	C*	00248098
248099	FORWARD SUBEP	00248099
248100	C*	00248100
248110	IF(N.CO.1)GO TO 3L	00248110
248120	DO 20 I=1,N	00248120
248130	IF(I.LE.1)GO TO 2C	00248130
248140	JL=1	00248140
248150	IF(I.GT.M)JL=I-M+1	00248150
248160	JR=I-1	00248160
248170	DO 11 J=JL,JR	00248170
248180	X(I)=X(I)-A(I,M+J-I)*X(J)	00248180
248190	21 CONTINUE	00248190
248200	20 CONTINUE	00248200
248210	C*	00248210
248220	C*    BACKWARD SUBSTITUTION	00248220

248230  
248240  
248250  
248260  
248270  
248280  
248290  
248300  
248310  
248320  
248330  
248340  
248350  
248360  
248370  
248380  
249000  
250000  
251000  
252000  
253000  
254000  
255000  
256000  
257000  
257500  
257550  
257600  
258000  
259000  
260000  
261000  
261200

```
C*
      TO DC 5( K=1,N
          I=N-K+1
          X(I)=X(I)/A(I,M)
          IF(M.EQ.1)GO TO 60
          IF(K.LE.1)GO TO 50
          JR=M
          IF(K.GT.M)JR=I+M-1
          JL=I+1
          DO 4( J=JL,JR
              X(I)=X(I)-A(I,M+J-I)*X(J)/A(I,M)
          40 CONTINUE
          50 CONTINUE
          60 CONTINUE
          RETURN
          END
```

C\*\*\*\*\*  
C\*

```
      SUBROUTINE CALDIF(A,NPT,NELE,NPAF,X)
```

C\*\*\*\*\*

```
      DIMENSION A(NPAF,S),X(NPT)
      COMMON NIBC,NR3C,
      *      NOUT,EPSL,IMPX,IMPT,XG(2),
      *      U(4),U(4,2),U1(4,2),U2(4,2)
      EN1(IF,IG,DC,BC,SC,DX)=(-DC*U2(IF,IG)*(2./DX)**2
      * +DC*U1(IF,I3))*(2./DX)
      * +DC*U1(IF,IG))*(.5*DX)*MOD(IF+1,2)
```

```
C*
      DO 11 M=1,S
      DO 13 H=1,NPAF
      11 A(N,M)=L.
      13 A(1,NIBC+S)=1.
```

00248230  
00248240  
00248250  
00248260  
00248270  
00248280  
00248290  
00248300  
00248310  
00248320  
00248330  
00248340  
00248350  
00248360  
00248370  
00248380  
00249000  
00250000  
00251000  
00252000  
00253000  
00254000  
00255000  
00256000  
00257000  
00257500  
00257550  
00257600  
00258000  
00259000  
00260000  
00261000  
00261200

261400  
262000  
262500  
263000  
263500  
264000  
264500  
265000  
265500  
266000  
266500  
267000  
267500  
268000  
268500  
269000  
269500  
270000  
270500  
271000  
271500  
272000  
272500  
273000  
273500  
274000  
274500  
275000  
275500  
276000  
276500  
277000  
277500  
278000  
278500  
279000  
279500  
280000  
280500  
281000  
281500  
282000  
282500  
283000  
283500  
284000  
284500  
285000  
285500  
286000  
286500  
287000  
287500  
288000  
288500  
289000  
289500  
290000  
290500  
291000  
291500  
292000  
292500  
293000  
293500  
294000  
294500  
295000  
295500  
296000  
296500  
297000  
297500  
298000  
298500  
299000  
299500  
300000

C\* AC(IPAF, NR 30-L) = 1.

M=1  
DO 4L NR=1, NFILE  
N=X(NR)  
XR=X(NR+1)  
XR=X(2\*N)  
DO 3L J=1, 3  
XC=(X+Y\*(J)+N+XC)\*L  
DC=F(XL)  
BC=FB(XC)  
LC=F(XC)  
N=N+1  
DO 2L L=1, 4  
L=L+1

10 CONTINUE  
20 CONTINUE  
30 CONTINUE

DO 5L I=1, NPT  
FX=UBX(XC...)  
10 CONTINUE  
RETURN  
END

C\*\*\*\*\*

SUBROUTINE CALSOU (B, NGR, NPT, NFILE, NPAF,  
\* X, NC, SF, XL, SS, FO, FN, NG, XR)

C-----

C\* SOURCE CALCULATION FOR GROUP NG  
C\*

00261400  
00262000  
00262500  
00263000  
00264000  
00265000  
00266000  
00268010  
00268120  
00268030  
00268040  
00268050  
00269000  
00275400  
00275500  
00275800  
00276700  
00279000  
00280000  
00280010  
00280020  
00280040  
00282000  
00283000  
00345000  
00346000  
00347000  
00348000  
00349000  
00350000  
00351000  
00352000  
00353000

```

354000 DIMENSION B(NPAF),XC(NPT),Nc(NELL)
355000 * OF(NGR,1),XI(IGR,1),SS(NGR,NGR,1),FO(NPAF,1),FN(NPAF,1),XRC(1)
356000 COMMON NLBC,NRRC
357000 *
358000 * HOUT,EPSSL,IMPX,IMPT,XG(2),
359000 * U(4),UU(4,2),U1(4,2),UR(4,2)
360000 C*
361000 C* INITIALIZATION
362000 C*
363000 DO 1L I=1,NPAF
364000 Y(I)=H(1)
365000 C* VECTOR CALCULATION
366000 C*
367000 DO 4L I=1,NELI,5
368000 X1=X(I)
369000 X2=X(I+1)
370000 DX=X2-X1
371000 H=I*2
372000 XGL=(DX*XG(1)+X1+X2)*0.5
373000 B(N)=B(N)+FF(XGL)
374000 H=I*2+1
375000 XGR=(DX*XG(2)+X1+X2)*0.5
376000 U(N)=B(N)+FF(XGR)
377000 4L CONTINUE
378000 C*
379000 RETURN
380000 END
381000 C*****
382000 C*
383000 SUBROUTINE CICTRL(SUBRT4,IVT,LVTH,X)
384000 C*
385000 C*****
386000 LVTH=LVTH-X-IVT+1

```

```

00354000
00355000
00356000
00357000
00358000
00359000
00360000
00361000
00362000
00363000
00364000
00365000
00366000
00367000
00368000
00369000
00370000
00371000
00372000
00373000
00374000
00375000
00376000
00377000
00378000
00379000
00380000
00381000
00382000
00383000
00384000
00385000
00386000
00387000
00388000
00389000
00390000
00391000
00392000
00393000
00394000
00395000
00396000
00397000
00398000
00399000
00400000

```



492000  
494000  
495000  
496000  
497000  
498000  
499000  
500000  
501000  
502000  
503000  
504000  
505000  
506000  
507000  
508000  
509000  
510000  
511000  
512000  
513000  
514000  
515000  
515010  
515020  
515030  
515040  
515050  
515060  
515070  
515080  
799400  
799600

IF (LVT.GT.C) GO TO 10  
WRITE(6,100) SUBRTH,LVT  
CALL EXIT

C\*  
CALL FCNMT('ITOC',MUCH MEMORY NEEDED - ",16," - LVT=",110)  
IF RETURN  
END

C\*\*\*\*\*  
C\*

SUBROUTINE PATY (DATE)

C\*  
C\*\*\*\*\*  
REAL DATE(4)

C\*  
DATE(1)=' / /'  
DATE(2)=''  
I=TIME(15)  
DATE(1)=CONCAT (DATE(1),B,47,47,16)  
DATE(1)=CONCAT (DATE(1),B,23,21,16)  
DATE(2)=CONCAT (DATE(2),B,47,15,16)

C\*  
RETURN  
END

C\*\*\*\*\*  
C\*

FUNCTION DF(T)

C\*\*\*\*\*  
DT=(TIME(15)-T)\*2.64-6

RETURN  
END

C\*\*\*\*\*  
C\*

00493000  
00494000  
00495000  
00496000  
00497000  
00498000  
00499000  
00500000  
00501000  
00502000  
00503000  
00504000  
00505000  
00506000  
00507000  
00508000  
00509000  
00510000  
00511000  
00512000  
00513000  
00514000  
00515000  
00515010  
00515020  
00515030  
00515040  
00515050  
00515060  
00515070  
00515080  
00799400  
00799600

799800	SUBROUTINE GRAFOX(K,11,XMIN,XMAX,F)	00799800
799900	C*	00799900
800000	C*****	00800000
800100	DIMENSION LETV(51),LETRA(3),X(11),IA(101,51),F(N)	00800100
800200	C*	00800200
800300	DATA LETV/5* , ,N*,2* , ,E*,2* , ,U*,2* , ,T*,2* , ,R*,2* , ,	00800300
800400	* ,Q*,2* , ,N*,5* , ,F*,2* , ,L*,2* , ,J*,2* , ,X*,9* , ,/	00800400
800500	DATA LETRA/1H ,1H+,1H+//	00800500
800600	WRITE(6,1000)K	00800600
800700	CALL MINMAX(N,E,FMIN,FMAX)	00800700
800800	DO 10 I=1,101	00800800
800900	DO 10 J=1,51	00800900
801000	IA(I,J)=LETRA(1)	00801000
801100	10 CONTINUE	00801100
801200	DO 20 I=1,101	00801200
801300	IA(I,1)=LETRA(3)	00801300
801400	IA(I,51)=LETRA(2)	00801400
801500	20 CONTINUE	00801500
801600	DO 30 J=1,51	00801600
801700	IA(1,J)=LETRA(2)	00801700
801800	IA(101,J)=LETRA(2)	00801800
801900	30 CONTINUE	00801900
802000	Q=50./(FMAX-FMIN)	00802000
802100	DO 40 I=1,101	00802100
802200	J=Q*(F(I)-FMIN)+1+.5	00802200
802300	IA(I,J)=LETRA(3)	00802300
802400	40 CONTINUE	00802400
802500	J=52	00802500
802600	DO 50 L=1,51	00802600
802700	J=J-1	00802700
802800	IF (MOD(L-1,5).EQ.0) FLUX=FMAX-(L-1)/9	00802800
802900	IF (MOD(L-1,5).EQ.0) WRITE(6,1200) LETV(L),FLUX,(IA(I,J),I=1,101)	00802900
803000	IF (MOD(L-1,5).NE.0) WRITE(6,1200) LETV(L),(IA(I,J),I=1,101)	00803000

81310  
 81320  
 81330  
 81340  
 81350  
 81360  
 81370  
 81380  
 81390  
 81400  
 81410  
 81420  
 81430  
 81440  
 81450  
 81460  
 81470  
 81480  
 81490  
 81500  
 81510  
 81520  
 81530  
 81540  
 81550  
 81560  
 81570  
 81580  
 81590  
 81600  
 81610  
 81620  
 81630  
 81640  
 81650  
 81660  
 81670  
 81680  
 81690  
 81700  
 81710  
 81720  
 81730  
 81740  
 81750  
 81760  
 81770  
 81780  
 81790  
 81800  
 81810  
 81820  
 81830  
 81840  
 81850  
 81860  
 81870  
 81880  
 81890  
 81900  
 81910  
 81920  
 81930  
 81940  
 81950  
 81960  
 81970  
 81980  
 81990  
 82000  
 82010  
 82020  
 82030  
 82040  
 82050  
 82060  
 82070  
 82080  
 82090  
 82100  
 82110  
 82120  
 82130  
 82140  
 82150  
 82160  
 82170  
 82180  
 82190  
 82200  
 82210  
 82220  
 82230  
 82240  
 82250  
 82260  
 82270  
 82280  
 82290  
 82300  
 82310  
 82320  
 82330  
 82340  
 82350  
 82360  
 82370  
 82380  
 82390  
 82400  
 82410  
 82420  
 82430  
 82440  
 82450  
 82460  
 82470  
 82480  
 82490  
 82500  
 82510  
 82520  
 82530  
 82540  
 82550  
 82560  
 82570  
 82580  
 82590  
 82600  
 82610  
 82620  
 82630  
 82640  
 82650  
 82660  
 82670  
 82680  
 82690  
 82700  
 82710  
 82720  
 82730  
 82740  
 82750  
 82760  
 82770  
 82780  
 82790  
 82800  
 82810  
 82820  
 82830  
 82840  
 82850  
 82860  
 82870  
 82880  
 82890  
 82900  
 82910  
 82920  
 82930  
 82940  
 82950  
 82960  
 82970  
 82980  
 82990  
 83000

```

      50 CONTINUE
      GO ON I=1,11
      W=(XDAY-XMIN)/T
      XC=X*(I-1)+XMIN
    60 CONTINUE
      U=TL(S,130)C(X),J=1,11
      100 FORMAT(20H PLOT OF NEUTRON FLUX FOR GROUP ,I2,4X,
      * 10H VERSUS POSITION //,54(1H+))
      110 FORMAT(3X,A1,4),IPE12.5,10TA1
      120 FORMAT(2X,A1,10X,11A1)
      130 FORMAT(16X,11(1F3.3-2X),//,54Y,18HP 0 3 1 1 0 0 X)
      RETURN
      END
    C+*****
    C+
      SUBROUTINE IMPSTA(NMA,NGR,NPT,NLELE,B,SR,SF,XI,SS,
      * X,NC)
    C+
    C+*****
      DIMENSION N(NGR,NMA),SR(NGR,NMA),SF(NGR,NMA)
      * XI(NGR,NMA),SS(NGR,NGR,NMA),X(NPT),NC(NLE)
      COMMON NLEB,NRSC,
      * NGR,EPRL,IMPX,IMPY,XC(2),
      * U(4),UU(4,2),U1(4,2),U2(4,2)
    C+
    C+ BOUNDARY CONDITIONS
    C+
      READ(5,/) NLEB,NRBC
    C+
    C+ GEOMETRIC DATA
    C+
      IF (NPT.GT.0) GO TO 10
      NPT=-NPT
  
```

81310  
 81320  
 81330  
 81340  
 81350  
 81360  
 81370  
 81380  
 81390  
 81400  
 81410  
 81420  
 81430  
 81440  
 81450  
 81460  
 81470  
 81480  
 81490  
 81500  
 81510  
 81520  
 81530  
 81540  
 81550  
 81560  
 81570  
 81580  
 81590  
 81600  
 81610  
 81620  
 81630  
 81640  
 81650  
 81660  
 81670  
 81680  
 81690  
 81700  
 81710  
 81720  
 81730  
 81740  
 81750  
 81760  
 81770  
 81780  
 81790  
 81800  
 81810  
 81820  
 81830  
 81840  
 81850  
 81860  
 81870  
 81880  
 81890  
 81900  
 81910  
 81920  
 81930  
 81940  
 81950  
 81960  
 81970  
 81980  
 81990  
 82000  
 82010  
 82020  
 82030  
 82040  
 82050  
 82060  
 82070  
 82080  
 82090  
 82100  
 82110  
 82120  
 82130  
 82140  
 82150  
 82160  
 82170  
 82180  
 82190  
 82200  
 82210  
 82220  
 82230  
 82240  
 82250  
 82260  
 82270  
 82280  
 82290  
 82300  
 82310  
 82320  
 82330  
 82340  
 82350  
 82360  
 82370  
 82380  
 82390  
 82400  
 82410  
 82420  
 82430  
 82440  
 82450  
 82460  
 82470  
 82480  
 82490  
 82500  
 82510  
 82520  
 82530  
 82540  
 82550  
 82560  
 82570  
 82580  
 82590  
 82600  
 82610  
 82620  
 82630  
 82640  
 82650  
 82660  
 82670  
 82680  
 82690  
 82700  
 82710  
 82720  
 82730  
 82740  
 82750  
 82760  
 82770  
 82780  
 82790  
 82800  
 82810  
 82820  
 82830  
 82840  
 82850  
 82860  
 82870  
 82880  
 82890  
 82900  
 82910  
 82920  
 82930  
 82940  
 82950  
 82960  
 82970  
 82980  
 82990  
 83000

1197003	FIMP=FOX(N,1)	01197003
1197005	WRITE(7,*)FEX,FIMP	01197005
1197100	ERN=ABS(FOX(N,1)-UBEX(XIMP(N)))	01197100
1197200	IF(ET.LT.ERN)E1=ERN	01197200
1197300	EL1=EL1+ERN	01197300
1197400	EL2=EL2+ERN*ERN	01197400
1199000	41  CONTINUE	01199000
1199100	EL3=SQRT(EL2)	01199100
1199200	WRITE(7,*)E1,EL1,EL2	01199200
1200000	C*	01200000
1200500	IF(NPLOT.EQ.0)RETURN	01200500
1200600	DO 60 K=1,NGR	01200600
1200700	CALL GRAFOX(K,IMP,XMIN,XMAX,FOX(1,Y))	01200700
1200800	60  CONTINUE	01200800
1203000	RETURN	01203000
1204000	END	01204000
1265000	C*****	01265000
1264000	C*	01264000
1265000	SUBROUTINE STATLOC(MM,NGR,NPT,NELE,NPAF,D,	01265000
1266000	*  SR,SF,X2,S1,X,NC,FU,FH,VT,LUTMAX)	01266000
1267000	C*	01267000
1268000	C*****	01268000
1269000	DIMENSION D(NGR,NMA),SR(NGR,NMA),SF(NGR,NMA),X1(NGR,NMA),	01269000
1270000	*  SS(NGR,NGR,NMA),X(NPT),NC(NELE),FU(NPAF,NGR),	01270000
1271000	*  FH(NPT,NGR),VT(1)	01271000
1272000	COMMON NLR,NRNG	01272000
1273000	*  NRU,LRSL,IMPY,THPT,XG(2),	01273000
1274000	*  UC(4),UC(4,2),L1(4,2),UT(4,2)	01274000
1275000	C*	01275000
1276000	C*  INITIALIZATIONS	01276000
1277000	C*	01277000
1277500	NPAF=NPAF*	01277500
1281000	LAF1	01281000

117640		CONTINUE X(NPT),FOXP(AE,H3D),FOX(IHP,NGR),XIMP(IHP),	1176400
117650		*D(NGR,H3D),NCC(ILE3)	1176500
117700	C+		1177000
117800		COMMON H3D2,IS3D,	1178000
117900		* NGRT,DRSL,IMPX,IMPT,XG(2),	1179000
118000		* U(4),U(4,2),U3(4,2),U7(1,2)	1179000
118100	L+		1180000
118120		ISHL=3	1181200
118140		LL1=1	1181200
118160		LL2=2	1181200
118180		EMAX=X(NPT)	1181200
118200		XMIN=X(1)	1181300
118220		IM=1	1181400
118240		DXIMP=(XMIN-XMAX)/FLOAT(IM-1)	1183000
118260		DO 4C N=1,IMP	1184000
118280		XIMP(N)=XMIN+DXIMP*FLOAT(N-1)	1185000
118300		DO 5 M=1,LL1	1186000
118320		KC=X(NI+1)	1187000
118340		IF(XC(LL1-XIMP(N))GT 10.5	1187000
118360		INT=N1	1187000
118380		GO TO 7	1187000
118400	2	CONTINUE	1187000
118420		INT=NELL	1187000
118440	7	CONTINUE	1187000
118460		NCC=NC(INT)	1187000
118480		DO 3C NG=1,NGR	1187000
118500		FOX(N,NG)=1	1187000
118520		DO 1C I=1,4	1187000
118540		II=(INT-1)*2+1	1187000
118560	1C	FOX(N,NG)=FOX(N,NG)+FO(II,NG)	1187000
118580	*	UIJ(I,X(1,II),X(1,II+1),XIMP(N))	1187000
118600	2C	CONTINUE	1187000
118620		FEX=UEX(XIMP(N))	1187000

1022000	READ (5,/) X	01022000
1023000	GO TO 30	01023000
1024000	10 READ (5,/) X(1),X(NPT)	01024000
1025000	DX=(X(NPT)-X(1))/FLOAT(MELE)	01025000
1026000	IF (MELE.LT.2) GO TO 30	01026000
1027000	DO 20 I=2,MELE	01027000
1028000	X(I)=X(1)+FLOAT(I-1)*DX	01028000
1029000	20 CONTINUE	01029000
1030000	30 CONTINUE	01030000
1096000	RETURN	01096000
1097000	END	01097000
1097120	C*****	01097120
1097140	C*	01097140
1097160	SUBROUTINE MINMAX(N,F,FMIN,FMAX)	01097160
1097180	C*	01097180
1097200	C*****	01097200
1097220	DIMENSION F(N)	01097220
1097240	FMIN=F(1)	01097240
1097260	FMAX=F(1)	01097260
1097280	DO 20 I=2,N	01097280
1097300	IF(FMIN.LT.F(I))GO TO 10	01097300
1097320	FMIN=F(I)	01097320
1097340	10 IF(FMAX.GT.F(I))GO TO 20	01097340
1097360	FMAX=F(I)	01097360
1097380	20 CONTINUE	01097380
1097400	RETURN	01097400
1097420	END	01097420
1097430	C*****	01097430
1171000	C*	01171000
1172000	SUBROUTINE OUTPUT(NMA,NGR,NPT,MELE,HPAF,NPLO1,IMP,X,D,NC,	01172000
1172500	*FO,FOX,XIMP,IT)	01172500
1173000	C*	01173000
1173000	C*****	01173000

1282400  
 1282500  
 1283000  
 1284000  
 1285000  
 1286000  
 1306000  
 1307000  
 1308000  
 1309000  
 1314000  
 1314000  
 1318100  
 1319000  
 1319100  
 1320000  
 1322000  
 1322300  
 1322400  
 1323000  
 1368000  
 1369000  
 1370000  
 1371000  
 1372000  
 1373000  
 1374000  
 1375000  
 1378000  
 1380000  
 1381000  
 1382000  
 1383000

```

IAF=IA+NPAF5*IGR
IB=JAF+NPAF5*IGR
IXR=IB+NPAF
IVT=IX2+HAXC(CD,4GR)*NELE
CALL CMCTRL("STATIC",IVT,LVTHAX)

C*
C*
DO 105 K=1,NGR
  IAK=IA+(K-1)*IPAF5
  IAFK=IAF+(K-1)*NPAF5
  CALL CALDIF(VT(IAK),NPT,NEL,NPAF,X)
  CALL CALSOJ(VT(IB),HGR,NPT,NLE,NPAF,X,NC,SF,X1,SS,FG,FN
    * ,K,VT(IXR))
  CALL DFAC(NPAF,3,5,VT(IAK))
  CALL BMSOL(NPAF,3,5,VT(IAK),VT(IB))
C*
  DO 102 I=1,NPAF
    FC(I,K)=VT(IB-I+1)
  102 FH(I,K)=VT(IB-I+1)
  105 CONTINUE
  RETURN
  END

C*****
C*
FUNCTION U1J (J,X1,X2,XC)
C*
C*
C*****
  H2=X2-X1
C*
  GO TO (2L,3C,3S,4I),J
C*
  2) U1J=3.+(X2-XC)/H2)+2-2.4((X2-XC)/H2)+3
  
```

01282000  
 01282500  
 01283000  
 01284000  
 01285000  
 01286000  
 01306000  
 01307000  
 01308000  
 01308500  
 01314000  
 01518000  
 01318100  
 01319000  
 01319100  
 01320000  
 01322200  
 01322300  
 01322400  
 01323000  
 01368000  
 01368000  
 01370000  
 01371000  
 01372000  
 01373000  
 01374000  
 01375000  
 01378000  
 01380000  
 01381000  
 01382000  
 01383000

1384000	RETURN	01384000
1385000	30: UIJ=((X2-XC)/H2)**2-((X2-XC)/H2)**3+H2	01385000
1386000	RETURN	01386000
1387000	35: UIJ=3+((XC-X1)/H2)**2-((XC-X1)/H2)**3	01387000
1388000	RETURN	01388000
1389000	40: UIJ=-((XC-X1)/H2)**2+((XC-X1)/H2)**3*H2	01389000
1390000	RETURN	01390000
1401000	END	01401000
1401100	C*****	01401100
1401200	C* PROBLEMA NUMERICO (1)	01401200
1401210	C*	01401210
1401300	FUNCTION FA(Z)	01401300
1401400	C*	01401400
1401500	C*****	01401500
1401550	FA=1.	01401550
1401560	RETURN	01401560
1401570	END	01401570
1402000	C*****	01402000
1402100	C*	01402100
1402200	FUNCTION FB(Z)	01402200
1402300	C*	01402300
1402400	C*****	01402400
1402410	FB=0.	01402410
1402420	RETURN	01402420
1402430	END	01402430
1402900	C*****	01402900
1403000	C*	01403000
1403100	FUNCTION FC(Z)	01403100
1403200	C*	01403200
1403300	C*****	01403300
1403310	FC=1.	01403310
1403320	RETURN	01403320
1403330	END	01403330



140570	[*****]	01403700
140571	C*	01403800
140572	FUNCTION F(Z)	01403900
140573	C*	01404000
140574	[*****]	01404100
140575	F=(1+D.1415926535897932**Z)*DCOS(D.1415926535897932*Z)	01404200
140576	RETURN	01404300
140577	END	01404400
140578	[*****]	01404500
140579	C*	01404600
140580	<u>SOLUCION ANALITICA</u>	01404700
140581	C*	01404800
140582	FUNCTION UEX(Z)	01404900
140583	C*	01405000
140584	[*****]	01405100
140585	DELTA=FXI(1.)-LXP(-1.)	01405200
140586	A=(-(1.+EXP(1.))/DELTA	01405300
140587	B=(1.+EXP(-1.))/DELTA	01405400
140588	UEX=A*EXP(Z)+B*EXP(-Z)+DCOS(D.1415926535897932*Z)	01405500
140589	RETURN	01405600
140590	END	01405700

ALEXA (04/03/87)

12:08 PM FRIDAY, APRIL 3, 1987

1000 FILE 1=MST/AUX3,UNIT=DISK,RECORD=1020,AREA=1\*200  
2000 FILE 2=MST/AUX4,UNIT=DISK,RECORD=1020,AREA=1\*200  
3000 FILE 5=MST/DATA,UNIT=DISK,RECORD=14,BLOCKING=420  
4000 FILE 6=IMP,UNIT=PRINTER,RECORD=22  
5000 FILE 7=MSG,UNIT=REMOTE  
6200 METODOS NUMERICOS DE ELEMENTOS FINITOS PARA LA -  
SOLUCION DE PROBLEMAS DE VALOR A LA FRONTERA  
6300  
6400 TESIS BASADA EN EL ARTICULO:  
6500 "A COMPUTATIONAL STUDY OF FINITE ELEMENT METHODS  
FOR SECOND ORDER LINEAR TWO-POINT BOUNDARY VALUE PROBLEMS"  
6600  
6700 BY P. KEAST, G. FAIRWEATHER & J.C. DIAZ.  
7400 METODO DE: GALERKIN  
10000 DIMENSION A(50000), B(27), HED(12), DATE(2)  
11000 CONNON NLBC, NRBC  
12000 \* NOUT, EPSL, IMPX, IMPT, UE(3).  
13000 \* UIS(3), UIJS(3,3), UXIJS(3,3)  
16000 C\* NMA = NUMBER OF MATERIALS  
17000 C\* NPT = NUMBER OF GROUPS  
19000 C\* NPT = NUMBER OF MESH POINTS  
20000 C\* NPLOT = NUMBER TO USE PLOTTER  
21000 C\*

```

22000 C*
23000 C* STATIC CALCULATION
24000 C*****
25000     TIHPS=0.
26000     TSTAT=0.
28000     TOUTP=0
29000     LD=27
30000     IF I=AX=50000
32000     TO HEAD(5,1000) HEAD
34000     READ(5,/) NMA,NGR,NPT,NPLOT
35000     IF(NMA.EQ.0) CALL EXIT
36000     CALL DATIT(DATE)
37000     WRITE(6,1100) HEAD,DATE
38000     IMPX=132
C*
39000     NLE=IABS(NI(I))-1
40000     LGH=NGR*NMA
41000     LGGH=LGH*ICR
C*
42000     ID=1
44000     ISR=ID+LGH
45000     ISF=ISR+LGH
46000     IXI=ISF+LGH
47000     ICC=IXI+LGH
48000     IX=ISS+LGGH
49000     INL=IX+I*DN(IPT)
50000     IVT=INCL*INL
C*
51000     CALL DCHTLC('IMPSTA',IVT,IF2,MAX)
C*
52000     TL=1.0E(10)
53000     CALL IIPST,(I,NGR,NPT,NLE,NL,*(ICR),A(ISR),A(ISF),A(IXI),A(ICE),
54000     * A(IX),A(INC))
55000
56000
57000
58000

```

```

00022000
00023000
00024000
00025000
00026000
00028000
00029000
00030000
00032000
00034000
00035000
00036000
00037000
00037010
00038000
00039000
00040000
00041000
00042000
00043000
00044000
00045000
00046000
00047000
00048000
00049000
00050000
00051000
00052000
00053000
00054000
00055000
00056000
00057000
00058000

```





284000	DO 5 J=1,N	00284000
285000	DO 5 I=1,M	00285000
286000	C(I,J)=0(C=J)	00286000
287000		00287000
288000	L=(M+1)/2	00288000
289000	L=L+L-1	00289000
290000	DO 10 I=1,LR-1	00290000
291000	L=I+L-1	00291000
292000	DO 10 J=1,L-1	00292000
293000	DO 10 J=2,I	00293000
294000	C(I,J-1)=C(I,J)	00294000
295000	C(I,J)=0	00295000
296000	C(I,J)=0	00296000
297000	C(I+1-2I,I+1)=0	00297000
298000	CONTINUE	00298000
299000		00299000
300000		00300000
301000	DO 10 I=1,M	00301000
302000	I=I+1	00302000
303000	I=I+1	00303000
304000	I=I+1	00304000
305000	I=I+1	00305000
306000	I=I+1	00306000
307000	I=I+1	00307000
308000	I=I+1	00308000
309000	I=I+1	00309000
310000	I=I+1	00310000
311000	I=I+1	00311000
312000	I=I+1	00312000
313000	I=I+1	00313000
314000	I=I+1	00314000
315000	I=I+1	00315000
316000	I=I+1	00316000

```

317000 DO 91 J=2,N
318000 C(I,J)=C(I,J)/C(I,1)
319000 50 CONTINUE
320000 DO 71 IR=1,M,LR
321000 I=C(I,1)
322000 V(IR)=V(IR)-I*V(1)
323000 DO 60 J=2,N
324000 C(IR,J-1)=C(IR,J)-I*C(I,J)
325000 60 CONTINUE
326000 C(IR,N)=C
327000 71 CONTINUE
328000 4F(LR,NE,N) LR=LR+1
329000 80 CONTINUE
330000 50
331000 V(N)=V(N)/C(I,1)
332000 JH=2
333000 DO 110 JR=1,MM1
334000 IR=N-JR
335000 DO 91 J=2,JH
336000 V(IR)=V(IR)-C(IR,J)*V(JR+J)
337000 91 CONTINUE
338000 IF(JH,NE,N)JH=JH+1
339000 110 CONTINUE
340000 50
341000 RETURN
342000 END
343000
344000 *****
345000 SUBROUTINE MPAC(M,L,L)
346000 C=
347000 *****
348000
349000 *****
350000
351000 *****
352000

```

```

00317000
00318000
00319000
00320000
00321000
00322000
00323000
00324000
00325000
00326000
00327000
00328000
00329000
00330000
00331000
00332000
00333000
00334000
00335000
00336000
00337000
00338000
00339000
00340000
00341000
00342000
00343000
00344000
00345000
00346000
00347000

```

342080	C*	CALCULATION OF THE ELEMENTS OF MATRIX L	00342080
342090	C*		00342090
342100		IF(N.LE.1)RETURN	00342100
342110		DO 10 I=1,1	00342110
342120		IF(I.LE.1)GO TO 30	00342120
342130		CONT	00342130
342140		IF(I.GT.M)J=1-N+1	00342140
342150		JR=I-1	00342150
342160		DO 20 J=JL,JR	00342160
342170		A(CI,M+J-1)=A(CI,M+J-1)/A(J,1)	00342170
342180		IF(J.LE.2)GO TO 20	00342180
342190		IF(J.LE.JL)GO TO 20	00342190
342200		KL=JL	00342200
342210		LR=J-1	00342210
342220		DO 30 K=KL,KR	00342220
342230		A(CI,M+J-2)=A(CI,M+J-1)-A(CI,M+K-1)*A(K,M+J-K)/A(J,M)	00342230
342240	10	CONTINUE	00342240
342250	20	CONTINUE	00342250
342260	C*		00342260
342270	C*	CALCULATION OF THE ELEMENTS OF MATRIX U	00342270
342280	C*		00342280
342290	3.	JR=N	00342290
342300		IF(I.LE.N-K+1)J=1+N-2	00342300
342310		JL=1	00342310
342320		DO 50 J=JL,JR	00342320
342330		IF(I.LE.1)GO TO 50	00342330
342340		V=1-N+1	00342340
342350		KJ=J-M+1	00342350
342360		KL=V-1	00342360
342370		IF(KJ.GT.K1)KL=KJ	00342370
342380		IF(KL.L1-1)KL=1	00342380
342390		KR=1-1	00342390
342400		DO 40 K=KL,KR	00342400



342410	A(I,M+J-1)=A(I,M+J-I)-A(I,M+K-1)+A(K,M+J-K)	00342410
342420	4) CONTINUE	00342420
342430	5) CONTINUE	00342430
342440	6) CONTINUE	00342440
342450	RETURN	00342450
342460	END	00342460
342470	*****	00342470
342480	C*	00342480
342490	SUBROUTINE B55OL(N,M,L,A,X)	00342490
342500	C*	00342500
342510	*****	00342510
342520	DIMENSION A(N,L),X(N)	00342520
342530	C*	00342530
342540	FORWARD SWEEP	00342540
342550	C*	00342550
342560	IF(M.EQ.1)GO TO 30	00342560
342570	DO 21 I=1,M	00342570
342580	IF(I.LE.1)GO TO 21	00342580
342590	JL=I	00342590
342600	IF(I.GT.M)JL=I-M+1	00342600
342610	JR=I-1	00342610
342620	DO 11 J=JL,JN	00342620
342630	X(I)=X(I)-A(I,M+J-I)*X(J)	00342630
342640	11 CONTINUE	00342640
342650	21 CONTINUE	00342650
342660	C*	00342660
342670	BACKWARD SUBSTITUTION	00342670
342680	C*	00342680
342690	DO 51 K=1,N	00342690
342700	I=N-K+1	00342700
342710	X(I)=X(I)/A(I,I)	00342710
342720	IF(M.EQ.1)GO TO 51	00342720
342730	IF(K.LE.1)GO TO 51	00342730



```

366200      DO 15 J=1,2
366300      V(I,J)=L
366400      DO 15 K=1,5
366500      XK=(DX*UE(K)+X2+XT)*L
366600      BC=FA(XK)
366700      V(IM,J)=V(I,J)+UES(K)*BC+UXIJS(K,IM)+UXIJS(K,J)
366800 15 CONTINUE
367000      DO 30 L=LMIN,LMAX
367100      XL=(DX*UE(L)+X2+XT)*L
367200      SC=FC(XL)
367300      BC=FB(XL)+GAP(XL)
368000      N=(I-1)*L+L-1+NLIC
369000      A(N,3)=A(N,3)+IJS(L,L)*SC+L*BX
370000      DO 30 K=LMIN,LMAX
371000      M=3*K-L
372000      SU A(N,M)=A(N,M)+UXIJS(L,K)*BC+UTS(L)+V(L,K)*2./DX
373000 30 CONTINUE
374000 C*
374100      DO 50 I=1,3
374200      FBX=UE(X(I))
374300      SI CONTINUE
374400      RETURN
375000      END
376000
440000 C*****
441000 C*
442000 SUBROUTINE SOURCE (M,NC,INT,NFILE,IPR,F,
443000 X,VC,SF,XT,US,FB,FI,IC)
444000 C*
445000 C*****
446000 C*
447000 C* SOURCE CALCULATED FOR APPROXIMATE
448000 C*
449000 P=MLLS, CP=(C,FC),X(NT),VC(NT,LL),

```

```

00366200
00366300
00366400
00366500
00366600
00366700
00366800
00366900
00367000
00367100
00367200
00367300
00368000
00369000
00370000
00371000
00372000
00373000
00374000
00374100
00374200
00374400
00375000
00376000
00440000
00441000
00442000
00443000
00444000
00445000
00446000
00447000
00448000
00449000

```

450000	* DEF (M1,1), X=C(1), Y=C(2), S=C(3), F=C(4), F1=C(5), F2=C(6), F3=C(7)	00450000
451000	CONTINUE	00451000
452000	* M1=C(1)*X+C(2)*Y+C(3)*S+C(4)*F+C(5)*F1+C(6)*F2+C(7)*F3	00452000
453000	* M1=C(1)*X+C(2)*Y+C(3)*S+C(4)*F+C(5)*F1+C(6)*F2+C(7)*F3	00453000
454000	CONTINUE	00454000
455000	CONTINUE	00455000
456000	CONTINUE	00456000
457000	CONTINUE	00457000
458000	CONTINUE	00458000
459000	CONTINUE	00459000
460000	CONTINUE	00460000
461000	CONTINUE	00461000
462000	CONTINUE	00462000
463000	CONTINUE	00463000
464000	CONTINUE	00464000
465000	CONTINUE	00465000
466000	CONTINUE	00466000
467000	CONTINUE	00467000
468000	CONTINUE	00468000
469000	CONTINUE	00469000
470000	CONTINUE	00470000
471000	CONTINUE	00471000
472000	CONTINUE	00472000
473000	CONTINUE	00473000
474000	CONTINUE	00474000
475000	CONTINUE	00475000
476000	CONTINUE	00476000
477000	CONTINUE	00477000
478000	CONTINUE	00478000
479000	CONTINUE	00479000
480000	CONTINUE	00480000
481000	CONTINUE	00481000
482000	CONTINUE	00482000
483000	CONTINUE	00483000
484000	CONTINUE	00484000
485000	CONTINUE	00485000
486000	CONTINUE	00486000
487000	CONTINUE	00487000
488000	CONTINUE	00488000
489000	CONTINUE	00489000
490000	CONTINUE	00490000
491000	CONTINUE	00491000
492000	CONTINUE	00492000
493000	CONTINUE	00493000
494000	CONTINUE	00494000
495000	CONTINUE	00495000
496000	CONTINUE	00496000
497000	CONTINUE	00497000
498000	CONTINUE	00498000
499000	CONTINUE	00499000
500000	CONTINUE	00500000

584100  
585100  
586000  
587000  
588000  
589000  
590000  
591000  
592000  
593000  
594000  
595000  
596000  
597000  
598000  
599000  
600000  
601000  
602000  
603000  
604000  
605000  
606000  
607000  
608000  
609000  
610000  
611000  
612000  
613000  
614000  
615000  
616000

```
SUBROUTINE CUBPL(SUBRTH,LVT,LVT1,X)  
C*  
C* *****  
LVT=LVTMAX-1/2+1  
IF (LVT.GT.L) GO TO 3  
WRITE(6,1000) SUBRTH,LVT  
CALL EXIT  
C*  
1000 FORMAT('1000 HIGH MEMORY NEEDED - "AG," - LVT =',LVT)  
30 RETURN  
END  
C* *****  
C*  
SUBROUTINE PWTY(PALE)  
C*  
C* *****  
FINAL DATE(1)  
C*  
DATE(1)=* / *  
DATE(2)=*  
DETUNE(1)  
DATE(3)=CONJUGATE(DATE(1),47,47,16)  
DATE(4)=CONJUGATE(DATE(2),47,47,16)  
DATE(5)=CONJUGATE(DATE(3),47,47,16)  
C*  
RETURN  
END  
C* *****  
C*  
END  
C* *****  
C*  
END
```

00584000  
00585000  
00586000  
00587000  
00588000  
00589000  
00590000  
00591000  
00592000  
00593000  
00594000  
00595000  
00596000  
00597000  
00598000  
00599000  
00600000  
00601000  
00602000  
00603000  
00604000  
00605000  
00606000  
00607000  
00608000  
00609000  
00610000  
00611000  
00612000  
00613000  
00614000  
00615000  
00616000

61700  
61800  
90700  
90800  
90900  
91000  
91100  
91200  
91300  
91400  
91500  
91600  
91700  
91800  
91900  
92000  
92100  
92200  
92300  
92400  
92500  
92600  
92700  
92800  
92900  
93000  
93100  
93200  
93300  
93400  
93500  
93600  
93700

RETURN  
LNH

\*\*\*\*\*  
C\*

C\* DIMENSION N(4),X(5),F(5)

\*\*\*\*\*  
C\* DIMENSION L(1,5),L(2,5),L(3,5),L(4,5),L(5,5),F(N)

C\* DATA LETV/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30  
C\* DATA LETR/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30  
C\* WRITE(6,300) N  
C\* CALL DIMMAX(L,F,N)  
C\* DO 10 I=1,5  
C\* DO 30 J=1,5

10 CONTINUE

DO 20 I=1,5

Z(I,1)=LETV(I)

L(I,5)=LETV(I)

L(I,5)=LETV(I)

20 CONTINUE

DO 30 J=1,5

L(I,1)=LETV(I)

L(I,1)=LETV(I)

30 CONTINUE

N=5\*(FMAX-FMIN)

DO 40 I=1,1

J=N\*(F(I)-FMIN)+1-5

L(I,1)=LETV(I)

40 CONTINUE

J=2

DO 50 L=1,5

00361700  
00461800  
00993700  
00993800  
00993900  
00994000  
00994100  
00994200  
00994300  
00994400  
00994500  
00994600  
00994700  
00994800  
00994900  
00995000  
00995100  
00995200  
00995300  
00995400  
00995500  
00995600  
00995700  
00995800  
00995900  
00996000  
00996100  
00996200  
00996300  
00996400  
00996500  
00996600  
00996700  
00996800  
00996900  
00997000  
00997100  
00997200  
00997300  
00997400  
00997500  
00997600  
00997700  
00997800  
00997900  
00998000  
00998100  
00998200  
00998300  
00998400  
00998500  
00998600  
00998700  
00998800  
00998900  
00999000  
00999100  
00999200  
00999300  
00999400  
00999500  
00999600  
00999700  
00999800  
00999900

930000  
939000  
940000  
941000  
942000  
943000  
944000  
945000  
946000  
947000  
948000  
949000  
950000  
951000  
952000  
953000  
954000  
9596000  
9597000  
9598000  
9599000  
960000  
9601000  
9602000  
9603000  
9604000  
9605000  
9606000  
9607000  
9608000  
9609000  
9610000  
9611000  
9612000  
9613000  
9614000  
9615000  
9616000  
9617000  
9618000  
9619000  
9620000

```
J=J-1  
IF (MOD(L-1,5).EQ.1) FLUX=FMAX-(L-1)/Q  
IF (MOD(L-1,5).EQ.2) WRITE(6,1100) LETV(L), FLUX, (IA(I,J), I=1,101)  
IF (MOD(L-1,5).NE.0) WRITE(6,1200) LETV(L), (IA(I,J), I=1,101)  
50 CONTINUE  
DO 60 I=1,5  
  U=(XMAX-XMIN)/10.  
  X(I)=U*(I-1)+XMIN  
60 CONTINUE  
WRITE(6,1300) (X(I), I=1,11)  
960000 FORMAT(33HTPLOT OF NEUTRON FLUX FOR GROUP 12,4X,  
  * 15 INVERSE POSITION //,54(1H*))//  
1100 FORMAT(3X,11,4X,1P12.5,101A1)  
1200 FORMAT(3X,11,16X,1(1A1))  
1300 FORMAT(16X,11(F3.3,2X),//,54X,18HP O S I T I O N X//)  
RETURN  
END  
C*****  
C*  
SUBROUTINE IUPSTACHINA,NGR,NPT,NELE,D,SR,SF,XI,SS,  
  * X,NC)  
C*  
C*****  
DIMENSION D(107,1MA),SR(NGR,1MA),SF(NGR,1MA)  
  * XI(NGR,1MA),SE(NGR,NGR,NMP),X(NPT),NC(NELE)  
COMMON MLDT,TRFC,  
  * NGJT,EPST,MPK,IHPT,HEFZ,  
  * U1SC3,U1JS(3,3),U1XJS(3,3)  
C*  
C* BOUNDARY CONDITIONS  
C*  
C* K(I,D(5,1),4,NC,1,NELE
```

00938000  
00939000  
00940000  
00941000  
00942000  
00943000  
00944000  
00945000  
00946000  
00947000  
00948000  
00949000  
00950000  
00951000  
00952000  
00953000  
00954000  
01094000  
01097000  
01098000  
01099000  
01100000  
01101000  
01102000  
01103000  
01104000  
01105000  
01106000  
01107000  
01108000  
01109000  
01110000  
01111000

```

1111000 CA CONTINUE END
1111001 C*
1111002     F=CBP*(T1-D)*R0000
1111003     T1=PHIME
1111004     T2=(C*Z)/F
1111005     C=1.0/T2
1111006     F=C*(C*Z)*N*(1-XC(0))
1111007     N2=(X(0)+D-C(1))/F*E*0.1*(C(1)+C)
1111008     T1=(C(1)+T2)*D+D*(1-T2)
1111009     X(1)=X(0)+F*E*0.1*(C(1)+C)*N
1111010     CONTINUE
1111011 C* CONTINUE
1111012     RETURN
1111013     END
1119000 C*****
1119001 C*
1119002     SUBROUTINE F(FC1,X(1),F,FM1,N,FMX)
1119003 C*
1119004 C*****
1119005     DIMENSION F(1)
1119006     FM1=F(1)
1119007     FMX=F(1)
1120000     DO 20 I=2,N
1120001     F(FM1+(I-1)*F(1))GO TO 10
1120002     FM1=F(1)
1120003     IF(F(1)*I.GT.F(1))GO TO 20
1120004     FMX=F(1)
1120005     CONTINUE
1120006     RETURN
1120007     END
1126000 C*****
1126001 C*

```

```

01112000
01113000
01114000
01115000
01116000
01117000
01118000
01119000
01120000
01121000
01122000
01123000
01124000
01125000
01126000
01190000
01191000
01192000
01193000
01194000
01195000
01196000
01197000
01198000
01199000
01200000
01201000
01202000
01203000
01204000
01205000
01206000
01207000
01208000
01209000

```



1264000		001ROUTINE SUBROUTINE SUBP(CNCR, NPT, I, L, E, NPT, X, Y, FU, FUX, XIMP, NT)	01264000
1265000	C*		01265000
1266000	C*		01266000
1267000	(*****		01267000
1268000		DIMENSION X(3), F(3), FUX(3), XIMP(3), IIMP(3)	01268000
1269000	C*		01269000
1270000		COMMON NCR, NPT,	01270000
1271000	*	NPT, I, L, E, NPT, X, Y, FU, FUX, XIMP, NT,	01271000
1272000	*	U1(3), U2(3), U3(3), U4(3), U5(3)	01272000
1273000	C*		01273000
1274000		YINX=X(NPT)	01274000
1275000		XIN4=X(4)	01275000
1276000		INT=1	01276000
1277000		BI=U1	01277000
1278000		BL1=U2	01278000
1279000		BL2=U3	01279000
1280000		DXIMP=(XIN4-YIN4)/FLOAT(NPT-1)	01280000
1281000		XC=U4	01281000
1282000		XIMP(N)=XIN4+XC	01282000
1283000		XC=X(NPT+1)	01283000
1284000		IF(XC, LE, XIMP(N)) GO TO 7	01284000
1285000		INT=BI	01285000
1286000		GO TO 7	01286000
1287000	1	CONTINUE	01287000
1288000		INT=BL1	01288000
1289000	2	CONTINUE	01289000
1290000		GO TO 7	01290000
1291000		GO TO 7	01291000
1292000		IF (XIN4, LE, XIMP(N)) INT=BI	01292000
1293000		IF (XIN4, LE, XIMP(N)) INT=BI	01293000
1294000		IF (XIN4, LE, XIMP(N)) INT=BI	01294000
1295000		IF (XIN4, LE, XIMP(N)) INT=BI	01295000

1296000	10 1) 1) = I * 1 / 1 + 1 / 4 * X	01296000
1297000	11 = (INT + 1) * 2 + I - 5 + NLRC	01297000
1298000	11 FOX(N, 1) = FOX(N, 1) + FO(17, NG) * UJL(C1, X(INT), X(INT+1), XIMP(N))	01298000
1299000	11 CONTINUE	01299000
1299500	11 = .3 * (FOX(N, 1) - UJL(XIMP(N)))	01299500
1299700	11 IF (1.1 * UJL(N) * 1) = 0	01299700
1299800	11 = 1.1 + E1	01299800
1299900	11 = E1 + E1 * E1	01299900
1300000	11 CONTINUE	01300000
1300100	11 = .001 * (1.1)	01300100
1300200	11 = (C7, * /) * UJL, E1, E2	01300200
1300300	C*	01300300
1300400	11 IF (NPLT, 1) = 1 RETURN	01300400
1300500	11 GO ON NEXT NG	01300500
1300600	11 CALL GFAPX(N, I, XMIN, XMAX, FOX(1, X))	01300600
1300700	11 CONTINUE	01300700
1300800	11 RETURN	01300800
1300900	11 END	01300900
1301000	11 *****	01301000
1301100	C*	01301100
1301200	11 SUBROUTINE STATIS(NMA, NGR, NPT, NLE, NPAF, U,	01301200
1301300	11 * SR, SF, XI, SC, X, NC, FO, FN, VT, LVTHAX)	01301300
1301400	11 C*	01301400
1301500	11 *****	01301500
1301600	11 DIMENSION V(NGR, NMA), SR(NGR, NMA), SF(NGR, NMA), XI(NGR, NMA),	01301600
1301700	11 * SC(NGR, NGR, NMA), X(NPT), NC(NLE), FO(NPAF, NGR),	01301700
1301800	11 * FN(NPAF, NGR), VT(1)	01301800
1301900	11 COMMON NLRC, NRDC	01301900
1302000	11 * NOUT, NRDC, IIMPX, IMPY, ME(3),	01446000
1302100	11 * UJL(3), UJL(3, 3), UJL(3, 3)	01447000
1302200	C*	01448000
1302300	C* INITIALIZATIONS	01449000
1302400	C*	01450000

1454000  
1454100  
1455000  
1456000  
1457000  
1458000  
1459000  
1460000  
1460700  
1480000  
1481000  
1482000  
1483000  
1486000  
1490000  
1491000  
1491010  
1492100  
1492110  
1492200  
1493000  
1532000  
1533000  
1534000  
1535000  
1536000  
1537000  
1539000  
1540000  
1541000  
1542000  
1543000  
1544000

```
IA=1
IAF=IA+NPAF*S+NGR
ID=IAF+NPAF*S+NGR
T=IIP+TINLP+DT(TD)
IXH=IB+NPAF
IVT=IXR+2*NELE
CALL CHCTHL("STATIC",IVT,LVTHAX)
C*
NPAF5=NPAF*S
C*
DC 165 K=1,NGR
IAK=IA+(K-1)*NPAF5
IAFK=IAF+(K-1)*NPAF5
CALL CALDIE(VT(IAK),RPT,NELF,NPAF,X)
CALL CALSCJ(VT(IAK),NGR,N01,"HE,NPAF,X,HC,SF,XT,SS,FO,FN,K")
CALL BDFAC(NPAF,3,5,VT(IAK))
CALL BDSOL(NPAF,3,5,VT(IAK),V(IAK))
DC 162 J=5,NPAF
FO(I,K)=VT(IAK-1+1)
102 FH(I,K)=VT(IAK-1+1)
145 CONTINUE
RETURN
END
(*****
C*
FUNCTION UZJ (J,X1,X2,XC)
C*
*****
HC=(X2-X1)*#E
C*
DO T=1,HC,100,J
C*
1) UZJ=(2.*X1-X2)*(XC-X2)/H
```

01454000  
01454100  
01455000  
01456000  
01457000  
01458000  
01459000  
01460000  
01460100  
01480000  
01481000  
01482000  
01483000  
01488000  
01490000  
01491000  
01491010  
01492100  
01492110  
01492200  
01493000  
01532000  
01533000  
01534000  
01535000  
01536000  
01537000  
01539000  
01540000  
01541000  
01542000  
01543000  
01544000

1545000	RETURN	
1546000	IF (UTJ-1) * (XU-1) * (XU-1) / H2	01545000
1547000	RETURN	01546000
1548000	IF (UJF-1) * (XU-1) * (XU-1) / H2	01547000
1549000	RETURN	01548000
1550000	C*	01549000
1551000	C*	01550000
1552000	C*****	01551000
1553000	C* FUNCTION F(CZ)	01552000
1554000	C*	01553000
1555000	C*****	01554000
1556000	C*****	01555000
1557000	ZT=Z	01556000
1558000	ALFA=5	01557000
1559000	F=1+ALFA+ALF*(Z-1)**2	01558000
1560000	RETURN	01559000
1570000	END	01560000
1571000	C*****	01570000
1572000	C*	01571000
1573000	C*****	01572000
1572200	C* FUNCTION F(CZ)	01572000
1572300	C*	01572100
1572400	C*****	01572200
1572500	C*****	01572300
1572600	ZT=Z	01572400
1572700	ALFA=5	01572500
1572800	F=2+ALFA*(Z-1)**2	01572600
1572900	RETURN	01572700
1573000	END	01572800
1573100	C*****	01572900
1573200	C*	01573000
1573300	C*****	01573100
1573400	C* FUNCTION F(CZ)	01573200
1573500	C*	01573300
1573600	C*****	

1573400	FCMP,	
1573500	RETURN	01573400
1573600	END	01573500
1573700	C*****	01573600
1573710	C=	01573700
1573715	FUNCTION FF(Z)	01573710
1573720	C=	01573715
1573725	C*****	01573720
1573726	ALFAS,	01573725
1573727	ZTPO,2	01573726
1573730	FAP2,2*ALFA*(Z-ZT)	01573727
1573735	RETURN	01573730
1573740	END	01573735
1573797	C*****	01573740
1573800	C=	01573797
1573900	FUNCTION FF(Z)	01573800
1574000	C=	01573900
1574100	C*****	01574000
1574200	ZTPO,2	01574100
1574300	ALFAS,	01574200
1574400	FF=2*(C1-ALFA*(Z-ZT)*CATN(ALFA*(Z-ZT))+ATAN(ALFA*Z))	01574300
1574500	RETURN	01574400
1574600	END	01574500
1574700	C*****	01574600
1574710	C=	01574700
1574800	C=	01574710
1574900	FUNCTION G(Z)	01574800
1575000	C=	01574900
1575100	C*****	01575100
1575200	ZTPO,2	01575200
1575300	ALFAS,	01575300
1575400	FF=2*(C1-ALFA*(Z-ZT)*CATN(ALFA*(Z-ZT))+ATAN(ALFA*Z))	01575400
1575500	RETURN	01575500
1575600		01575600

1575711

LHD

1575700

1000 FILE 1=MST/AUX1,UNIT=DISK,RECORD=1020,AREA=1\*200  
2000 FILE 2=MST/AUX2,UNIT=DISK,RECORD=1020,AREA=1\*200  
3000 FILE 5=MST/DATA,UNIT=DISK,RECORD=14,BLOCKING=420  
4000 FILE 6=IMP,UNIT=PRINTER,RECORD=22  
5000 FILE 7=MSG,UNIT=REMOTE  
6200 METODOS NUMERICOS DE ELEMENTOS FINITOS PARA LA  
SOLUCION DE PROBLEMAS DE VALOR A LA FRONTERA  
  
6300  
6400 TESIS BASADA EN EL ARTICULO:  
6500 "A COMPUTATIONAL STUDY OF FINITE ELEMENT METHODS  
FOR SECOND ORDER LINEAR TWO-POINT BOUNDARY VALUE PROBLEMS"  
  
6600  
6700 BY P. KEAST, G. FAIRWEATHER & J.C. DIAZ.  
7300 METODO DE: COLOCACION - GALEARKIN  
10000 DIMENSION A(50000), B(27), HED(12), DATE(2)  
11000 COMMON NLBC, NRBC.  
12000 \* NOUT, EPSL, IMPX, IMPT, UE(3),  
13000 \* UIS(3), UIJS(3,3), UXIJS(3,3), UX21JS(3)  
16000 C\* NMA = NUMBER OF MATERIALS  
17000 C\* NGR = NUMBER OF GROUPS  
19000 C\* NPT = NUMBER OF MESH POINTS  
20000 C\* NPLT = NUMBER TO USE PLOTTER

21000	C*		00021000
22000	C*		00022000
23000	C*	STATIC CALCULATION	00023000
24000	C*	*****	00024000
25000		TIME=.	00025000
26000		TSTAT=.	00026000
26500		TOUTP=.	00028000
29000		LB=27	00029000
30000		IFIPAX=50000	00030000
32000		TO READ(5,1000) HED	00032000
34000		READ(5,/) NMA,NGR,NPT,NPLOT	00034000
35000		IF(NMA.EQ.0) CALL EXIT	00035000
36000		CALL DATIT(DATE)	00036000
37000		WRITE(6,1700) HED,DATE	00037000
38000	C*		00038000
39000		NELE=IABS(NPT)-1	00039000
40000		LGH=NGR*NMA	00040000
41000		LGGM=LGM*NGR	00041000
42000	C*		00042000
42100		IMPX=132	00042100
43000		ID=1	00043000
44000		ISR=ID+LGM	00044000
45000		ISF=ISR+LGM	00045000
46000		IXI=ISF+LGM	00046000
47000		ISS=IXI+LGM	00047000
48000		IX=ISS+LGM	00048000
49000		INC=IX+IABS(NPT)	00049000
50000		IVT=INC+NGR	00050000
51000		CALL CHECKL("INPSTA",IVT,IFIPAX)	00051000
52000	C*		00052000
53000	C*		00053000
56000		TU=TIME(12)	00056000
57000		CALL INPSTA(NMA,NGR,NPT,NELE,A(ID),A(ISR),A(ISF),A(IXI),A(ISS),	00057000



58700	* A(IX),A(INC)	00058300
59700	TINPS=TINPS+DY(CT)	00059300
60000		00060300
61700		00061300
62700	C*	00062000
63000	NPAF=2*NELI-1+NLBC+NRBC	00063000
64000	C*	00064300
65700	IFO=IVT	00065000
66700	IFN=IFO+NPAF*HGR	00066000
67000	IVT=IFO+NPAF*HGR	00067000
68700	CALL CMTL('STATIC',IVT,IFN*H)	00068300
69000	LVT=IFN*H-IVT+1	00069000
70000	TL=TIME(CT)	00070000
71000	CALL STATIC(1,1,1,1,1,HPT,NELI,NPAF,1,1,1,1,A(1SR),A(1SF),A(IX),	00071000
72000	*A(1SS),A(IX),A(1IC),1,1,1,1,A(1FN),A(IVT),LVT)	00072000
73000	TSTAT=TSTAT+DT(CT)	00073000
74000	IMP=IMPX	00074000
75000	IF(NPLOT.EQ.1) NP=1	00075000
76000	IVT1=IVT+NP	00076000
77000	IVT1=IVT1+IMP*HGR	00077000
78000	IVT2=IVT1+IMP	00078300
79700	TL=TIME(CT)	00079000
83700	C*	00083100
84700	TL=TIME(CT)	00084000
85700	CALL OUTPT(HGR,HPT,NELI,NPAF,NELCT,IMP,A(IX),A(IFO),A(IVT),	00085000
86700	* A(IVT1),1)	00086000
87700	TOUTP=TOUTP+DT(CT)	00087000
88000	C*	00088000
89700	WRITE(6,100) TINPS,TSTAT,1,OUTP	00089000
93000	100) FORMAT(17F5)	00093000
94000	110) FORMAT(1I1,12G6,4X,'DATE : ',2A6/96(1H*))	00094000
95700	130) FORMAT(1H1,4X,'OVERALL TIME LOG FOR STATIC CALCULATION'/	00095300
96700	*4X,50(1H*))	00096300

237100  
238100  
239100  
241100  
251100  
253000  
254000  
255000  
256000  
257000  
258000  
259000  
260000  
261000  
262000  
263000  
264000  
265000  
266000  
267000  
268000  
269000  
270000  
271000  
272000  
273000  
274000  
275000  
276000  
277000

```
*SXX,"SUPEROUTLINE",BLX,"TIME(CPHI SPEC.)",///
*20X,"INPUTS( INPUT OF STATIC DATA).....",1PE12.5//
*3X,"STATIC(SOURCE ITERATION FOR STEADY STATE CALC.)",1PE12.5//
*23X,"OUTPUT (PRINT THE CRITICAL NEUTRON FLUXES).....",1PE12.5//
GO TO 1.
END
```

```
C*
*****
BLOCK DATA
*****
C*
COMMON NUBC,VRBC,
      R0UT,VRPS1,VRPPY,VRPI,VRP2,
      UIC(3),U1JS(3,3),U21JS(3,3),U22JS(3)
DATA UIC/
* 1.
DATA U1JS/
* 3.333333333E-11, 1.377333333E+10, 3.333333333E-11
DATA U2JS/
* 3.333333333E-11, 1.377333333E+10, 3.333333333E-11
DATA U21JS/
* 1.377333333E+10, 1.377333333E+10, 1.377333333E+10
DATA U22JS/
* 1.377333333E+10, 1.377333333E+10, 1.377333333E+10
DATA U1JJS/
* 1.377333333E+10, 1.377333333E+10, 1.377333333E+10
DATA U2JJS/
* 1.377333333E+10, 1.377333333E+10, 1.377333333E+10
END
C*
*****
```

00237000  
00238000  
00239000  
00241000  
00251000  
00253000  
00254000  
00255000  
00256000  
00257000  
00258000  
00259000  
00260000  
00261000  
00262000  
00263000  
00264000  
00265000  
00266000  
00267000  
00268000  
00269000  
00270000  
00271000  
00272000  
00273000  
00273000  
00273000  
00273000  
00274000  
00275000  
00276000  
00277000

```

278000 C*
279000 C*****
280000 DIMENSION A(N,M),C(N,M),V(N)
281000 C*
282000 DO 5 J=1,N
283000 DO 5 I=1,M
284000 5 C(I,J)=D(I,J)
285000 C*
286000 LR=(M+1)/2
287000 LRM1=LR-1
288000 DO 15 I=1,LRM1
289000 LRM2=M-LR+1
290000 DO 15 J=1,LRM2
291000 DO 15 K=1,LR
292000 C(I,J)=C(K,I,J)
293000 15 CONTINUE
294000 5 C(I,J)=C(I,J)
295000 C(I,J)=C(I,J)
296000 15 CONTINUE
297000 C*
298000 M1=M-1
299000 DO 20 I=1,M1
300000 10 V=1
301000 10 I=I+1
302000 DO 20 J=1,M1
303000 20 C(I,J)=C(I,J)+C(I+1,J)
304000 20 CONTINUE
305000 10 V=V+1
306000 10 V=V
307000 V(I)=V(I)*V
308000 V(I)=V(I)
309000 10 I=I+1
310000 20 V=V(I)

```

```

00278000
00279000
00280000
00281000
00282000
00283000
00284000
00285000
00286000
00287000
00288000
00289000
00290000
00291000
00292000
00293000
00294000
00295000
00296000
00297000
00298000
00299000
00300000
00301000
00302000
00303000
00304000
00305000
00306000
00307000
00308000
00309000
00310000

```

313000		C(I,J)=C(IPIV,J)	00311000
313500		C(IPIV,J)=1	00312000
313500	50	CONTINUE	00313000
314000	60	V(I)=V(I)/C(I,I)	00314000
314500		DO 50 J=2,N	00315000
316000		C(I,J)=C(I+J)/C(I,I)	00316000
317000	50	CONTINUE	00317000
316500		DO 70 J=1,I-1	00318000
319000		T=C(I,J)	00319000
320000		V(I)=V(I)-T*V(J)	00320000
321000		DO 60 J=2,N	00321000
322000		C(I,J-1)=C(I,J)-T*C(I,J)	00322000
325000	60	CONTINUE	00323000
324000		C(I,N)=0	00324000
325000	70	CONTINUE	00325000
326000		IF(LR.NE.N) LR=LR+1	00326000
327000	80	CONTINUE	00327000
328000	00		00328000
329000		V(N)=V(N)/C(N,1)	00329000
330000		JM=2	00330000
331000		DO 100 IRR=1,NM	00331000
332000		IR=N-IRR	00332000
333000		DO 90 J=2,N	00333000
334000		V(IR)=V(IR)-C(IR,J)*V(J)	00334000
335000	90	CONTINUE	00335000
336000		IF(JM.NE.N) JM=JM+1	00336000
337000	100	CONTINUE	00337000
338000	00		00338000
340000		END	00340000
340010	*****		
340020	00		00340010
340030		SUBROUTINE DDPA(N,M,L,A)	00340020
340040	00		00340030
			00340040

340050	C*****	00340050
340060	DIMENSION A(N,L)	00340060
340070	C*	00340070
340080	C* CALCULATION OF THE ELEMENTS OF MATRIX L	00340080
340090	C*	00340090
340100	IF(N.EQ.1)RETURN	00340100
340110	DO 6L I=1,N	00340110
340120	IF(I.LE.1)GO TO 30	00340120
340130	JL=1	00340130
340140	IF(I.GT.N)JL=I+N+1	00340140
340150	JR=1	00340150
340160	DO 21 J=JL,JR	00340160
340170	A(I,M+J-1)=A(I,M+J-1)/A(J,R)	00340170
340180	IF(I.LE.1)GO TO 21	00340180
340190	IF(J.LE.JR)GO TO 21	00340190
340200	KL=J1	00340200
340210	KR=J+1	00340210
340220	DO 11 K=KL,KR	00340220
340230	A(I,M+J-1)=A(I,M+J-1)+A(K,M+J-1)*A(K,M+J-K)/A(J,M)	00340230
340240	10 CONTINUE	00340240
340250	20 CONTINUE	00340250
340260	C*	00340260
340270	C* CALCULATION OF THE ELEMENTS OF MATRIX U	00340270
340280	C*	00340280
340290	30 J1=N	00340290
340300	IF(I.LE.N+1)JR=I+1-2	00340300
340310	JL=1	00340310
340320	DO 50 J=JL,JR	00340320
340330	IF(I.LE.1)GO TO 50	00340330
340340	K1=J-1+1	00340340
340350	KJ=J+1+1	00340350
340360	KL=K1	00340360
340370	IF(KJ.GT.K1)KL=KJ	00340370



340730  
340740  
340750  
340760  
340770  
340780  
340790  
340800  
340810  
340820  
340830  
340840  
340850  
341000  
342000  
343000  
344000  
345000  
346000  
347000  
348000  
349000  
350000  
351000  
352000  
353000  
354000  
355000  
356000  
357000  
358000  
361000  
362000

```
X(I)=X(I)/A(I,M)
IF(M.EQ.1)GO TO 6C
IF(K.LE.1)GO TO 5H
JR=N
IF(K.GT.M)JR=I+M-1
JL=I+1
DO 4I J=JL,JR
X(I)=X(I)-A(I,M+J-1)*X(J)/A(I,M)
4H CONTINUE
5H CONTINUE
6H CONTINUE
RETURN
END
```

C\*\*\*\*\*  
C\*

C\* SUBROUTINE CALDIF(A,NPT,NELE,NPAF,X)

C\*\*\*\*\*  
DIMENSION A(NPAF,5),X(NPT),V(3,3)

COMMON NLBC,NRBC

\* KOUT,EPSL,IMHX,IMPT,UR(5)

\* ULS(2),UIJS(5,3),UXIJS(3,3),UX2IJS(3)

C\*  
DO 10 M=1,5  
DO 10 N=1,NPAF  
10 A(N,M)=0.

C\*  
DO 40 I=1,NELE

X1=X(I)

X2=X(I+1)

DX=X2-X1

LIM=N-1

IF (I.EQ.1.AND.NEQ.4)OR(I.EQ.3) LIM=N-2

00340730  
00340740  
00340750  
00340760  
00340770  
00340780  
00340790  
00340800  
00340810  
00340820  
00340830  
00340840  
00340850  
00341000  
00342000  
00343000  
00344000  
00345000  
00346000  
00347000  
00348000  
00349000  
00350000  
00351000  
00352000  
00353000  
00354000  
00355000  
00356000  
00357000  
00358000  
00361000  
00362000

```

363000      LMAX=2
364000      IF (L=1,HL=0,AND,HR=BC,ER=1) LMAX=2
364100      DO 15 J=1,3
364200      DO 15 J=1,3
364300      V(IM,J)=.5
364400      DO 15 K=1,3
364500      XK=(IX*UE(K)+X2+X1)*.5
364600      DC=FA(XK)
364700      V(IM,J)=V(IM,J)+UX2(K)+DC+UX1S(K,ZI)+UX1S(K,J)
364800      15 CONTINUE
364900      DO 17 J=1,3
364920      XI=(X1+X2)*C.L
364930      DC=FA(XI)
364940      V(Z,J)=DC+UX21S(J)
364950      17 CONTINUE
365000      DO 30 L=L+1,LMAX
365100      XL=(IX*UE(L)+X2+X1)*.5
365200      SC=FC(XL)
365300      IF(L=NR,2)DC=FB(XL)+FAP(XL)
365310      IF(L=ER,2)DC=FB(XL)
366000      N=(I-1)+2*L-1+HLBC
367000      A(N,2)=A(N,3)+U1S(L,L)*SC*.5+.X
368000      DO 21 K=L+1,LMAX
369000      M=3+K-L
370000      20 A(M,1)=A(N,1)+UX1S(L,K)+BC+U1S(L,L)+V(L,K)*2./DX
371000      41 CONTINUE
372000      C*
372100      DO 51 I=1,NPT
372200      FEX=UEX(X(I))
372400      51 CONTINUE
373000      RETURN
374000      END
438000      C*****

```

```

00363000
00364000
00364100
00364200
00364300
00364400
00364500
00364600
00364700
00364800
00364810
00364820
00364830
00364840
00364850
00365000
00365100
00365200
00365300
00365310
00366000
00367000
00368000
00369000
00370000
00371000
00372000
00372100
00372210
00372400
00373000
00374000
00438000

```



```

439000 C*
440000 SUBROUTINE CUBSOU (F, HGR, HPT, HLE, HPAF,
441000 * X, HCF, HFX, HSF, FCF, FH2HG)
442000 C*
443000 C*****
444000 C*
445000 C* SOURCE CALCULATION FOR GROUP HG
446000 C*
447000 DIMENSION JND(3), X(3), HCF(3),
448000 * SF(HG, 1), X1CN(3), J1(3), HCF(1), FCF(1),
449000 * COMMON HLR(3), HPT,
450000 * HGR, HPAF, HSF, HFX, HLE(3),
451000 * HCF(3), HJ(3), HX1J(3), HX2J(3)
452000 C*
453000 C* INITIALIZE
454000 C*
455000 JND(1)=1
456000 HJ(1)=1
457000 C*
458000 C* VECTOR CALCULATION
459000 C*
460000 DO 41 I=1, HLE
461000 X1=X(I)
462000 X2=Y(I+1)
463000 BX=X2-X1
464000 I H=H+1
465000 IF(1.E0+1.ADR+HJ(3)) X1=H+1
466000 HX=X3
467000 IF(2.E0+HJ(3)) H=H+1
468000 H=31 H=H+1
469000 KK=(3-1)*H+1
470000 XK=(HX+UL(K))+CX+X1*W
471000 DO 20 I=1, H

```

```

00439000
00440000
00441000
00442000
00443000
00444000
00445000
00446000
00447000
00448000
00449000
00450000
00451000
00452000
00453000
00454000
00455000
00456000
00457000
00458000
00459000
00460000
00461000
00462000
00463000
00464000
00465000
00466000
00467000
00468000
00469000
00470000
00471000
00471200
00472000

```

475000	IF(K.NC.2)3(KK)=R(KK)+FF(XK)+HX*E.5=UIS(K)	00475000
474100	IF(K=0.2)R(KK)=R(KK)+FF(XK)+HX*E.5	00474100
481900	20 CONTINUE	00481000
482000	30 CONTINUE	00482000
483000	40 CONTINUE	00483000
484100	C*	00484000
485000	RETURN	00485000
486000	END	00486000
580000	C*****	00580000
581000	C*	00581000
582000	SUBROUTINE CHCTR=(SUBRTN,LVT,LVTMAX)	00582000
583000	C*	00583000
584000	C*****	00584000
585000	LVT=LVTMAX-LVT+1	00585000
586000	IF (LVT.GT.1) GO TO 10	00586000
587000	WRITE(6,1000) SUBRTN,LVT	00587000
588000	CALL EXIT	00588000
589000	C*	00589000
590000	1000 FORMAT('1000 HIGH MEMORY NEEDED - ",A6," - LVT =",I10)	00590000
591000	10 RETURN	00591000
592000	END	00592000
593000	C*****	00593000
594000	C*	00594000
595000	SUBROUTINE DATE1(DATE)	00595000
596000	C*	00596000
597000	C*****	00597000
598000	REAL DATE(2)	00598000
599000	C*	00599000
600000	DATE(1)=' / /'	00600000
601000	DATE(2)=' / /'	00601000
602000	D=TIME(15)	00602000
603000	DATE(1)=CONCAT(DATE(1),R,47,47,16)	00603000
604000	DATE(1)=CONCAT(DATE(1),B,23,31,16)	00604000

```
605000      DATE(2)=CONCAT(DATE(2),8,47,15,16)
606000      C=
607000      RETURN
608000      END
609000      C=*****
610000      C=
611000      FUNCTION DTCT)
612000      C=
613000      C=*****
614000      DT=(TIME(1,2)-T)*2.45-4
615000      RETURN
616000      END
617000      C=*****
618000      C=
619000      SUBROUTINE GRAFOX(N,NL,KNIN,KMAX,F)
620000      C=
621000      C=*****
622000      DIMENSION LEV(51),LETRA(3),X(11),IA(101,51),F(N)
623000      C=
624000      DATA LETV/R*1.0,2*1.0,E,2*1.0,U,2*1.0,T,2*1.0,R,2*1.0,
625000      *10,2*1.0,5*1.0,F,2*1.0,L,2*1.0,10,2*1.0,X,9*1.0 //
626000      DATA LETR/1H,1H,1H//
627000      WRITE(6,1)NL,N
628000      CALL MINMAX(N,F,FINI,FMAX)
629000      DO 10 I=1,101
630000      DO 10 J=1,51
631000      IA(J,I)=LETRA(1)
632000      10 CONTINUE
633000      DO 20 I=1,101
634000      IA(I,1)=LETRA(2)
635000      IA(I,51)=LETRA(3)
636000      20 CONTINUE
637000      DO 30 J=1, 51
```

```
03605000
04666000
00607000
00608000
00609000
00610000
00611000
00612000
00613000
00614000
00615000
00616000
00617000
00618000
00619000
00620000
00621000
00622000
00623000
00624000
00625000
00626000
00627000
00628000
00629000
00630000
00631000
00632000
00633000
00634000
00635000
00636000
00637000
00638000
00639000
00640000
00641000
00642000
00643000
00644000
00645000
00646000
00647000
00648000
00649000
00650000
```

```

920001      IAC(I,J)=LETRA(C)
921001      IAC(I+1,J)=LETRA(C2)
922001      30 CONTINUE
923001      Q=PI*W*(FMAX-FMIN)
924001      DO 41 I=1,IC1
925001      J=Q*(F(I)-F(M))+1
926001      IAC(I,J)=LETRA(C)
927001      40 CONTINUE
928001      J=J2
929001      DO 50 L=1,51
930001      J=J-1
931001      IF(MOD(L-1,2).EQ.0) FLUX=FMAY*(L-1)/2
932001      L=FMOD(L-1,2)+1; DO 50 I=1,IC1; LETV(L),FLDY,ICAC(I,J),INT,INT)
933001      IF(MOD(L-1,2).EQ.1) DO 50 I=1,IC1; LETV(L),ICAC(I,J),INT,INT)
934001      50 CONTINUE
935001      DO 60 Y=1,50
936001      W=(XMAX-Y)/50
937001      X(L)=W*(I-1)+XMIN
938001      60 CONTINUE
939001      WRITE(C2,150)ICAC(I,J),I=1,IC1)
940001      1100 FORMAT(20H POINT OF INTERSE FLUX FOR GROUP ,I2,XX,
941001      * 15PVEI SUB POSITION//,IAC(I+1,J))
942001      1100 FORMAT(CX,I2,XX,1PVEI,5,11AX)
943001      1200 FORMAT(CX,I2,15X,I2,11)
944001      1200 FORMAT(CX,I2,CFB,S+7X)//,XX,15HP ( C : I I O A X)
945001      RETURN
946001      END
1085001 *****
1086001 C*
1087001      SUBROUTINE LIPDTACHN, HCO, NPT, NELL, D, SR, SF, Y, SS,
1088001      * X, NC)
1089001 C*
1180001 *****

```

```

00920000
00921000
00922000
00923000
00924000
00925000
00926000
00927000
00928000
00929000
00930000
00931000
00932000
00933000
00934000
00935000
00936000
00937000
00938000
00939000
00940000
00941000
00942000
00943000
00944000
00945000
00946000
01085000
01086000
01087000
01088000

```

1089300	DIMENSION D(NR,NPA),SR(NR,NPA),SF(NR,NPA)	01089700
1090300	* DX(NC,NP1),SS(NC,NR,NPA),X(NP1),N(C,NLE)	01090000
1091300	COMMON /F10, F11/	01091100
1092300	* NCUT,LMN,LMN,IMPT,UF(3)	01092100
1093300	* UIC(3),UT(3,3),UX1UR(3,T),UX2UR(3)	01093000
1094300	C*	01094000
1095300	C* BOUNDARY CONDITIONS	01095000
1096300	C*	01096000
1097300	REN(5,/) NLBC,NRBC	01097000
1098300	C*	01098000
1099300	C* GEOMETRIC DATA	01099000
1100300	C*	01100000
1101300	IF (NPT.GT.0) GO TO 10	01101000
1102300	NPT=-NPT	01102000
1103300	REAF (5,/) X	01103000
1104300	GO TO 7;	01104000
1105300	10 READ (5,/) X(1),X(NPT)	01105000
1106300	DX=(X(NPT)-X(1))/FLOAT(NP1)	01106000
1107300	IF (NLE.LT.2) GO TO 30	01107000
1108300	DO 20 I=2,NLE	01108000
1109300	X(I)=X(1)+FLOAT(I-1)*DX	01109000
1110300	20 CONTINUE	01110000
1111300	30 CONTINUE	01111000
1112300	RETURN	01112000
1113300	END	01113000
1114300	C*****	01114000
1115300	C*	01115000
1116300	SUBROUTINE H(MIN,XN,F,FM1,FMX)	01116000
1117300	C*	01117000
1118300	C*****	01118000
1119300	DIMENSION F(1)	01119000
1120300	FM1=F(1)	01120000
1121300	FMX=F(1)	01121000

1187000	DO 21 I=2,N	01187000
1188000	IF(FMIN.LT.F(I))GO TO 10	01188000
1189000	FMIN=F(I)	01189000
1190000	11 IF(FMAX.GT.F(I))GO TO 20	01190000
1191000	FMAX=F(I)	01191000
1192000	20 CONTINUE	01192000
1193000	RETURN	01193000
1194000	END	01194000
1249000	C*****	01249000
1250000	C*	01250000
1251000	SUBROUTINE OUTPUT(NGR,NPT,NELE,NPAF,NPLOT,IMP,X,FO,FOX,XIMP,NT)	01251000
1252000	C*	01252000
1253000	C*	01253000
1254000	C*****	01254000
1255000	DIMENSION X(NPT),FO(NPAF,NGR),FOX(IMP,NGR),XIMP(IMP)	01255000
1256000	C*	01256000
1257000	COMMON NLBC,NRBC,	01257000
1258000	* NOUT,EPSSL,IMPX,IMPT,UE(3),	01258000
1259000	* UIS(3),IJS(3,3),UXIJS(3,3),UXZJS(3)	01259000
1260000	C*	01260000
1261000	XMAX=X(NPT)	01261000
1262000	XMIN=X(1)	01262000
1263000	I=0.0	01263000
1264000	EL1=0.0	01264000
1264500	EL2=0.0	01264500
1265000	INT=1	01265000
1266000	DX=HP*(XMAX-XMIN)/FLDAT(IMP-1)	01266000
1267000	DO 40 N=1,IMP	01267000
1268000	XIMP(N)=XMIN+DXIMP*FLOAT(N-1)	01268000
1269000	DO 5 NI=1,NELE	01269000
1269500	XC=X(NI+1)	01269500
1269800	IF(XC.LE.XIMP(N))GO TO 5	01269800
1269900	INT=NI	01269900

1269740	GO TO 7	01269040
1269750	5. CONTINUE	01269050
1269760	INT=NELE	01269060
1269770	7. CONTINUE	01269070
1270000	DO 30 NGR=1,NJR	01270000
1271000	FOX(N,NG)=0.	01271000
1272000	IMIN=1	01272000
1273000	IF (INT.EQ.1.AND.NLRC.EQ.0) IMIN=2	01273000
1274000	IMAX=3	01274000
1275000	IF (INT.EQ.4-LL-VIR.NE.NRC.IR.0) IMAX=2	01275000
1276000	DO 10 I=IMIN,IMAX	01276000
1277000	II=(INT+I)+2+I-5+NLRC	01277000
1278000	50 FOX(N,II)=FOX(N,II)+FO(I,NG)*ULJ(C1,X(INT),X(INT+1),XIMP(N))	01278000
1279000	30 CONTINUE	01279000
1279100	ERR=ABS(FOX(N,1)-LE*(XIMP(N)))	01279100
1279200	IF(ER.LT.EP) GOTO 40	01279200
1279300	EL1=EL1+ERR	01279300
1279400	EL2=EL2+ERR**2	01279400
1280000	40 CONTINUE	01280000
1280100	EL2=SQRT(EL2)	01280100
1280200	WRITE(7,*) EL1,EL2	01280200
1281000	C*	01281000
1286000	IF(NPLOT.EQ.0) RETURN	01286000
1287000	DO 40 K=1,NGR	01287000
1288000	CALL GRAFOX(K,IMP,XMIN,XMAX,FOX(*,K))	01288000
1289000	60 CONTINUE	01289000
1297000	RETURN	01297000
1298000	END	01298000
1327000	C*****	01327000
1328000	C*	01328000
1329000	SUBROUTINE STATTC(NM,NGR,NPT,NELE,NPAF,D,	01329000
1330000	* SN,SF,X1,S2,X,NC,FO,FA,VI,LVTHAX)	01330000
1331000	C*	01331000

1392000	C*****	01392000
1393000	DIMENSION 2(NGR,NMA),SR(NGR,NMA),SF(NGR,NMA),X1(NGR,NMA)	01393000
1394000	* SS(NGR,NR,1IA),X(NPT),NC(NELE),FO(NPAF,NGR)	01394000
1395000	* FN(NPAF,NGR),VT(1)	01395000
1396000	COMMON NLELE,IRRC	01396000
1397000	*  HOUT,EPOL,1NPK,1MPT,UE(3)	01397000
1398000	*  UIS(2),1IJS(3,3),UXIJS(3,3),UX2IJS(3)	01398000
1399000	C*	01399000
1400000	C*  INITIALIZATIONS	01400000
1401000	C*	01401000
1402000	IA=1	01402000
1403000	IAF=IA+NPAF*5*NGR	01403000
1404000	IB=IAF+NPAF*5*NGR	01404000
1405000	TIN1P=TIN1P+DT(TIJ)	01405000
1406000	IXR=IB+NPAF	01406000
1407000	IVT=IXR+2*NELE	01407000
1408000	CALL CMCTEL("STATIC",IVI,LVTMAX)	01408000
1409000	C*	01409000
1410000	NPAF5=NPAF*5	01410000
1411000	C*	01411000
1412000	DO 165 K=1,NGR	01412000
1413000	IAK=IA+(K-1)*NPAF5	01413000
1414000	IAFK=IAF+(K-1)*NPAF5	01414000
1415000	CALL CALDIF(VT(IAK),NPT,NELE,NPAF,X)	01415000
1416000	CALL CALSDU(VT(IB),NGR,NPT,NELE,NPAF,X,NC,SF,XI,SS,FO,FN,K)	01416000
1417000	CALL BDFAC(NPAF,3,5,VT(IAK))	01417000
1418000	CALL BDBOL(NPAF,3,5,VT(IAK),VT(IXR))	01418000
1419000	DO 162 I=1,NPAF	01419000
1420000	FO(I,K)=VT(I*3-1+1)	01420000
1421000	162 FN(I,K)=VT(IB-1+1)	01421000
1422000	165 CONTINUE	01422000
1423000	RETURN	01423000
1424000	END	01424000



1485000	C*****	01485000
1486000	C*	01486000
1487000	FUNCTION H1J (J,X1,X2,XC)	01487000
1488000	C*	01488000
1489000	C*	01489000
1490000	C*****	01490000
1491000	H2=(XC-X1)*H	01491000
1492000	C*	01492000
1493000	GO TO (1,2,3,4)J	01493000
1494000	C*	01494000
1495000	10 H1J=(C.*XC-X1-X2)+(XC-X2)/H2	01495000
1496000	RETURN	01496000
1497000	20 H1J=A.*(XC-X1)+(XC-XC)/H2	01497000
1498000	RETURN	01498000
1499000	30 H1J=(XC-X1)*(C.*XC-X1-X2)/H2	01499000
1500000	RETURN	01500000
1501000	C*	01501000
1502000	END	01502000
1503000	C*****	01503000
1503010	C* <u>PROGRAMA ARITMETICO 11.1.</u>	01503010
1503100	C*	01503100
1503200	FUNCTION F1(Z)	01503200
1503300	C*	01503300
1503400	C*****	01503400
1503500	F1=1.	01503500
1503600	RETURN	01503600
1503700	END	01503700
1504000	C*****	01504000
1504100	C*	01504100
1504200	FUNCTION F1(C2)	01504200
1504300	C*	01504300
1504400	C*****	01504400
1504700	END	01504700

1504300	RETURN	01504800
1504900	END	01504900
1505100	C*****	01505000
1505100	C*	01505100
1505200	FUNCTION FC(Z)	01505200
1505300	C*	01505300
1505400	C*****	01505400
1505410	A=30.	01505410
1505510	FC=A**2	01505500
1505600	RETURN	01505600
1505700	END	01505700
1505800	C*****	01505800
1505900	C*	01505900
1506000	FUNCTION FAP(Z)	01506000
1507000	C*	01507000
1508000	C*****	01508000
1508300	FAP=0.	01508300
1508400	RETURN	01508400
1508500	END	01508500
1508600	C*****	01508600
1508700	C*	01508700
1508800	FUNCTION FF(Z)	01508800
1508900	C*	01508900
1509000	C*****	01509000
1509200	A=20.	01509200
1509300	FF=-A**2*(DCOS(3.1415926535897932*Z))**2-2.*3.1415926535897932*	01509300
1509400	**2*DCOS(2.*3.1415926535897932*Z)	01509400
1509500	RETURN	01509500
1509600	END	01509600
1509700	C*****	01509700
1509710	C* <u>SOLUCION ANALITICA</u>	01509710
1509800	C*	01509800
1509900	FUNCTION UEX(Z)	01509900

01510000  
01511000  
01512000  
01513000  
01514000  
01515000  
01516000

C\*  
C\*\*\*\*\*  
A=21.  
UEX=(EXP(A\*(1.-2))+EXP(A\*2))/(1.+EXP(A))-(COS(3.1415926535897932  
\*2))\*\*2  
RETURN  
END

01510000  
01511000  
01512000  
01513000  
01514000  
01515000  
01516000

151000	C*	01510000
151100	C*****	01511000
151200	A=Z	01512000
151300	UC)=(EXP(A*(1-Z))+EXP(A*Z))/(1+EXP(A))-COS(3.1415926535897932	01513000
151400	**Z)**2	01514000
151500	RETURN	01515000
151600	END	01516000