



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
U.A.C.P. y P. del C.C.H.
I.I.M.A.S.

Singularidades no puntuales en la frontera *

Tesis

que para obtener el grado de
maestro en ciencias de la computación
presenta

Isaac Juan Rudomón Goldberg
México D.F. 1984

TESIS CON
FALLA DE ORIGEN



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.

CONTENIDO

Resumen

CAPITULO 1	Planteamiento del problema	pag 2
seccion 1.1	Introducción al problema	2
seccion 1.2	Comportamiento singular de el problema de interés	2:2
seccion 1.3	Proceso seguido	2:4
CAPITULO 2	Problemas Variacionales	pag 3
seccion 2.1	Metodos directos	3:3
CAPITULO 3	Introduccion a FEM y aproximación	pag 4
seccion 3.1	Aproximación de Ritz.	4:1
seccion 3.2	Metodo de Galerkin	4:4
CAPITULO 4	El metodo de elementos finitos	pag 5
seccion 4.1	Funciones base	5:1
CAPITULO 5	Un programa de elementos finitos	pag 6
seccion 5.1	Construcción del sistema de ecuaciones lineales	6:1
seccion 5.2	Integración	6:2
seccion 5.3	Factorización	6:3
CAPITULO 6	Modificaciones al método de elementos finitos	pag 7
seccion 6.1	Transformación Isoparametrica	7
seccion 6.2	Ecuaciones no-lineales	7:2
seccion 6.3	Criterio para detener la iteración.	7:3
seccion 6.4	Factorización	7:4

seccion 6.5	Integración numérica	7:5
seccion 6.6	Problema parabólico	7:7
seccion 6.7	Convergencia de la aproximación	7:9
CAPITULO 7	Singularidades	pag 8
seccion 7.1	Funciones base especiales	8
seccion 7.2	Elementos con punto cuarto	8:2
seccion 7.3	Refinamiento local	8:4
seccion 7.4	Diferencias Finitas y Elemento Finito	8:6
CAPITULO 8	Resultados en el caso unidimensional	pag 9
seccion 8.1	Problemas independientes del tiempo id.	9:1
seccion 8.2	Problemas parabólicos id.	9:7
CAPITULO 9	Sobre el caso bidimensional	pag 10
seccion 9.1	Transformación isoparamétrica	10:1
seccion 9.2	La inversa a la transformación isoparamétrica	10:2
seccion 9.3	Integración Numérica	10:4
seccion 9.4	Elementos singulares con punto cuarto	10:5
seccion 9.5	Refinamiento local	10:7
CAPITULO 10	Resultados en el caso bidimensional	pag 11
seccion 10.1	El problema	11:1
seccion 10.2	Descripción de los demás problemas	11:2
CAPITULO 11	Convergencia y Conclusiones	pag 12
seccion 11.1	Resumen de convergencia	12
seccion 11.2	Conclusiones	12:1
subseccion 11.2.1	El caso unidimensional	12:2
subseccion 11.2.2	El caso bidimensional	12:3

Bibliografía

Apendices:

Tablas y Graficas

Figuras

Programas

Singularidades no-puntuales en la frontera.

Isaac Rudomin Goldberg

Resumen

En este trabajo se analiza la ecuación de difusión:

$$\frac{\partial u}{\partial t} = \nabla \cdot (p \nabla u) - q u + f$$

en donde $u=0$ en parte de la frontera, digamos $\partial \Omega_1$, y $n \cdot \nabla u = 0$ (i.e. la componente normal) en el resto de la frontera, es decir en $\partial \Omega_2$. En esta ecuación, $p = p(x, y, t, u) \geq 0$, $q = q(x, y, t, u) \geq 0$ y $f = f(x, y, t, u)$ de lo que resulta una ecuación parabólica no-lineal.

Hasta ahora se ha usado con éxito el método de elementos finitos para ecuaciones elípticas lineales, con singularidades puntuales en la frontera haciéndole modificaciones apropiadas al método (Whiteman y Wait). Ha sido utilizado también en el caso de ecuaciones parabólicas sin singularidades (Wait). En el problema arriba planteado, para $p=0$ la no-linealidad de la ecuación produce soluciones singulares, donde las singularidades son no puntuales (i.e. singularidades en toda una parte de la frontera) que permanecen al avanzar en el tiempo.

Se aplican las modificaciones al método de elemento finito sugeridas por Whiteman y Wait para ecuaciones elípticas con singularidades puntuales, es decir los métodos de elemento singular con puntos cuartos, de funciones base singulares y refinamiento local. Se trata básicamente el caso unidimensional, buscándose posteriormente extender los resultados encontrados al caso bidimensional.

En los problemas unidimensionales funcionan los tres métodos, es decir al utilizarlos se obtiene una mejor aproximación a la solución que con el método de elementos finitos con elementos regulares. Para mallas gruesas, es mejor utilizar refinamiento local, pero también mucho más caro pues requiere mayor tiempo de cómputo. Para mallas finas el incremento en tiempo de cómputo es relativamente menor, pero los resultados ya no son distinguibles de los otros métodos.

En el caso bidimensional se obtiene el comportamiento esperado al aplicar refinamiento local, es decir, sucede lo mismo que en el caso unidimensional. Sin embargo, y esto es muy importante, en el caso del método de elementos con punto cuarte no se encontró una mejor aproximación a la solución como se esperaba; sin embargo los experimentos hechos no son concluyentes. Se discuten varias explicaciones posibles, pero los resultados obtenidos no permiten decidir cual es la verdadera causa del comportamiento observado.

En ambos casos (uni y bi-dimensionales) al resolver problemas independientes del tiempo, es decir problemas elípticos, como si fueran parabólicos no fue posible asegurar (con los resultados obtenidos) si el proceso estaba convergiendo a la solución como teóricamente se esperaba. Con dos o tres pasos en el tiempo sería poco probable observar tal convergencia.

El presente trabajo se ha dividido en capítulos, en los cuales se tratan los siguientes temas:

- 1) En el primer capítulo se describe el problema que nos interesa, su procedencia e importancia.
- 2) En el capítulo 2 se hace una pequeña introducción a problemas variacionales, que sirve de base teórica al método de elementos finitos.
- 3) En los capítulos 3 y 4 se describe muy someramente el método de elemento finito, tratando aspectos teóricos.

4) En los capítulos 5, 6 y 7 se habla de la implementación, así como las diversas adaptaciones al método necesarias para tratar el problema de interés; asimismo se discuten las modificaciones al método que han sido utilizadas para la solución de ecuaciones singulares.

5) En el capítulo 8 se muestran los resultados obtenidos al aplicar estos métodos al problema de interés en el caso unidimensional.

6) En los capítulos 9 y 10 se estudia la extensión de los resultados anteriores al caso bidimensional.

7) En el capítulo 11 se analiza la convergencia del método en los diferentes casos y se resumen las conclusiones generales del trabajo.

CAPITULO 1 Planteamiento del problema

Una aplicación del método de elementos finitos (en lo sucesivo denominado F. E. M.) que ha recibido mucha atención es el estudio de las singularidades. England y Hennart analizan un problema de difusión que surge en estudios sobre simulación de plasma en dispositivos de fusión (como puede verse en [1] y [2]). Este problema lleva a la ecuación

$$\frac{\partial u}{\partial t} = \sum_{i,j=1}^2 \frac{\partial}{\partial x_i} \left(D_{ij}(x_1, x_2, u) \frac{\partial u}{\partial x_j} \right) + f(x_1, x_2, t) \quad (x_1, x_2, t) \in \Omega \times (0, T]$$

donde Ω es una sección del plano (x_1, x_2) , $D(x_1, x_2, u)$ es el tensor de difusión y $f(x_1, x_2, t)$ es una fuente externa de plasma. Tenemos la condición inicial $u(x_1, x_2, 0) = u_0(x_1, x_2)$ $(x_1, x_2) \in \Omega$ y las condiciones de frontera:

$$\frac{\partial u}{\partial n} u(x_1, x_2, t) = n \cdot \nabla u(x_1, x_2, t) = 0 \quad \forall (x_1, x_2, t) \in \int \Omega_2 \times (0, T]$$

$$u(x_1, x_2, t) = 0 \quad \forall (x_1, x_2, t) \in \int \Omega_1 \times (0, T]$$

En los trabajos arriba citados, England y Hennart utilizan el método de elementos finitos para encontrar la solución al problema descrito. Tuvieron problemas para alcanzar la precisión deseada en casos singulares en que $\beta=0$ en parte de la frontera, lo cual es una condición que surge frecuentemente. Este problema motivó el presente trabajo.

sección 1.1 Introducción al problema

En este trabajo se considera el uso de F. E. M. para resolver aquellos

problemas que son parabólicos y conservan las singularidades originales a lo largo del tiempo. El método en su forma usual, es decir cuando se usan elementos regulares, produce buenas aproximaciones a problemas como el de difusión si las soluciones son suficientemente suaves (ver [8]). Sin embargo cuando se tienen singularidades, la función de aproximación del método de elementos finitos, que es casi siempre una función polinomial por trozos, es incapaz de reproducir la forma singular. Esto es causa de que puedan ocurrir grandes errores cerca de la singularidad.

Hay varias técnicas que se han aplicado a casos con singularidades puntuales en ecuaciones elípticas, por ejemplo las descritas en las referencias de Whiteman y Wait ([3], [4] y [8]). Nos interesa analizar que sucede en el caso parabólico con singularidades que persisten a lo largo del tiempo y se localizan en trozos de la frontera.

Se analiza una ecuación de difusión generalizada semejante a la estudiada por England & Hennart, a saber:

$$\frac{\partial u}{\partial t} = \nabla \cdot (\rho \nabla u) - qu + f \quad (1)$$

en donde $u=0$ en parte de la frontera, digamos $\partial\Omega_1$ y $\nabla u=0$ en el resto de la frontera, es decir $\partial\Omega_2$. En esta ecuación:

$$\rho = \rho(x, y, t, u) \geq 0$$

$$q = q(x, y, t, u) \geq 0$$

$$f = f(x, y, t, u)$$

de lo que resulta una ecuación parabólica no-lineal.

Hasta ahora se ha usado con éxito el método de elementos finitos para ecuaciones elípticas lineales, con singularidades puntuales en la frontera, y para ecuaciones parabólicas sin singularidades. En este trabajo se aplican estos métodos al problema arriba descrito, que es parabólico, y en el que encontramos singularidades no puntuales (i.e. singularidades en toda una parte de la frontera) que son producidas por la no-linealidad de la ecuación cuando se tiene $\rho=0$. Estas singularidades se mantienen a lo largo del tiempo.

seccion 1.2

Comportamiento singular de el problema de interés

Veamos un problema semejante al arriba planteado (por simplificar el analisis aqui presento el caso unidimensional; un análisis semejante se haria en un caso bidimensional):

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(u \frac{\partial u}{\partial x} \right)$$

Supongamos que la solución tiene la forma

$$u = u(t, x) = \phi(x) \psi(t)$$

Sustituyendo, resulta

$$\phi \frac{\partial \psi}{\partial t} = \psi^2 \frac{\partial}{\partial x} \left(\phi \frac{\partial \phi}{\partial x} \right)$$

utilizando separación de variables se tiene:

$$\frac{1}{\psi^2} \frac{\partial \psi}{\partial t} = \kappa \quad \frac{1}{\phi} \frac{\partial}{\partial x} \left(\phi \frac{\partial \phi}{\partial x} \right) = \kappa \quad (2)$$

donde κ es constante.

Se propone

$$\phi = \phi(x) = a_0 x^{p_0} + a_1 x^{p_1} + \dots$$

donde

$$0 < p_0 < p_1 < \dots$$

y $a_0 \neq 0$ ($\rho_0 > 0$ para que $\mu_0 = 0$ en la singularidad que se supone en $x=0$).

Entonces

$$\frac{\partial \phi}{\partial x} = \rho_0 a_0 x^{(\rho_0-1)} + \rho_1 a_1 x^{(\rho_1-1)} + \dots$$

de donde

$$\begin{aligned} \phi \frac{\partial \phi}{\partial x} &= \rho_0 a_0^2 x^{(2\rho_0-1)} + \rho_1 a_1^2 x^{(2\rho_1-1)} + \\ &+ a_0 a_1 (\rho_0 + \rho_1) x^{(\rho_0 + \rho_1 - 2)} + \\ &+ \dots \end{aligned}$$

sustituyendo resulta:

$$\begin{aligned} \frac{\partial}{\partial x} \left(\phi \frac{\partial \phi}{\partial x} \right) &= \rho_0 a_0^2 (2\rho_0-1) x^{(2\rho_0-2)} + \rho_1 a_1^2 (2\rho_1-1) x^{(2\rho_1-2)} \\ &+ a_0 a_1 \rho_0 (\rho_0 + \rho_1 - 1) x^{(\rho_0 + \rho_1 - 2)} \\ &+ a_0 a_1 \rho_1 (\rho_0 + \rho_1 - 1) x^{(\rho_0 + \rho_1 - 2)} + \\ &+ \dots \end{aligned}$$

Si $\psi(t) \neq 0$ entonces de (2) resulta que

$$\frac{\partial}{\partial x} \left(\phi \frac{\partial \phi}{\partial x} \right) - \kappa \phi = 0$$

implica que el coeficiente del término dominante debe ser nulo (ningún otro término lo puede anular) es decir que $\rho_0 a_0^2 (2\rho_0 - 1) = 0$ pero como $a_0 \neq 0$ necesariamente $\rho_0 (2\rho_0 - 1) = 0$. Como $\rho_0 > 0$ entonces $2\rho_0 - 1 = 0$ es decir $\rho_0 = \frac{1}{2}$. Por ser el término dominante es claro que μ tiene la forma \sqrt{x} .

seccion 1.3 Proceso seguido

Como ya se mencionó, el propósito de este trabajo es resolver el problema $\frac{\partial u}{\partial t} = \nabla \cdot (\rho \nabla u) - \gamma u + f$ (1)

con las condiciones $u=0$ en $\partial\Omega_1$ y $n \cdot \nabla u = 0$ en $\partial\Omega_2$. El proceso que se utilizó para resolver problemas de este tipo se describe a continuación:

En primer lugar se examinó el caso elíptico donde $u=u(x)$. En tal caso el problema queda expresado como

$$\frac{\partial}{\partial x} \left(p \frac{\partial u}{\partial x} \right) - q u + f = 0$$

Despues se considero el caso parabólico $u=u(x,t)$, en el que resulta

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(p \frac{\partial u}{\partial x} \right) - q u + f$$

Finalmente se trabajo con el caso bidimensional y parabólico donde se tiene $u=u(x,y,t)$:

$$\frac{\partial u}{\partial t} = \nabla \cdot (r \nabla u) - q u + f$$

Se contaba con el programa PVF/FEMQ (de J. P. Hennart) que trata el caso elíptico lineal en una dimensión con una malla uniforme y una base cuadrática. Se procedio entonces a convertir el programa en isoparametrico (ya que así se hace mas sencilla la inclusión de mallas con refinamiento local y elementos con nodos al punto cuarto, dos de las tecnicas que aplicaremos para tratar las singularidades).

Después se consideró la posibilidad de incluir las ecuaciones no lineales. Posteriormente se incluyeron los métodos para tratar las singularidades, despues de lo cual se modificó el programa para resolver el caso parabólico.

Una vez completada esta fase el programa se transformó en uno para dos dimensiones, lo cual no fue trivial debido al incremento significativo del tiempo de proceso si se utilizan simplistamente los mismos métodos.

Las singularidades fueron tratadas de tres maneras distintas, con el objeto de obtener mejores aproximaciones a la solución de la ecuación diferencial en relación con el método de elementos finitos con elementos regulares:

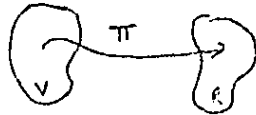
1) Modificando el tipo de elementos cerca de la singularidad, usando los llamados elementos singulares con puntos cuartos, que han sido utilizados con éxito en singularidades puntuales en problemas elípticos por Wait (ver [4]).

2) Mediante un refinamiento local alrededor de la singularidad. Este método también ha sido ensayado con éxito (en problemas elípticos) por Whiteman y Wait (ver [3] , [4]) entre otros.

3) Utilizando funciones base especiales como ha hecho Whiteman (ver [3]) también para casos elípticos.

CAPITULO 2 Problemas Variacionales

Sea $V: \mathbb{R} \rightarrow \mathbb{R}$ un espacio de funciones; se llama funcionales a los mapeos $\Pi: V \rightarrow \mathbb{R}$:



$$f \in V \quad f(x) \in \mathbb{R}$$

$$\Pi(f(x)) \in \mathbb{R}$$

Los métodos de resolución de problemas variacionales, es decir problemas sobre obtención de máximas y mínimos de los funcionales son semejantes a los utilizados para obtener extremos de funciones. Necesitamos las siguientes definiciones (ver [11]):

- 1) Si Π es un funcional que depende de $y(x)$ la denotamos por $\Pi(y(x))$.
- 2) El incremento o variación δy del argumento $y(x)$ del funcional $\Pi(y(x))$ es la diferencia entre dos funciones $\delta y = y(x) - y_0(x)$ donde $y(x)$ varía arbitrariamente sobre cierta clase de funciones.
- 3) El funcional $\Pi(y(x))$ es continuo para $y = y_0(x)$ en el sentido de proximidad de k -ésimo orden si para todo $\epsilon > 0$ existe un $\delta > 0$ tal que

$$| \Pi(y(x)) - \Pi(y_0(x)) | < \epsilon$$

para

$$| y(x) - y_0(x) | < \delta$$

$$| y'(x) - y_0'(x) | < \delta$$

⋮

$$| y^{(k)}(x) - y_0^{(k)}(x) | < \delta$$

(tomando la función $y_0(x)$ de la clase de funciones sobre la cual el funcional $\Pi(y(x))$ está definido).

4) Un funcional es lineal si satisface

$$\pi(c y(x)) = c \pi(y(x))$$

$$\pi(y_1(x) + y_2(x)) = \pi(y_1(x)) + \pi(y_2(x))$$

donde c es una constante arbitraria y y_1, y_2 funciones en el dominio adecuado.

5) Si

$$\Delta \pi = \pi(y(x) + \delta y) - \pi(y(x))$$

puede representarse como

$$\Delta \pi = L(y(x), \delta y) + \beta(y(x), \delta y) \max |\delta y|$$

donde $L(y(x), \delta y)$ es un funcional lineal respecto a δy y $\beta(y(x), \delta y) \rightarrow 0$ cuando $\max |\delta y| \rightarrow 0$ entonces $L(y(x), \delta y)$ es la variación del funcional $\pi(y(x))$ y se representa como $\delta \pi$. Es la parte principal del incremento.

Para funcionales $\pi(y(x))$ (o más complejos) podemos definir la variación como la derivada del funcional $\pi(y(x) + \alpha \delta y)$ con respecto a α , para $\alpha = 0$ es decir:

$$\delta \pi = \left. \frac{\partial}{\partial \alpha} \pi(y(x) + \alpha \delta y) \right|_{\alpha = 0}$$

Si el funcional $\pi(y(x))$, que posea variación, alcanza su extremo para $y = y_0(x)$ siendo $y_0(x)$ un punto interior de la región de definición del funcional, entonces para $y = y_0(x)$ tendremos $\delta \pi = 0$.

En el caso de funcionales

$$\pi(y(x)) = \int_{x_0}^{x_1} F(x, y, y') dx$$

con puntos de frontera fijos para las curvas admisibles $y(x_0) = y_0$ y $y(x_1) = y_1$, tendremos que si consideramos los valores del funcional solo en la familia de curvas $y = y(x, \alpha) = y_0 + \alpha \delta y$ que contiene para $\alpha = 0$ la curva en la cual se alcanza el extremo, el funcional se transforma en función de α , $\Pi(y(x, \alpha)) = \ell(\alpha)$ ya que el valor del parámetro α determina una curva de la familia $y = y(x, \alpha)$ y por lo tanto el valor de $\Pi(y(x, \alpha))$. Esta función $\ell(\alpha)$ tiene un extremo en $\alpha = 0$ puesto que para este valor se obtiene $y = y(x)$.

La condición necesaria para que $\Pi(y(x))$ tenga un extremo en $\ell(\alpha)$ es $\ell'(\alpha) = 0$ y como

$$\ell(\alpha) = \int_{x_0}^{x_1} F(x, y(x, \alpha), y'(x, \alpha)) dx$$

entonces, considerando $y = y(x)$ y $y' = y'(x)$ como variables:

$$\ell'(\alpha) = \int_{x_0}^{x_1} \left[F_y \frac{\partial}{\partial \alpha} y(x, \alpha) + F_{y'} \frac{\partial}{\partial \alpha} y'(x, \alpha) \right] dx$$

donde

$$F_y = \frac{\partial}{\partial y} F(x, y(x, \alpha), y'(x, \alpha))$$

$$F_{y'} = \frac{\partial}{\partial y'} F(x, y(x, \alpha), y'(x, \alpha))$$

Como

$$\frac{\partial}{\partial \alpha} y(x, \alpha) = \frac{\partial}{\partial \alpha} (y(x) + \alpha \delta y) = \delta y \quad \frac{\partial}{\partial \alpha} y'(x, \alpha) = \frac{\partial}{\partial \alpha} (y'(x) + \alpha \delta y') = \delta y'$$

entonces

$$\ell'(\alpha) = \int_{x_0}^{x_1} \left[F_y(x, y(x), y'(x)) \delta y + F_{y'}(x, y(x), y'(x)) \delta y' \right] dx$$

como habiamos dicho, entonces si $\mathcal{L}'(a) = \delta\pi$, para un extremo se tiene $\delta\pi = 0$ es decir que si

$$\pi(y(x)) = \int_{x_0}^{x_1} F(x, y, y') dx$$

se necesita

$$\int_{x_0}^{x_1} (F_y \delta y + F_{y'} \delta y') dx = 0$$

Como $\delta y|_{x=x_0} = 0$ y $\delta y|_{x=x_1} = 0$, entonces la condición necesaria de extremo toma la forma

$$\delta \pi = \int_{x_0}^{x_1} (F_y - \frac{\partial}{\partial x} F_{y'}) \delta y dx = 0$$

Basta tener

$$F_y - \frac{\partial}{\partial x} F_{y'} = 0$$

en la curva $y = y(x)$ que realiza el extremo del funcional π considerado. Esto se generaliza para funcionales que dependen de las funciones, de funciones de variables independientes, o de derivadas de mayor orden.

Las ecuaciones diferenciales de los problemas variacionales se integran de manera analítica (como en el ejemplo de la sección anterior) solo excepcionalmente, por lo que surgen los llamados (ver [1]) métodos directos cuya idea es considerar al problema variacional como límite de un problema de extremo para una función de un número finito de variables, solucionándose este por métodos comunes, y con el paso al límite obtener la solución del problema correspondiente.

A continuación menciono dos métodos directos comunmente usados:

1) El método de diferencias finitas de Euler, que no considera los valores del funcional en las curvas arbitrarias admisibles en el problema variacional dado, sino en las líneas quebradas formadas por un número dado de segmentos rectilíneos, cuyas abscisas están dadas (esto es en una dimensión) y donde el funcional $\Pi(y(x))$ se transforma en una función $f(y_1, y_2, \dots, y_n)$ de las ordenadas de estos vértices, determinando y_1, y_2, \dots, y_n del sistema de ecuaciones:

$$\frac{\partial f}{\partial y_1} = 0, \quad \frac{\partial f}{\partial y_2} = 0, \quad \dots, \quad \frac{\partial f}{\partial y_n} = 0$$

obteniendo una solución aproximada al problema variacional.

2) El llamado método de Ritz (que detallaremos más adelante) que considera los valores del funcional no en curvas arbitrarias admisibles para el problema variacional dado, sino solo en todas las combinaciones lineales:

$$U_n(x) = \sum_{i=1}^n U_i \gamma_i(x)$$

con coeficientes constantes posibles formadas por las primeras funciones de cierta sucesión elegida de funciones $\gamma_1(x), \gamma_2(x), \dots, \gamma_n(x)$. Tales combinaciones lineales deben ser admisibles en el problema considerado, por lo que hay ciertas limitaciones para escoger las $\gamma_i(x)$.

El funcional $\Pi(y(x))$ se transforma en una función de los coeficientes U_1, U_2, \dots, U_n los que se escogen para que $f(U_1, U_2, \dots, U_n)$ tenga un extremo, es decir se determinan del sistema $\frac{\partial f}{\partial U_i} = 0 \quad 1 \leq i \leq n$.

Si nos limitamos a los primeros términos

$$U_n(x) = \sum_{i=1}^n U_i \gamma_i(x)$$

se tiene una solución aproximada al problema variacional. Para que las U_n sean admisibles hay que satisfacer ante todo las condiciones de frontera. La elección de la sucesión de funciones $\gamma_1, \gamma_2, \dots, \gamma_n$ llamadas funciones base, influye considerablemente en el grado de complejidad de los cálculos ulteriores, como se analiza posteriormente.

CAPITULO 3 Introducción al método de elementos finitos

El método de elementos finitos (ver [6] , [7] y [5]) consiste en una descripción aproximada de un continuo en términos de un número finito, aunque grande, de partes que puedan representarse por pocos parámetros, llamados grados de libertad. El concepto se origina en la teoría de estructuras, donde una estructura física se construye usualmente a partir de elementos estructurales, los cuales se interconectan solo en determinados puntos llamados nodos.

Para cada elemento, las propiedades estructurales (por ejemplo las relaciones fuerza-desplazamiento) están definidas en forma única en términos de los grados de libertad permitidos en los nodos. A partir del comportamiento en cada nodo, es posible describir el de la estructura ensamblada.

Dado un continuo, este es dividido en un número finito de pequeñas regiones ajenas; se les llama elementos. Podemos visualizar a estos elementos como interconectados solo en un cierto número de puntos discretos en sus fronteras comunes. A estos puntos, además de algunos puntos especiales escogidos en el interior de los elementos, se les llama nodos. A cada nodo están asociados un número finito de grados de libertad. El comportamiento de los elementos está caracterizado por estos grados de libertad, que son las coordenadas generalizadas de la mecánica.

En mecánica usualmente se describe un problema de campo mediante un conjunto de ecuaciones diferenciales con las condiciones de frontera apropiadas, o por el extremo de un principio variacional. Si este extremo no existe basta un enunciado variacional más débil. Con las restricciones adecuadas, los dos puntos de vista son matemáticamente equivalentes (ver Tong & Rossetto [pag 7-8] donde se demuestra esto).

Como ya se mencionó, dado un problema diferencial, en general se tiene un problema variacional asociado. Para mayores detalles vease el segundo capítulo de Mitchell & Wait, en donde se indican ciertos principios variacionales comunes, y su relación con problemas diferenciales.

sección 3.1 Aproximación de Ritz.

A continuación se describe un método para aproximar la solución de la ecuación diferencial elíptica lineal $Au = v$, dadas ciertas condiciones de frontera (es el mismo método anteriormente resumido en la sección 2.1). Si suponemos que el operador A es tal que la función u es la solución al problema variacional $\delta \Pi(u) = 0$ para algún funcional $\Pi(u)$, entonces puede escribirse

$$\Pi(u) = \frac{1}{2} (Au, u) - (v, f)$$

donde

$$(u, f) = \int_R u(x) f(x) dx$$

para R una región de integración.

Además requerimos que el funcional Π opere sobre un espacio $H^{(1)}$ de funciones f tal que, si denotamos $\frac{\partial u}{\partial s}$ como $u^{(1)}$ se cumple que

$$\int (u^{(1)})^2 dx < \infty.$$

Un espacio con tales características es llamado espacio de Sobolev.

Dada X y funciones base, que llamaremos

$$\psi_i(x) \quad 1 \leq i \leq n$$

podemos aproximar la solución mediante la siguiente interpolación:

$$U(x) = \sum_{i=1}^n U_i \psi_i(x)$$

El procedimiento de minimizar un funcional utilizando una aproximación de esta forma es conocido como aproximación de Ritz. Para la aproximación se utiliza un subespacio de dimensión finita del espacio de Sobolev.

El procedimiento a seguir en este caso ha sido descrito por Tong y Rossetto (ver [7] [pag 30-36]). Baste decir que a partir del problema variacional podemos encontrar la solución en cada elemento (suponiéndolo aislado) digamos $\bar{u} = (U_i)$ como la solución de $\bar{a} \bar{u} = \bar{f}$ donde para $\bar{a} = (a_{ij}) \quad 1 \leq i, j \leq n$, es la que llamamos matriz de rigidez del elemento y $\bar{f} = (f_i) \quad 1 \leq i \leq n$ el vector de fuerza (también del elemento) que se obtienen del funcional al aplicarlo a las funciones base del elemento, como se describirá más adelante.

Con los resultados para cada elemento, si tenemos que $1 \leq I, J \leq N$, donde N es el número total de nodos es posible encontrar lo que se llama la matriz de rigidez del sistema, $\bar{A} = (A_{IJ})$ y el vector de fuerza $\bar{F} = (F_I)$ (también del sistema) de manera que al resolver $\bar{A} \bar{U} = \bar{F}$ obtenemos la solución global aproximada $\bar{U} = (U_I)$. Las U_i generalmente son valores de U en los nodos, i. e. $U_i = U(x_i)$.

El siguiente ejemplo ilustra el procedimiento a seguir. Dada la ecuación

$$-\frac{\partial}{\partial x} \left(p \frac{\partial u}{\partial x} \right) + q u = f$$

con condiciones de frontera $u(x_p) = u(x_f) = 0$ (donde el intervalo $[x_p, x_f]$ denota por x_p el valor al principio y por x_f el valor al final), el funcional apropiado es

$$\Pi(u) = \int_{x_p}^{x_f} \left(\frac{1}{2} p(x) (u')^2 + \frac{1}{2} q(x) u^2 - f(x) u \right) dx$$

es decir $\tilde{\Pi}(U) = \frac{1}{2} \bar{U}^T \bar{A} \bar{U} - \bar{b}^T \bar{U}$ que para cada elemento se ve como

$$\tilde{\Pi}_i(U) = \frac{1}{2} \bar{u}^T \bar{a} \bar{u} - \bar{b}^T \bar{u}$$

Por lo tanto el problema se reduce a minimizar

$$\tilde{\Pi}(U) = \sum_{I=1}^M \left(\frac{1}{2} \bar{u}^T \bar{a} \bar{u} - \bar{b}^T \bar{u} \right)$$

donde M es el número total de elementos. Entonces para obtener $\delta \Pi(U) = 0$ respecto a U resulta que basta resolver el sistema $\bar{A} \bar{U} = \bar{b}$ con las condiciones de frontera adecuadas.

En resumen los pasos a seguir son:

- 1) discretizar el sistema físico.
- 2) determinar las $\bar{a} = (a_{ij})$ y $\bar{b} = (b_i)$ a partir de principios variacionales y procedimientos de interpolación adecuados, caracterizando así los elementos.
- 3) ensamblar los a_{ij} , b_i de los elementos para obtener los $\bar{A} = (A_{IJ})$, $\bar{b} = (b_I)$ globales.
- 4) aplicar las restricciones apropiadas para obtener un sistema de ecuaciones algebraicas soluble.
- 5) resolver tal sistema.

Todo esto si tenemos que el operador A es positivo-definido y simétrico. Nuestro caso es distinto, por lo cual tenemos que considerar de otra manera la cuestión, como se ve en seguida.

seccion 3.2 Método de Galerkin

Si la ecuacion diferencial en su forma clásica

$$Au = v$$

se multiplica por funciones $f(x)$ (llamadas de prueba) y se integra sobre Ω (utilizando el Teorema de Green o sea integración por partes) se obtiene

$$(Au, f) = (v, f)$$

donde

$$(u, f) = \int_{\Omega} u(x) f(x) dx$$

que es llamada la forma debil de la ecuacion diferencial.

La aproximacion de Ritz a la solución del problema variacional.

$$U(x) = \sum_{i=1}^n U_i \gamma_i(x)$$

se puede escribir como la solución de

$$\frac{\partial}{\partial U_i} (\Pi(U)) = (AU, \gamma_i) - (\gamma_i, v) = 0 \quad i=1, \dots, n$$

solo si el operador lineal A es positivo-definido y simétrico.

Pero si esto no sucede, el sistema de ecuaciones $(AU, \gamma_i) - (\gamma_i, v) = (AU - v, \gamma_i) = 0$

todavía define una solución aproximada. El método de Galerkin no es sino una discretización de la forma débil. Dado un subespacio S^h del espacio de soluciones y un subespacio V^h del espacio de prueba V tenemos la solución de Galerkin U es el elemento de S^h que satisface

$$(A U, f) = (v, f).$$

El lado izquierdo requiere de s integraciones por partes (ver Strang & Fix [pags. 117-118] y Mitchell & Wait [pags 50-52]).

La aproximación de Galerkin arriba construida satisface

$$J^h(U, \psi_i) = (v, \psi_i)$$

donde

$$J^h(u, f) \equiv (v, f)$$

es decir la integración por partes ya mencionada.

En las ecuaciones de Galerkin es natural definir la aproximación U y las funciones V usando el mismo conjunto de funciones base ψ_i donde $1 \leq i \leq n$, asegurando así que en aquellos problemas en que tanto la aproximación de Ritz y de Galerkin puedan definirse, lleven a formulaciones alternativas de la misma función. Hay otras aproximaciones en las que no se escoge la misma base, un ejemplo de esto es el método de mínimos cuadrados. Además, como se ve en Mitchell & Wait [pag 52], es posible aplicar el Método de Galerkin a ciertos problemas no-lineales.

CAPITULO 4 El método de elementos finitos

En el método de elementos finitos obtenemos la solución aproximada interpolando la función en la frontera al interior del elemento. Por ejemplo para $[x_1, x_2]$ dividido en subintervalos $x_1 \leq x \leq x_{i+1}$, se puede obtener la solución aproximada U a partir de valores nodales mediante una interpolación lineal

$$U = U(x) = U_1 \gamma_1 + U_2 \gamma_2$$

donde se llama a γ_1 y γ_2 funciones de interpolación o funciones base. U_1 , U_2 son parámetros desconocidos que se determinan a partir del valor estacionario de funcionales variacionales (como se vió anteriormente). En general se tiene $U(x) = \sum_{i=1}^n U_i \gamma_i(x)$.

En el método de Elemento finito generalmente se utilizan como las U_i los valores de U , i.e. $U_i = U(x_i)$, (la solución al problema variacional) en un punto discreto del dominio (por ejemplo los nodos); se escogen las γ_i de manera que desaparezcan en todas partes excepto en una pequeña región del continuo, usualmente unos cuantos elementos, para hacer que su influencia sea local y por tanto tengamos cálculos más económicos. Casi siempre se escogen funciones de interpolación (llamadas también funciones base) tal que dentro de un elemento la función de interpolación dependa únicamente de las coordenadas generalizadas con los nodos del elemento.

En otras palabras se escogen funciones tales que sean suaves dentro del elemento y puedan ser construidas independientemente del resto de las funciones del elemento, pero exigiendo compatibilidad en las fronteras adyacentes, es decir, las funciones de interpolación deben tener cierto grado de diferenciabilidad, al menos en las fronteras entre los elementos.

En general se escogen polinomios como funciones de interpolación pues son fácilmente manejables y porque localmente toda función suave parece un polinomio. Para seleccionar el orden y términos de los polinomios es necesario considerar la completéz de la representación polinomial y la compatibilidad entre elementos adyacentes.

sección 4.1 Funciones base

Al utilizar el método de elementos finitos el paso más esencial es escoger el elemento. Se deben aproximar las variables del problema por un conjunto de funciones en términos de las coordenadas generalizadas, es decir aproximar una función $u(x)$ dentro de un elemento mediante la interpolación ya mencionada. Las funciones de interpolación pueden ser de diversas clases, como la de Hermite que aproxima la función $u(x)$ mediante un polinomio $V(x)$ en términos de la propia $u(x)$ y sus derivadas. Un caso especial es la que utiliza solamente $u(x)$ es decir

$$V(x) = \sum_{i=1}^n u(x_i) \psi_i$$

que es la llamada fórmula de interpolación de Lagrange.

Las funciones de interpolación de Lagrange están dadas por

$$\psi_i(x) = \frac{L(x)}{x - x_i}$$

donde $L(x) = (x - x_1)(x - x_2) \dots (x - x_n)$ que satisface $\psi_i(x_k) = \delta_{ik}$ para $1 \leq i, k \leq n$.

Se tiene entonces un polinomio de grado $n-1$. En el caso $h=2$ (unidimensional) resultan funciones lineales

$$\psi_1(x) = \frac{1-x}{2}$$

$$\psi_2(x) = \frac{1+x}{2}$$

si se toman $x_1 = -1$, $x_2 = 1$.

En el caso $n=3$ con $x_1 = -1$, $x_2 = 0$, $x_3 = 1$ resultan los polinomios cuadráticos

$$\gamma_1(x) = -x(1-x)/2$$

$$\gamma_2(x) = (1+x)(1-x)$$

$$\gamma_3(x) = (1+x)(x)/2$$

que es lo que se utiliza en el presente trabajo en el caso unidimensional.

Similarmente en el caso bidimensional, donde usualmente se escoge entre elementos rectangulares y triangulares. Aquí se utilizan elementos triangulares cuadráticos organizados como muestra la figura 3(b). En el presente trabajo se utiliza interpolación cuadrática en elementos triangulares.

Sea el plano (β_1, β_2) ; se tiene un triángulo rectángulo standard formado por $\beta_1 = 0$, $\beta_2 = 0$ y $\beta_1 + \beta_2 = 1$, es decir el triángulo con vértices en

$$p_1 = (1, 0)$$

$$p_2 = (0, 1)$$

$$p_3 = (0, 0)$$

Si se introduce una nueva coordenada $\beta_3 = 1 - \beta_1 - \beta_2$, se tiene un nuevo sistema coordenado local que en este caso coincide con las llamadas coordenadas triangulares o de arco (ver Strang y Fix [pag. 94] y figura 4a).

Se utiliza como función de interpolación:

$$v(\beta_1, \beta_2, \beta_3) = \sum_{i=1}^3 \gamma_i(\beta_1, \beta_2, \beta_3) U_i = \bar{p}^T \bar{U}$$

donde $\bar{p} = (\gamma_i) = \begin{pmatrix} \beta_1 (2\beta_1 - 1) \\ \beta_2 (2\beta_2 - 1) \\ \beta_3 (2\beta_3 - 1) \\ 4\beta_2\beta_3 \\ 4\beta_3\beta_1 \\ 4\beta_1\beta_2 \end{pmatrix}$

son las funciones base. A estos triángulos (con estas funciones base) les llamaremos elementos regulares o standard en el caso bidimensional.

El trabajo comenzó a partir del programa para el caso mas simple (PROGRAMA/ORIGINAL del Dr. J. P. Hennart) mismo que se describe a continuación y esta contenido en los apendices (aclaro que se incluye ademas de cada programa, un indice propio de las rutinas de este, asi que al referirme a la rutina 'x' del programa 'y' basta consultar el indice del programa 'y').

Se trata de un programa para elementos finitos cuadráticos en una dimensión que resuelve el problema lineal

$$-\frac{\partial}{\partial x} \left(p \frac{du}{\partial x} \right) + q u = f$$

donde $p(x) > 0$, $q(x) > 0$, $f(x)$; $\frac{du}{\partial x} = 0$ $x \in \partial \Omega_2$ y con $u = 0$ $x \in \partial \Omega_1$ (en la frontera), sobre el intervalo $[x_1, x_2]$ utilizando cuadratura Gaussiana con cuatro puntos para calcular los elementos de la matriz de rigidez \bar{A} y del vector de fuerza \bar{F} .

Se hace lo siguiente:

1) Se lee el numero de subintervalos a considerar (NI) y las condiciones de frontera (ILBC, IRBC).

2) Se llama a la subrutina STURM que es la que realmente controla el proceso de solución. Lo que sigue es lo que va haciendo STURM.

i) Se llama a STIFF que se encarga de calcular la matriz de rigidez \bar{A} y el vector de fuerza \bar{F} ; es la parte medular del método de elemento finito.

ii) Mediante la llamada factorización de Cholesky, contenida en las rutinas: CHOBNT, (la factorización de Cholesky), y CHOBNS, (la resolución de la ecuación matricial) se obtiene lo que es a la vez solución de $\bar{A}\bar{U} = \bar{F}$ y aproximación discreta a la solución de la ecuación diferencial:

Se utiliza esta factorización ya que debido a las funciones base elegidas se tiene que \bar{A} es una matriz simétrica positiva definida.

- iii) Se compara con la solución analítica esperada. Para ello se utiliza la rutina ERROR.

seccion 5.1 Construcción del sistema de ecuaciones lineales

La rutina STIFF calcula los elementos de la matriz de rigidez \bar{A} y el vector \bar{b} como se describe a continuación:

Los elementos de la matriz $\bar{A} = (A_{IJ})$ son simplemente el resultado de, sumando sobre los M elementos del dominio, hacer:

$$A_{IJ} = \sum_{k=1}^M \left(\int_{x_1}^{x_2} q \psi_I \psi_J dx + \int_{x_1}^{x_2} p \frac{\partial \psi_I}{\partial x} \frac{\partial \psi_J}{\partial x} dx \right)$$

y el vector $\bar{b} = (b_I)$ se calcula como sigue:

$$b_I = \sum_{k=1}^M \left(\int_{x_1}^{x_2} f \psi_I dx \right)$$

en donde x_1, x_2 son respectivamente el principio y el final de el subintervalo considerado (que en este caso corresponde al elemento correcto). Las ψ_I son funciones base apropiadas para el elemento en cuestión. STIFF hace justamente estos calculos tomando en cuenta que debido a las funciones base elegidas se tiene que $\bar{A} = (A_{IJ})$ es una matriz simétrica pentadiagonal la cual se almacena en forma comprimida. Esto permite ahorrar bastante memoria.

seccion 5.2 Integración

Al evaluar las matrices resultantes en la sección anterior debemos calcular muchas integrales, que pueden ser difíciles o imposibles de integrar analíticamente. Por ello se utiliza comúnmente la integración numérica. Si se aproxima $u(x)$ utilizando

$$u(x) = \sum_{i=1}^n U_i \psi_i$$

entonces

$$\int u(x) dx$$

se puede aproximar como

$$\sum_{k=1}^n (w_k u(x_k) + E_k)$$

donde

$$w_k = \int \psi_k(x) dx$$

es un factor de peso y

$$E_k = \int (u(x_k) - U(x_k))^2 dx$$

es el error. Los valores escogidos para x_k pueden o no tener algo que ver con los nodos utilizados en la interpolación, pero en el caso de la cuadratura Gaussiana se utilizan puntos de integración dados en una tabla, y los pesos asociados a estos (ver [7] pag. 187).

En el programa unidimensional se utiliza para integrar la rutina FINT, que es la cuadratura gaussiana de cuatro puntos, i.e.:

$$\int_a^{x_2} f(x) \bar{Y}_i \bar{Y}_j dx = \sum_{k=1}^4 w_k f^k \bar{Y}_i^k \bar{Y}_j^k$$

donde los w_k son los pesos adecuados (ver [7] [pag. 187-189]) para esta forma de integrar, y $f^k = f(x_k)$, $\Phi_i^k = \Phi_i(x_k)$ y $\Phi_j^k = \Phi_j(x_k)$, si los x_k son los puntos de integración correspondientes a los w_k .

Las Φ_i son simplemente las funciones base η_i o sus derivadas $\partial \eta_i / \partial x$ segun el valor asignado a ID (que es argumento de FINT). Si este es 0 se tendran las funciones base, y si es 1, sus derivadas.

En esta ecuación f significa la función adecuada en cada caso (i.e. f, \dot{f}, \ddot{f} de la ecuacion diferencial).

seccion 5.3 Factorización

Se han obtenido \bar{A} , \bar{P} para factorizar \bar{A} se utilizó la factorización de Cholesky tomando en cuenta que debido a que $\rho > 0$, $q \gg 0$, se tiene una matriz simétrica positiva definida (que es pentadiagonal debido a la base escogida en el caso unidimensional).

El conocer esta descomposición nos permite resolver sistemas de la forma $\bar{A}\bar{U} = \bar{P}$ resolviendo primero $\bar{R}^T \bar{W} = \bar{P}$ y después $\bar{R} \bar{U} = \bar{W}$. Estos sistemas son los que resuelve la rutina CHOBNS, de la manera esperada al ser triangular inferior y superior respectivamente, como se podrá apreciar en el listado de PROGRAMA/ORIGINAL, incluido en los apendices.

CAPITULO 6 Modificaciones al metodo de elementos finitos

Para pasar al caso mas general considerado en esta tesis, se divide el trabajo en dos etapas. Una primera que consiste en resolver a partir del programa original ya descrito, con las modificaciones apropiadas, el caso general en una dimension. La segunda consiste en extender al caso bidimensional los resultados obtenidos en la primera etapa. La primera etapa se describe a continuacion. Para mas detalles consultar el listado del programa modificado final para el caso unidimensional, PROGRAMA/1D/1, contenido en los apendices.

seccion 6.1 Transformación Isoparamétrica

La mayoría de las modificaciones requeridas hacen necesario trabajar con mallas un poco mas generales que las presentes en el programa original. Por tanto, utilizar elementos que puedan transformarse a elementos standard nos da la flexibilidad deseada (ver [51]) y facilita el trabajo de adaptar los métodos para tratar singularidades.

Se utilizan elementos standard, y una transformación que nos lleve a estos a partir de los elementos en el dominio en el que está planteado el problema; en el caso unidimensional, esta consiste en transformar nuestros intervalos a un intervalo standard con el tercer punto en el centro.

La transformación que lleva un intervalo (en general elemento) en el espacio de las x a el espacio de las T donde $T \in [-1, 1]$ se define por medio de

$$x = \sum_{i=1}^n x_i \gamma_i(T)$$

donde x_i son valores nodales para $i=1, \dots, n$ y n es el número de funciones base.

Si se define una transformación

$$V(x) = \sum_{i=1}^n u_i w_i(T)$$

donde U_i son valores nodales correspondientes a los puntos x_i , y se tienen ciertas funciones $w_i(T)$, la coordenada local T se define en terminos de x mediante la inversa de la transformación.

La necesidad de que la inversa de la transformación esté bien definida impone varias condiciones sobre la disposición de estos puntos. Si para aproximar el valor de x se utilizan las funciones base ψ_i (en el lugar de las w_i) se tiene la llamada transformación isoparamétrica.

En el caso cuadrático unidimensional, que es el que nos interesa, dado T se calculan las funciones base $\psi_1(T)$, $\psi_2(T)$, $\psi_3(T)$; entonces se obtiene

$$x(T) = x_1 \psi_1(T) + x_2 \psi_2(T) + x_3 \psi_3(T)$$

donde x_1, x_2, x_3 son valores nodales. Dado que también se tienen los valores de u en los tres puntos, es decir U_1, U_2, U_3 (inicialmente basta alguna primera aproximación) se calcula $V(T)$ como:

$$V(T) = U_1 \psi_1(T) + U_2 \psi_2(T) + U_3 \psi_3(T)$$

Para esto se utiliza el procedimiento CALC, contenida en PROGRAMA/ID/1, al que le pasamos T además de x_1 (el punto inicial del subintervalo), x_2 (el punto intermedio), y x_3 (el punto final). Notar que la reenumeración dentro del intervalo hace que x_3 corresponda a lo que era x_2 . Se utiliza a la rutina UIJ que determina el valor de la función base adecuada.

Al integrar, donde se tenía $\int_{x_1}^{x_2} f \Psi_i \Psi_j dx$

con $f(x)$, $\tilde{F}_1(x)$ y $\tilde{F}_0(x)$ se tiene ahora

$$\int_{-1}^1 \tilde{f}(T) \tilde{F}_1(T) \tilde{F}_0(T) \frac{\partial x}{\partial T} dT$$

donde $x(T)$ y $T \in [-1, 1]$ como era de esperarse. Para los cálculos la única diferencia es que en lugar de hacer los cálculos en base a x se harán a partir de T y algún indicador del intervalo en que se está, y así obtener $x = x(T)$.

En la rutina de ERROR es necesario obtener T dado x para estimar el error en una malla uniforme (esta malla no tiene nada que ver con la usada a la hora de los cálculos). Por razones históricas se optó por calcular en forma explícita la inversa de la transformación, suponiendo que para cada método se sabía exactamente cual era la transformación de CALC en forma analítica i.e. usualmente lineal, y cuadrática en el caso de elementos con punto cuarto.

Dado que esto no funciona en el caso de funciones base especiales que se estudia mas adelante, (para las que sin embargo se pudo proceder en forma semejante a la del problema original) se incluyeron en el programa de una dimension dos rutinas diferentes para el cálculo del error. En el caso bidimensional, como se vera mas adelante, se decidió, dada la mayor complejidad de la transformación, aproximar la inversa usando un método iterativo de Newton.

seccion 6.2 Ecuaciones no-lineales

En un principio se manejaron tan solo problemas lineales, es decir, donde dada la ecuación

$$-\frac{d}{dx} \left(p \frac{du}{dx} \right) + qu = f$$

se tenía que $p(x)$, $q(x)$, $v(x)$. Dado que hemos transformado el programa de manera que conociendo el intervalo y T se obtienen $x(T)$ y $V(T)$ fácilmente (dados valores aproximados para u mediante CALC) en el caso $p = p(T, x(T), u(T))$, $q = q(T, x(T), u(T))$, $f = f(T, x(T), u(T))$; basta llamar a CALC en el lugar apropiado y utilizar $x(T)$ y $V(T)$ tal como este los calcula, resolviendo como para un problema lineal, pero iterando hasta converger (con una precisión dada) a la solución del problema no-lineal, (Ver [5] pag 111).

Este es el llamado método de sustitución sucesiva, es decir evaluar el coeficiente no-lineal a la n -ésima aproximación U_n y determinar U_{n+1} como la solución de un problema lineal.

Este método converge a la solución correcta, pero hay que tener cuidado y primero establecer la ecuación no-lineal para la función minimizante V y después resolver el problema iterando; en otro caso se puede converger de manera incorrecta.

seccion 6.3 Criterio para detener la iteración.

La rutina CHECK es la encargada de detener las iteraciones; sea k el índice usado para la iteración; lo que hace es simplemente comparar la nueva solución $\bar{U}^{(k+1)}$ al sistema

$$\bar{A}^{(k)} \bar{U}^{(k+1)} = \bar{F}^{(k)}$$

con la solución a

$$\bar{A}^{(k-1)} \bar{U}^{(k)} = \bar{F}^{(k-1)}$$

Para el caso unidimensional esta comparación se hace de la siguiente manera:

Digamos que

$$RMAX = \max_I |U(x_I)^{(k+1)} - U(x_I)^{(k)}|$$

$$SMA S = \max_I |U(x'_I)^{(k+1)}|$$

$$EPS = \epsilon_{abs} + \epsilon_{rel} (SMA X)$$

donde las x_I son los nodos de la malla usada para el cálculo, N el número de nodos (llamado NX en el programa) y ϵ_{abs} , ϵ_{rel} son dos números fijados externamente.

El criterio para detener la iteración del método de elementos finitos usado es simplemente que

$$RMAX < EPS$$

Además hay un contador de iteración ITER que permite detener artificialmente el proceso si el número de iteraciones es mayor o igual a LIMIT que es también un número fijado externamente.

Se utilizan $\epsilon_{rel} = .001$, $\epsilon_{abs} = .0000001$ y LIMIT=7.

En el caso general que nos interesa no es posible asegurar que la matriz de rigidez resultante sea positiva-definida, (el operador puede no serlo).

Abandonamos el esquema de factorización de Cholesky por uno más general de factorización triple, (ver [7] pag. 85, adaptado para matrices bandadas simétricas. Esto tiene un costo en eficiencia en aquellos casos en que si hubiéramos obtenido matrices positivas-definidas (en el caso bidimensional resultó tan ineficiente que probablemente hubiera sido mejor utilizar algún método iterativo).

Consiste en factorizar \bar{A} como $\bar{A} = \bar{L} \bar{D} \bar{L}^T$ donde \bar{L} es triangular inferior y \bar{D} es diagonal; resolvemos $\bar{A} \bar{U} = \bar{B}$ resolviendo primero $\bar{L} \bar{y} = \bar{B}$ y después $\bar{L}^T \bar{U} = \bar{y}$.

sección 6.5 · Integración numérica

En el programa original se utiliza para integrar la aproximación de Gauss de cuatro puntos (ver [7] [pag 187-188.1]). Lo mismo se hará en el caso general, pero es necesario tomar en cuenta los efectos de la transformación isoparamétrica en la integral.

Se vio ya que en la integral analítica donde en un principio se tenía

$$\int_{x_1}^{x_2} f \Psi_i \Psi_j dx$$

con $f(x)$, $\Psi_i(x)$ ahora se tiene

$$\int_{-1}^1 \tilde{f} \tilde{\Psi}_i \tilde{\Psi}_j \frac{dx}{dT} dT$$

con $x = x(T)$.

En la aproximación se hace exactamente lo mismo, lo que implica tener un valor para $\partial x / \partial T$. Dado que se utilizan las funciones base para calcular x en función de T mediante la rutina CALC, es también lógico pensar en calcular $\partial x / \partial T$ de manera similar, pero utilizando las derivadas de las funciones base en vez de las funciones base:

$$\frac{\partial x}{\partial T} = \sum_{i=1}^n x_i(T) \frac{\partial \tilde{\psi}_i(T)}{\partial T}$$

Recordemos que se desea utilizar FINT para evaluar cualquiera de las expresiones siguientes:

$$\int_{x_1}^{x_2} f \psi_i \psi_j dx \quad \int_{x_1}^{x_2} f \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} dx \quad \int_{x_1}^{x_2} f \psi_i dx$$

Durante la aproximación resulta que (por la regla de la cadena) en el primero y tercer casos es necesario multiplicar cada sumando (i.e. el valor del integrando en cada uno de los puntos de Gauss) por el valor de la aproximación a la derivada de x con respecto a T , es decir:

$$\int_{x_1}^{x_2} f \tilde{\psi}_i \tilde{\psi}_j \frac{\partial x}{\partial T} dx$$

mientras que en el segundo, aparece aparte el inverso al cuadrado de esta, y por tanto hay que dividir entre este valor, o sea que, si denotamos $\partial \tilde{\psi}_i / \partial T$ por $\tilde{\psi}'_i$ entonces

$$\int_{x_1}^{x_2} f \frac{\tilde{\psi}'_i \tilde{\psi}'_j}{(x')^2} x' dx = \int_{x_1}^{x_2} f \frac{\tilde{\psi}'_i \tilde{\psi}'_j}{x'} dx$$

Todo esto lo trata con FINT, utilizando el indicador ID para saber de cual integral se trata, de manera consistente a como se hace en PROGRAMA/ORIGINAL. Si al valor calculado para la derivada se le llama \mathcal{J} entonces basta multiplicar por $\mathcal{J}^{(1-2*ID)}$ cada término de la suma en FINT. En el programa (ver los apéndices) a \mathcal{J} se le llama RJAC, y se tiene que $RMULT=RJAC*(1-2*ID)$.

seccion 6.6 Problema parabolico

Hasta ahora el programa resuelve problemas elípticos, es decir problemas de la forma

$$\frac{d}{dx} \left(p \frac{du}{dx} \right) - q u + f = 0$$

Para ecuaciones de la forma

$$\frac{du}{dt} = \frac{d}{dx} \left(p \frac{du}{dx} \right) - q u + f$$

donde $u = u(x, t)$, es necesario resolver un sistema de ecuaciones un poco distinto al arriba mencionado. Si los U_I son valores nodales de la solución, entonces, por Galerkin se tiene:

$$\begin{aligned} M_{IJ} \frac{dU_I}{dt} &= -U_I \left(\int p \frac{\partial \psi_I}{\partial x} \frac{\partial \psi_J}{\partial x} dx + \int q \psi_I \psi_J dx \right) + \int f \psi_I dx \\ &= -A_{IJ} U_I + B_I \end{aligned}$$

donde

$$\begin{aligned} A_{IJ} &= \int p \frac{\partial \psi_I}{\partial x} \frac{\partial \psi_J}{\partial x} dx + \int q \psi_I \psi_J dx \\ B_I &= \int f \psi_I dx \\ M_{IJ} &= \int \psi_I \psi_J dx \end{aligned}$$

Aproximamos $\frac{d\bar{u}}{dt}$ como $\frac{\bar{u}(t+1) - \bar{u}(t)}{\Delta t}$ donde $\bar{u}(t)$ significa \bar{u} evaluado en el tiempo t . El lado derecho de la ecuación diferencial lo calculo como un promedio ponderado con peso θ en el tiempo $t+1$, y un peso $1-\theta$ en el tiempo t , es decir el llamado método θ ; entonces obtenemos:

$$\frac{1}{\Delta t} \bar{M} (\bar{u}(t+1) - \bar{u}(t)) = (1-\theta) (\bar{P}(t) - \bar{A}(t) \bar{u}(t)) + \theta (\bar{P}(t+1) - \bar{A}(t+1) \bar{u}(t+1))$$

lo que implica que el sistema a resolver es

$$\bar{M} + \theta \Delta t \bar{A}(t+1) \bar{u}(t+1) = \bar{M} - (1-\theta) \Delta t \bar{A}(t) \bar{u}(t) + \Delta t (1-\theta) \bar{P}(t) + \Delta t \theta \bar{P}(t+1)$$

o sea que

$$\bar{M} + \theta \Delta t \bar{A}(t+1) \bar{u}(t+1) = \bar{u}_{aux} + \Delta t \theta \bar{P}(t+1)$$

si hacemos

$$\bar{u}_{aux} = \bar{M} - (1-\theta) \Delta t \bar{A}(t) \bar{u}(t) + \Delta t (1-\theta) \bar{P}(t)$$

Este esquema es incondicionalmente estable solo si $\theta \in [\frac{1}{2}, 1]$ que es un caso implícito. En el programa se utiliza $\theta = \frac{2}{3}$.

En el programa esta nueva situación se ve reflejada en STARM.

Aproximamos $\frac{d\bar{U}}{dt}$ como $\frac{\bar{U}(t+1) - \bar{U}(t)}{\Delta t}$ donde $\bar{U}(t)$ significa \bar{U} evaluado en el tiempo t . El lado derecho de la ecuación diferencial lo calculo como un promedio ponderado con peso θ en el tiempo $t+1$, y un peso $1-\theta$ en el tiempo t , es decir el llamado método θ ; entonces obtenemos:

$$\frac{1}{\Delta t} \bar{M} (\bar{U}(t+1) - \bar{U}(t)) = (1-\theta) (\bar{B}(t) - \bar{A}(t) \bar{U}(t)) + \theta (\bar{B}(t+1) - \bar{A}(t+1) \bar{U}(t+1))$$

lo que implica que el sistema a resolver es

$$\bar{M} + \theta \Delta t \bar{A}(t+1) \bar{U}(t+1) = \bar{M} - (1-\theta) \Delta t \bar{A}(t) \bar{U}(t) + \Delta t (1-\theta) \bar{B}(t) + \Delta t \theta \bar{B}(t+1)$$

o sea que

$$\bar{M} + \theta \Delta t \bar{A}(t+1) \bar{U}(t+1) = \bar{U}_{aux} + \Delta t \theta \bar{B}(t+1)$$

si hacemos

$$\bar{U}_{aux} = \bar{M} - (1-\theta) \Delta t \bar{A}(t) \bar{U}(t) + \Delta t (1-\theta) \bar{B}(t)$$

Este esquema es incondicionalmente estable solo si $\theta \in [\frac{1}{2}, 1]$ que es un caso implícito. En el programa se utiliza $\theta = \frac{2}{3}$.

En el programa esta nueva situación se ve reflejada en STARM.

En esta seccion se analizan las propiedades de convergencia del método de elemento finito. En una dimension (ver Strang y Fix pag. 62) si M es el numero de elementos (intervalos) y $h = (x_f - x_r) / M$ resulta que si u es la solucion analítica y U la aproximación encontrada por el metodo de elementos finitos, entonces, si la norma utilizada es una norma en un espacio de Sobolev (ya vimos que es un espacio de Sobolev) donde si

$$u^{(s)} = \frac{\partial^s u}{\partial x^s}$$

entonces:

$$\|u\|_s^2 = \int_{x_r}^{x_f} (\sum (u^{(s)})^2) dx$$

El caso $\|u\|_0$ es el de la norma usual RMS (root mean square) o L2.

Tenemos entonces que:

$$\|u - U\|_0 \leq C h^{k-1} \|u^{(k)}\|_0$$

donde la base esta formada por polinomios de orden $k-1$ y C es una constante. Esta constante es una indicación directa de las propiedades del elemento particular escogido; depende de la construccion del elemento y de los parametros nodales. Para geometrias regulares un buen estimado asintótico para C es el error para aproximar polinomios del grado inmediatamente superior k . La potencia a la que se encuentra elevada h es simplemente función del grado de los polinomios utilizados. es facil de encontrar numérica-

mente, indicando la razón de convergencia al refinar la malla. El tercer factor, es decir el valor de la norma de $V(k)$ depende del problema en sí, es decir, de la suavidad de la solución, es decir de que tan fácil es aproximarla por polinomios.

En general (ver [5] [pag. 166]) se obtiene el siguiente resultado:

Si el espacio S^h de elementos finitos es de orden $k-1$ (es decir si $P_{k-1} \subset P$ la restricción de S^h a un elemento) y la ecuación es elíptica, se tiene que para la s derivada, en caso de que esta exista,

$$\|u - U\|_s \leq \begin{cases} C h^{k-s} \|u\|_k & 2m - k \leq s \\ C h^{2(k-m)} \|u\|_k & s \leq 2m - k \end{cases}$$

donde $2m$ es el orden problema diferencial. La convergencia se da solo si $k > m$, es decir si se pueden reproducir exactamente soluciones que son polinomios de grado m .

La convergencia en lo anterior es esencialmente la convergencia de la m derivada de U . Para la s derivada no se puede obtener una convergencia mejor que $O(h^{k-s})$, que es el orden de la mejor aproximación a u en un espacio S^h cuyos polinomios son de grado $k-1$ que es el caso de la aproximación de Ritz.

En el caso cuadrático unidimensional estudiado resulta un error de discretización de orden $O(h^2)$. Las propiedades de los polinomios usados en triángulos en dos dimensiones se resumen en Strang y Fix (pag. 84) de donde resulta que en el caso del polinomio cuadrático en que $k=3$ se tiene también un error de discretización de orden $O(h^2)$. La razón de convergencia en puntos individuales es el mismo siempre y cuando tenga derivadas en todo punto (cuando hay singularidades esto ya no sucede y por tanto las razones de convergencia en el sentido RMS y puntuales pueden ser muy distintas). Por otra parte en ciertos puntos especiales el error puede converger con mayor velocidad que al

promedio. Entonces si para calcular los errores se utiliza una malla que coincida en ciertos puntos con los nodos de la malla de elementos finitos, es posible obtener una mayor convergencia debido a la superconvergencia que se produce en estos puntos (ver [5] pag. 168).

Todos estos resultados son válidos para problemas elípticos lineales con condiciones adecuadas de frontera, donde además:

- 1) No hay singularidades.
- 2) No se utilizan mallas irregulares.
- 3) Son resultados acerca de convergencia al refinar uniformemente la malla.

Además se ha supuesto que se calcula la aproximación de manera exacta. Como de hecho se utiliza integración numérica y hay errores de truncamiento, sobretodo en la resolución del sistema lineal, hay que cuidar los detalles. En cuanto a la integración basta que las $2(k-m)$ derivadas de las funciones base se integren exactamente para que no haya problemas. Curiosamente hay ocasiones en las cuales una integración numérica inexacta compensa el "stiffness" del sistema y de hecho mejora la aproximación.

Sin entrar en detalles, Strang (pag. 107) dice que siempre y cuando se tenga una transformación isoparamétrica no singular $n \times n$ tiene problemas de convergencia. La transformación de punto cuarto no es uniformemente suave y por tanto debe disminuir la razón de convergencia.

Para problemas no-lineales monotonos tampoco encuentra problemas para extender los resultados anteriores, (ver [5] pag 111). En cuanto al tiempo, hemos supuesto una aproximación de Galerkin computada continuamente en el tiempo. Para problemas de valor inicial se encuentra una separación entre el error debido al método utilizado en el tiempo y el debido a la aproximación. En el caso parabólico con el orden óptimo sigue siendo ya que el operador sigue siendo el mismo operador elíptico que ocurre en problemas estáticos.

Para un estudio de la convergencia en problemas parabólicos (el caso lineal) tratados por métodos que avanzan un paso en el tiempo a la vez, (que es el método escogido), se puede consultar Strong y Fix, pag 250, y Mitchell & Wait [pag 163-165] quienes llegan al siguiente resultado que es consistente con lo esperado:

Si S^h es un espacio de elementos finitos de orden $k-1$ el error en la aproximación de Galerkin para un problema parabólico satisface (utilizando la norma vista):

$$\|u(t) - v(t)\|_0 \leq Ch^k \left[\|u(t)\|_k + e^{-\lambda_1 t} \|u_0\|_k + \int_0^t e^{-\lambda_1(z-t)} \|u_t(z)\|_k dz \right]$$

Donde λ_1 es el tiempo característico del modo fundamental; el error es del mismo orden $O(h^k)$ que el encontrado para los problemas estacionarios, y decae tan rápido como el modo fundamental si no hay un término fuente.

CAPITULO 7 Singularidades

El efecto de las singularidades en la convergencia del método es que esta es mas lenta, especialmente cerca de la singularidad (ver [8] pag 39-43). Por tanto mediante el método de elementos finitos con elementos regulares se obtienen aproximaciones que son inexactas en esa zona. Existen varias adaptaciones al método standard que superan esta dificultad (ver [8] [pag 44-49]). Estas pueden dividirse en dos clases:

1) Métodos que explotan la forma de la singularidad, y por tanto requieren un conocimiento de esta, y ya vimos que en nuestro caso conocemos la forma de la singularidad.

2) Métodos que aunque toman en cuenta la presencia de la singularidad ignoran su forma. Estos pueden funcionar aun sin conocer la forma de la singularidad, pero si esta es conocida, pueden mejorarse la aproximación al tomarla en cuenta.

seccion 7.1 Funciones base especiales

Entre la primera clase de métodos esta el de aumentar el espacio de funciones base con funciones que tengan la forma de la singularidad, que es un método práctico obvio para enfrentar la singularidad (ver [9]). Este consiste en definir las funciones base nuevas de manera que tengan la forma apropiada (singular) cerca de la singularidad, sean cero excepto en la vecindad de la singularidad, y tengan una continuidad adecuada globalmente. Strang (pag. 264-268) justifica el método y demuestra que al incluir funciones singulares es posible recuperar la razón de convergencia de problemas suaves.

Sea r la distancia de x a la singularidad. Si se desea representar mediante funciones base algunas funciones de la forma \sqrt{r} que son las que resultan en nuestro caso, las funciones base cuadráticas no son la mejor manera de lograrlo. Lo que se hace es incluir en la base funciones con la forma deseada \sqrt{r} .

La base usada (para un elemento normalizado a $x \in [-1, 1]$) con una singularidad en $x = -1$ es:

$$\psi_1 = x + 1 - (1 + \sqrt{2}) \sqrt{x + 1}$$

$$\psi_2 = -(1 + \sqrt{2})(x + 1 - \sqrt{2} \sqrt{x + 1})$$

$$\psi_3 = (1 + \sqrt{2})(\sqrt{x + 1})/\sqrt{2}$$

Estas funciones tienen la forma \sqrt{r} deseada. Se utilizó esta base únicamente donde efectivamente se tiene este comportamiento, es decir cerca de la singularidad. En el resto del espacio se utilizó la base cuadrática. Hay varias maneras de incluir las funciones especiales; o se aumentan a la base cuadrática, que es lo que se hace usualmente, o bien se sustituyen en vez de y^2 únicamente cerca de la singularidad, donde son necesarias. Este último fue el camino seguido, a pesar de que aunque se mejora la aproximación no se recupera el orden de convergencia del caso no singular (dice J. P. Hennart), ya que así se evita que se compliquen demasiado los cálculos, al no permitir que la matriz \bar{A} crezca o pierda su estructura bandeda.

Por otra parte, la rutina de Gauss para aproximar integrales es buena para polinomios, (en este caso de grado dos). Sin embargo para integrar funciones de la forma \sqrt{r} es francamente mala, y se puede perder más de lo ganado si utilizamos simplistamente esta opción. En realidad requerimos integrar de otra manera.

Resulta fácil ver por la forma de la base, que lo único que se necesita hacer es un cambio de variable, utilizando \sqrt{r} en vez de r . Así se podrán integrar exactamente polinomios cuadráticos en \sqrt{r} que es lo que se necesita.

Para las funciones base especiales es necesario modificar FIAT pues en vez de integrar polinomios en x es necesario hacerlo en $\sqrt{x-x_1}$, $\sqrt{x-x_2}$. Esto es un simple cambio de variable e implica cambios en la forma en que deben usarse RMULT y T.

En este caso se debe tratar el primer y último intervalos de manera distinta a los demás, para lo que se utiliza IC como indicadores. Para el primer intervalo se usa IC=2, para el último utilizando IC=3, y para los demás, IC=1.

Lo que debemos hacer es simplemente hacer el cambio de variable apropiado según el valor de IC y ajustar el determinante jacobiano. La rutina puede consultarse en los apendices (ver PROGRAMA/1P/1).

seccion 7.2 Elementos con punto cuarlo

Otra posibilidad es utilizar ciertas propiedades de la transformación isoparamétrica. Si $\bar{x}_i = \frac{\partial \bar{x}}{\partial x}$ y $\bar{x}_T = \partial \bar{x} / \partial \tau$ resulta que $\bar{x}_i = \bar{J}^{-1} \bar{J}_T$ donde \bar{J} es la matriz jacobiana de la transformación $x = x(\tau)$. En problemas con solución singular de la forma γ^p con $0 < p \leq 1$, que son los que nos interesan, resulta que las derivadas se van a infinito, y no es posible aproximar este comportamiento mediante las funciones base polinomiales usuales.

Una manera de incluir este comportamiento es a través de la matriz jacobiana ya que se puede causar que el determinante jacobiano \bar{J} desaparezca en ciertos nodos mediante una manipulación adecuada de las posiciones de los nodos, formando lo que se llaman elementos singulares.

Si en un elemento cuadrático ponemos el nodo intermedio en el punto cuarto en vez de a la mitad del lado del elemento (hacia el lado en que esta la singularidad) la transformación hacia el elemento cuadrático standard nos produce un jacobiano singular de la forma $\sqrt{\tau}$ en el nodo en que esta la singularidad. Se ha visto, (por ejemplo [4]), que en problemas elípticos con singularidades de tipo $\sqrt{\tau}$ se recupera de esta manera la razón de convergencia de una solución suave.

En general si tengo la transformación (ver [4]):

$$x = \sum_{i=1}^n x_i l_i$$

donde para $i, k = 1, \dots, n$ con

$$l_i(k/n) = \begin{cases} 1 & i=k \\ 0 & i \neq k \end{cases}$$

Entonces tengo una interpolación de Lagrange que interpola valores de x_i en los puntos k/n del intervalo $[0, 1]$ y si tomo los valores de x_i como

$$x_0 + (x_n - x_0) l_i(k/n)^n$$

se sigue que

$$x - x_0 = (x_n - x_0) l_i(k/n)^n$$

Si $g = 0$ una función lineal p puede representar una función de la forma $\tau \sqrt{\tau}$ si los puntos intermedios se seleccionan de la manera adecuada. El caso $n = 2$ es el del método de puntos cuartos.

La técnica consiste en rodear la singularidad de una capa de estos elementos singulares, con los puntos cuartos orientados hacia la singularidad. El otro lado de estos elementos es compatible con los elementos usuales que se utilizan para el resto del espacio.

En el caso de una dimensión, en que los elementos son simplemente intervalos (ver [6]), si el nodo interno se pone en el punto cuarto (o tres cuartos) del intervalo, las funciones base pueden representar funciones de la forma \sqrt{r} (donde r es la distancia al punto donde esta la singularidad), alrededor de la singularidad. Esto queda mas claro si se consulta la figura número 1.

Por otra parte es necesario hacer notar que en alguna forma el método de elementos con puntos cuartos y el de funciones base especiales hacen lo mismo; adaptar el método standard haciendo una transformación de un espacio a otro. La diferencia es básicamente en donde se hace esto. Los puntos cuartos transforman la malla en si, las funciones especiales la base. Inclusive el espacio de funciones que resulta es el mismo (la base es distinta). Es claro que en los elementos de punto cuarto se tienen solamente funciones formadas por $1, r, \sqrt{r}$ y no r^2 mientras que en los demás elementos solo se tiene $1, r, r^2$ lo cual es claramente lo mismo que tentamos en funciones singulares, es decir no hemos aumentado la base incluyendo \sqrt{r} , si no mas bien la hemos sustituido en lugar de r^2 cerca de donde esperamos la singularidad, tal y como lo hicimos con las funciones base especiales.

seccion 7.3 Refinamiento local

La otra clase de métodos es aquella donde, si bien se sabe que existe una singularidad, se ignora su forma. Tienen la ventaja de poderse aplicar donde no es conocida la forma de la singularidad. Estos métodos conllevan el refinar la malla de alguna manera. Obviamente, si se refina la malla solamente alrededor de la singularidad, es decir

donde realmente es necesario, se obtienen mejores aproximaciones a menor costo que si se refina uniformemente, pues al crecer el número de elementos se dispara el tamaño del sistema a resolver.

Como ya se vio, al tener una malla fina se obtiene una mejor aproximación a la solución (utilizando funciones base cuadráticas), pero por ser esto caro, solo vale la pena hacerlo cuando hay razones para suponer que esto mejorara sensiblemente la aproximación, es decir cerca de la frontera, donde se espera que la solución se comporte de forma no cuadrática, por estar cerca de la singularidad.

Es decir, se necesitan cada vez intervalos mas pequeños al acercarse a la singularidad. Al no suponer un comportamiento específico cerca de la singularidad, es posible obtener aproximaciones mejores a la del método standard aun cuando la singularidad sea de otro tipo al estudiado, lo que no sucede con los otros dos métodos vistos.

A pesar de lo dicho anteriormente, cuando si es conocida la forma de la singularidad, como en nuestro caso, es posible refinar la malla de manera que podamos mejorar aun mas la aproximación a la solución, e inclusive recuperar el orden de convergencia obtenida para un problema no singular:

Una regla útil que permite refinar localmente para mejorar la convergencia del método en presencia de singularidades (ver [5] [pag. 155]) es sugerida por la propiedad de que para el elemento e_k la diferencia entre u y v satisface:

$$|u - v|_{s, e_k} \leq C h_k^{k-s} |u|_{k, e_k}$$

La regla sugerida es refinar de manera que

$$h_k^{k-s} |u|_{k, e_k}$$

permanezca constante de un elemento a otro. En una dimensión esto significa que $h_k^{k-s + \frac{1}{2}} \times d - k$

(donde x^0 es la singularidad en el origen) debe permanecer aproximadamente constante. Esto dice que se puede lograr el mismo orden de exactitud para una solución singular que para una regular si hacemos un refinamiento apropiado.

En el caso unidimensional se escogió el esquema de refinamiento que se ilustra en la figura 1, el cual no cumple con el criterio arriba expuesto, por lo cual aunque esperamos una mejor aproximación a la solución, no esperamos recuperar por completo el orden de convergencia del problema no singular.

El subintervalo mas cercano a una singularidad, es dividido en cuatro subintervalos iguales; ahora de estos se toma el mas cercano a la singularidad y se divide otra vez en cuatro partes, etc. En la practica estamos haciendo esto solo dos veces, convirtiendo en siete este primer subintervalo; Como lo hacemos en realidad en el primero y ultimo subintervalos, se estan agregando doce subintervalos. Teoricamente esto debe dar siempre un mejoramiento (para problemas elipticos) sobre el metodo standard, como es claro en [2] (pag 47) a pesar de que el refinamiento pueda no ser el optimo desde el punto de vista del criterio arriba mencionado.

seccion 7.4 Diferencias Finitas y Elemento Finito

El punto en el que el metodo de elemento finito ha contribuido con una idea nueva y valiosa con respecto a la técnica de diferencias finitas establecida es que en vez de operar con una variable desconocida ^{de} de manera que haya una ecuación por punto de la malla, en F.E.M. es posible acoplar desplazamientos y pendientes desconocidas logrando mas exactitud y aproximando derivadas de mayor orden sin abandonar la naturaleza local de la ecuación de diferencias.

Entre mas irregular sea la malla o curva la frontera mas importante se vuelve esto, ya que entre las mayores ventajas sobre el método de diferencias finitas esta la de tener una mayor habilidad para tratar cualquier tipo de dominio o condición de frontera con propiedades físicas diversos y la de poder lograr mayor exactitud utilizando elementos de mayor orden, dando una aproximación continua a la solución desconocida y permitiendo dar a la computadora mas responsabilidad en la construcción de la discretización, así como el ensamblado y solución de las ecuaciones de "stiffness", es decir logrando una mayor automatización.

Dado el caracter local de las funciones base, se tienen ecuaciones del tipo de diferencias finitas. Se obtienen matrices bandedas y que por tanto pueden invertirse de manera mas rápida.

Sin embargo, como se puede apreciar en las referencias de Whiteman (ver [3] y [8]) ni el metodo de elemento finito ni el de diferencias finitas en sus formas standard son buenos para resolver problemas con singularidades en la frontera, pero las modificaciones al método de elemento finito tienen un éxito evidente en mejorar la aproximación de las soluciones numéricas (ver [12]).

Whiteman (ver [3]) discute los resultados obtenidos por los métodos standard de elementos finitos y de diferencias finitas en casos singulares. El método de Elementos Finitos permite siempre una mejor aproximación a menor costo. Whiteman ha usado también refinamiento local y la inclusión de funciones singulares en los métodos standard de diferencias finitas, que mejoran la aproximación a la solución, así que no es imposible utilizar ideas similares en ambos casos; sin embargo parece ser que la implementación es mas natural en el caso de el metodo de elemento finito.

CAPITULO 8 Resultados en el caso unidimensional

En resumen, las modificaciones más importantes hechas al programa original para obtener el programa final del caso unidimensional son:

- 1) Adaptación al caso isoparamétrico.
- 2) Tratamiento del caso no lineal.
- 3) Inclusión de las técnicas que toman en cuenta las singularidades.

Todo esto para permitir resolver ecuaciones parabólicas no-lineales y encontrar una mejor aproximación a la solución en los casos de problemas singulares con un comportamiento conocido, que es el caso que nos interesa.

En este capítulo se aplican los métodos arriba descritos a problemas específicos. Es importante notar que los problemas se escogieron por ser representativos, i.e. el primer problema es elíptico-lineal; el segundo ya es no-lineal, y así hasta llegar a un problema parabólico, no-lineal y con una singularidad en la frontera que se comporta como deseáramos (como \sqrt{r} donde r es la distancia a la singularidad) el cual es el que realmente nos interesa; no obstante, a pesar de que corro el riesgo de perder al lector, creo que los resultados intermedios ayudan a entender lo que está sucediendo.

Se revisan en orden: Simplemente hay que recordar que la ecuación es

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(p \frac{\partial u}{\partial x} \right) - q u + f$$

y ver que son p, q, f cual es u (la solución analítica), sin olvidar las condiciones de frontera, que en este capítulo serán $u(0) = u(1) = 0$ si $u = u(x)$ y $u(0, t) = u(1, t) = 0$ si $u = u(x, t)$ para todos los problemas.

Cada problema es resuelto mediante cuatro métodos, a saber: el de elemento finito con elementos regulares, el de elementos con puntos cuartos, el de refinamiento local, y finalmente el de funciones base especiales. Dentro de cada uno, se analiza el efecto de distintas mallas en X (cuatro), y en t (dos), donde t es tiempo, comparando no solo los resultados entre los diferentes métodos, sino entre las distintas mallas para cada uno.

Se utiliza una malla regular con $NI=21$ puntos para el cálculo de error; esta se obtiene de distribuir de manera regular los puntos sobre el intervalo $[0, 1]$. A partir de esta malla se calculan los errores, utilizando dos estimados distintos del error:

1) El error máximo se calcula como el máximo valor absoluto de las diferencias entre los valores de u , la solución analítica, y U , la aproximación encontrada, para cada punto en esta malla, es decir

$$\max_{NI} |u(x_i) - U(x_i)| \quad 1 \leq i \leq NI$$

2) El error que llamaremos L_2 (realmente es una aproximación al error RMS o "root mean square" que es la norma de Sobolev que se está utilizando para minimizar) es decir el error L_2 discreto:

$$\sqrt{\frac{1}{NI} \left(\sum_{i=1}^{NI} (u(x_i) - U(x_i))^2 \right)}$$

En la tabla 0 tenemos la presentación resumida de los problemas independientes del tiempo que se analizaron: son los siguientes:

1) Como primer problema se tiene que:

$$p=1$$

$$q=0$$

$$f = \pi^2 \sin(\pi x)$$

$$u = \sin(\pi x)$$

es decir que

$$\frac{\partial^2 u}{\partial x^2} + \pi^2 \sin(\pi x) = 0$$

El problema es elíptico, lineal, y no tiene singularidades; tampoco muestra un comportamiento tipo \sqrt{r} cerca de la frontera, por lo cual al hacerle el conjunto de pruebas no se espera ninguna mejoría por tratarlo con los diferentes métodos.

2) El segundo problema es un poco más complejo pues no es lineal, pero todavía se espera una solución no singular; pues aquí:

$$p = 1 + u^2$$

$$q = 0$$

$$f = \pi^2 \sin(\pi x) (1 - 3 \sin^2(\pi x))$$

$$u = \sin(\pi x)$$

Aquí tampoco hay dependencia del tiempo, por lo que la ecuación queda como:

$$\frac{\partial}{\partial x} \left((1+u^2) \frac{\partial u}{\partial x} \right) - \pi^2 \sin(\pi x) (1 - 3 \sin^2(\pi x)) = 0$$

El problema es elíptico, no lineal, sin singularidades; no se comporta como \sqrt{r} .

No se espera ninguna mejoría resultante de aplicar los métodos modificados, mas bien que haya pérdida en cuanto a precisión en el caso de elementos con puntos cuartos y en el de las funciones base especiales.

3) El tercer problema es elíptico no-lineal, con una singularidad en la frontera (i. e. degenera, allí) pero con una solución no-singular.

Aquí: $p = u$

$$q = 0$$

$$f = -\pi^2 (1 - 2\sin^2(\pi x))$$

$$u = \sin(\pi x)$$

En este caso no se tiene dependencia del tiempo, y entonces la ecuación queda como

$$\frac{d}{dx} \left(u \frac{du}{dx} \right) - \pi^2 (1 - 2\sin^2(\pi x)) = 0$$

Tampoco se espera ninguna mejoría notable resultante de aplicar los métodos modificados que implican conocer la forma de la singularidad (es decir el de elementos con puntos cuartos y el de funciones base especiales), aunque tampoco deben ser muy malos ya que γ^2 esta ausente en la expansión asintótica. Es de esperarse una pequeña mejoría con el método de refinamiento local.

4) El cuarto problema es elíptico no-lineal

con una singularidad y una solución singular de la forma \sqrt{x} con

$$p = u$$

$$q = 0$$

$$f = \pi^2 u^2 / 2$$

$$u = \sqrt{\sin(\pi x)}$$

donde entonces

$$\frac{d}{dx} \left(u \frac{du}{dx} \right) + \frac{\pi^2 u^2}{2} = 0$$

Ahora se espera una mejoría notable por aplicar los métodos modificados. En refinamiento local se espera buen funcionamiento simplemente por que se esta agregando intervalos a la malla allí donde se necesitan; en elementos con puntos cuartos y funciones base especiales porque se esta simulando un comportamiento tipo \sqrt{y} en la frontera para la aproximación, conociendo que la solución analítica de hecho tiene tal comportamiento.

En la tabla 1 se analiza el error máximo para el problema 1, utilizando el metodo standard, es decir el de elementos regulares, a lo largo del tiempo para $\Delta t=0.1$ y $\Delta t=0.005$. La tabla 2 es un análisis semejante para el error L2. En ambos casos, y tanto para $\Delta t=0.1$ como para $\Delta t=0.005$ se observan oscilaciones, aunque estas parecen estabilizarse al avanzar en el tiempo. Las oscilaciones se presentan sobre todo cuando NI es grande, y pueden ser debidas a que se está usando un método de ecuaciones parabólicas para resolver un problema elíptico, lo cual nos asegura que a la larga se tiene convergencia a la solución del problema elíptico discretizado, pero no que esto sucederá de manera inmediata (es mas, las oscilaciones que se presentan en un principio deberan amortiguarse si dejamos transcurrir tiempo suficiente).

Es difícil observar de manera precisa la influencia de la discretización en el espacio sobre el error en estas pruebas pues nunca dejamos transcurrir tiempo suficiente como para observar una estabilización.

Hay modos oscilantes presentes al principio pero estos se amortiguan rápidamente. Son modos del tipo "stiff" que no están en la solución analítica. Como la solución analítica y la solución del problema elíptico discretizado no coinciden, hay componentes de estos modos en los primeros pasos de tiempo. Para analizar el error de la discretización de problemas elípticos es mejor utilizar valores en un tiempo avanzado cuando las oscilaciones se han amortiguado.

Es importante recordar que hay mas modos oscilantes cuando la discretización del espacio es mas fina (ver Strang y Fix [pag. 245-250]), por lo cual refinar la malla en espacio puede empeorar la situación en cuanto a convergencia en tiempo.

Como a pesar de todo esto se observa una cierta tendencia a estabilizarse, en la tabla 3 se analiza el problema 1 para todos los métodos únicamente en el último tiempo, es decir $t=0.3$, donde hay más estabilidad; lo hacemos para las dos mallas de tiempo utilizadas; es fácil ver que el comportamiento observado para ambas mallas es tan parecido que no vale la pena analizar los dos casos; de aquí en adelante se analizan los demás problemas independientes de tiempo en $t=0.3$ y $t=0.1$, ya que para esta malla necesitamos hacer muchos menos cálculos.

Del análisis de las tablas 1 a 5 y las gráficas 1 a 5 es claro que tal como esperábamos no hay una mejora importante con ninguno de los métodos usados en el caso de los problemas 1 a 3, los cuales no se comportan como \sqrt{h} . Se observa una mejora pequeña para refinamiento local tal cual se había predicho, pero nada para los elementos con puntos cuartos o funciones base especiales. Esto es lo que se esperaba pues se está desechando la función base $\sqrt{2}$ en favor de la inútil \sqrt{h} . Sin embargo, como en estos problemas la expansión de Taylor de la solución analítica nos muestra que tampoco es útil $\sqrt{2}$ los errores con puntos cuartos no son significativamente distintos a los obtenidos con elementos regulares, especialmente cuando los elementos son suficientemente pequeños.

Para el caso en el que la solución sí se comporta como \sqrt{h} , es decir el problema 4, se observa una importante mejora para los métodos usados en comparación con el método original como puede comprobarse en la gráfica 5 y en la tabla 13. El análisis de estos resultados debe tomar en cuenta que estamos deteniendo el proceso cuando la diferencia entre iteraciones es menor que un número de orden 10. En el problema 4 esto sucede para $N=40$ donde no se llega a hacer suficientes iteraciones para converger a un resultado, pero el criterio usado detiene el proceso. Por tanto, los resultados no son confiables; haremos el análisis sobre las demás mallas. De hecho deberíamos utilizar otros valores para detener el proceso en este caso, pero esto haría difícil hacer comparaciones.

Parece ser que el método de refinamiento local es mejor para las mallas gruesas. Pero sin duda alguna el método de elementos con

puntos cuartos y el de funciones base especiales son más baratos en estos casos, ya que para refinar localmente, agregamos a la malla en x 12 intervalos, lo cual incrementa grandemente el tamaño del sistema a resolver. Para mallas de 5, 10 (e inclusive 20) intervalos. Este mismo incremento en el tamaño explica el buen funcionamiento del refinamiento local.

Para el problema 4 se obtienen buenos resultados con los métodos de elementos con puntos cuartos y funciones base especiales. En general para mallas finas es mejor el método de elementos con puntos cuartos que el de funciones base especiales; para las mallas gruesas (esto también se observó en los problemas anteriores) la diferencia es menor pero a favor de el método de funciones base especiales, ya que si bien hay mas errores de redondeo para estas, para las mallas gruesas los errores de redondeo son poco importantes; por otra parte el tener una malla irregular produce errores grandes sobre todo en mallas gruesas. De hecho se tiene una malla no uniforme en el caso de elementos con puntos cuartos y no en el de las funciones base especiales.

En cuanto a la convergencia (respecto a h) de los métodos, tal como lo predice la teoría, como indican las tablas 1 a 5 y las gráficas 1 a 4, encontramos ordenes de convergencia de h^3 o mas en aquellos problemas en que no hay soluciones singulares tanto para el error máximo como para el L_2 (es decir los problemas 1, 2 y 3) encontramos mas debido a factores que pueden causar superconvergencia, como el que la malla en que se calcula el error contenga en su mayoría nodos de los elementos finitos. La gráfica número 1 así como la tabla número 12 son muy claras en este aspecto. Ver también los problemas 1, 2 y 3 según la tabla 13. Para el problema 4 no es tan claro que como era de esperarse según la teoría logremos restaurar el orden de convergencia que se tenía para el caso de el método con elementos regulares en los problemas sin singularidades. Para las mallas con $NI=5, 10$ y 20 parece ser que si se observa esto, pero, como ya se había mencionado, los resultados con $NI=40$ parecen ser poco confiables debido al criterio que se utilizó para detener el proceso.

Esto implica que es conveniente trabajar con mallas suficientemente finas en x ya que por ejemplo, al duplicar el esfuerzo se obtienen resultados varias veces mejores.

seccion 8.2 Problemas parabolicos id.

Se analizan ahora de manera similar los problemas parabólicos seleccionados; veamos que sucede al aplicar los métodos modificados. Estos son problemas parabólicos, por lo cual es conveniente ver que sucede a lo largo del tiempo, al menos para el error L2; Como el efecto de las distintas mallas en x es ya conocido, al igual que el efecto de distintas mallas en tiempo, parece suficiente analizar los resultados para $N1=10$ y para $\Delta t=0.1$. Se escoge justamente $N1=10$ pues se encuentra una situación un poco extrema (al menos en apariencia), en cuanto a las mallas en x ya que se observa que si $N1$ vale 20 o 40, la precisión no cambia mucho; Desde luego esto es explicable si se toma en cuenta cual es la precisión que se tiene y cual el criterio que detiene los calculos. De hecho, no se obtiene mayor precisión al utilizar mas intervalos debido a que en un momento dado resulta que los cambios en cada iteración son menores que el valor fijado para terminar las iteraciones, y por tanto la iteración se detiene.

Esto dificulta comparar los métodos entre sí pues no todos convergen de la misma manera. Precisamente para evitar estas distracciones sin sacrificar precisión se escoge $N1=10$ para las tablas. La selección de problemas se hace con un criterio idéntico al usado para los problemas independientes del tiempo, es decir:

1) Un problema parabolico lineal (problema 5), con una solución no-singular en el que al ser

$$p=1$$

$$q=0$$

$$f=0$$

$$u = \sin(\pi x) e^{-\pi^2 t}$$

la ecuacion diferencial queda como

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

2) Un problema parabolico no-lineal, sin singularidad, y con una solución no singular (problema 6): $p = 1 + u^2$, $q = 2\pi^2 \cos^2(\pi x) e^{-2\pi^2 t}$
 $f = \pi^2 u^3$ $u = \sin(\pi x) e^{-\pi^2 t}$

Entonces la ecuación queda como

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left((1 + u^2) \frac{\partial u}{\partial x} \right) - (2 \pi^2 \cos^2(\pi x) e^{-2\pi^2 t}) u + \pi^2 u^3$$

3) Un problema parabólico singular no lineal, con una solución no-singular (sin comportamiento tipo \sqrt{t} en la frontera) (problema 7). Aquí:

$$p = u$$

$$q = 1/(t - t_0)$$

$$f = -\cos(2\pi x) / (\pi(t - t_0))^2$$

$$u = \sin(\pi x) / (\pi(t - t_0))^2$$

$$t_0 = -1$$

Entonces se tiene

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(u \frac{\partial u}{\partial x} \right) - \frac{1}{t - t_0} \left(u - \frac{\cos(2\pi x)}{\pi^2(t - t_0)} \right)$$

Como último problema (problema 8) se tiene uno parabólico no-lineal, con una singularidad y una solución singular de la forma \sqrt{t} :

$$p = 1$$

$$q = 1$$

$$f = \pi^2 u^2 / 2$$

$$u = \sqrt{\sin(\pi x)} e^{-t}$$

Entonces la ecuación queda como

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(u \frac{\partial u}{\partial x} \right) - u + \frac{\pi^2 u^2}{2}$$

Este último problema es el problema que en un principio nos planteamos resolver, es decir, es el problema de interés.

Para este último problema se espera una mejoría notable por aplicar los métodos modificados (en forma parecida a lo observado en el problema número 4, que aunque independiente del tiempo era similar). Para refinamiento local se espera buen funcionamiento simplemente por que estamos agregando intervalos a la malla allí donde se necesitan.

También se espera buen funcionamiento al usar las técnicas de elementos con puntos cuartos y funciones base especiales porque estamos simulando un comportamiento tipo \sqrt{x} en la frontera para la aproximación, conociendo que la solución analítica de hecho tiene tal comportamiento.

El resumen de los problemas parabólicos unidimensionales y los resultados obtenidos se muestran en las tablas 6 y 7 respectivamente:

Como puede observarse, tal cual se había predicho, para los problemas 5 y 6, el método de refinamiento no es significativamente mejor que el método con elementos regulares. Los de elementos con puntos cuartos y funciones base especiales deberían ser peores, pero como la forma asintótica de la solución no tiene término en x^2 la diferencia es menor a la esperada, e inclusive a veces a favor de estos métodos. Ambos problemas evolucionan en el tiempo de manera que la diferencia entre los errores es cada vez menor aunque el error en sí crece.

Esto se debe a que el modo fundamental del sistema discretizado es parecido para las diferentes discretizaciones, pero no coincide con el modo fundamental del sistema diferencial.

En el problema 7 sin embargo, por razones que no alcanzo a comprender, resulta que el método de funciones base especiales es bastante bueno comparado con el de elementos con puntos cuartos en cuanto a orden de magnitud. Sin embargo este último resulta ser constante en el tiempo mientras que los demás métodos muestran un crecimiento aproximadamente lineal del error. Para $t = 0.3$ la diferencia entre funciones base especiales y elementos con puntos cuartos ya es pequeña.

Para el problema B, que es el problema de interés, realmente se observa la mejoría esperada para los métodos modificados en

comparacion con el metodo original. El método de refinamiento local es obviamente superior (para esta malla), ya que su error es de un orden de magnitud de 10^{-5} , contra uno de 10^{-4} con funciones base especiales y elementos con punto cuarto y una de 10^{-3} para el método original. También resulta una mayor estabilidad comparativa en todos los métodos para este problema ya que solamente se tienen pequeñas oscilaciones.

En cuanto a la evolución en el tiempo de los problemas parásiticos es bastante descorazonadora; por ser pocos tiempos no se ve claro si estan oscilando, (lo cual parece ser cierto en general) o si de inclusive estan disparandose.

Como se ilustra en la grafica B inclusive aquellos problemas cuyo error no esta disminuyendo tienen diferencias decrecientes entre los errores, lo cual no es tan mala si recordamos que los errores en tiempos anteriores se estan acumulando al usar una aproximación en t_{i-1} para calcular la aproximación en t_i .

Como ya se mencionó, el modo fundamental del sistema discretizado no coincide con el del sistema diferencial. Aparentemente estamos convergiendo (de manera semejante para todos los métodos) a la solución del sistema discretizado; esta no parece coincidir con la solución analítica.

En resumen, los resultados confirman la utilidad de los métodos modificados para los problemas unidimensionales que nos interesan; Si se conoce la forma de la singularidad conviene utilizar los métodos que la toman en cuenta, i.e. funciones base especiales y elementos con puntos cuartos. De lo contrario conviene utilizar refinamiento local. Para mallas gruesas el refinamiento local es caro.

El análisis hecho para ecuaciones en una dimensión (espacial) tenía el objetivo de estudiar el funcionamiento de las diversas técnicas en el tratamiento de ecuaciones parabólicas con singularidades generadas por un coeficiente de difusión degenerado. La mayoría de los casos de interés práctico (ver por ejemplo [1] y [2]) son de dos o más dimensiones.

En seguida se hace un estudio en dos dimensiones con el objeto de averiguar si los métodos funcionan de manera similar a como lo hacen en el caso unidimensional. El programa construido resuelve la ecuación parabólica bidimensional

$$\frac{\partial u}{\partial t} = \nabla \cdot (\rho \nabla u) - q \cdot u + f$$

para $u(x, y, t)$ una región rectangular con condiciones de frontera:

$$\begin{aligned} u &= 0 & \forall (x, y) \in \partial \Omega_1 \\ n \cdot (\rho \nabla u) &= 0 & \forall (x, y) \in \partial \Omega_2 \end{aligned}$$

Utiliza elementos triangulares, y hace de manera completamente automática el cálculo de esta malla de elementos triangulares. Es el programa PROGRAMA/2D y puede consultarse en los apéndices.

La malla efectivamente usada puede verse en la figura 3. Strang (pag. 152) muestra que esta manera de construir la malla es mejor que si se consideran rectángulos con las dos diagonales para formar los triángulos; además escoger la diagonal adecuada según el problema permite lograr matrices con bandas ligeramente menores, haciendo un poco más eficiente el trabajo.

Es una malla triangular formada a partir de una malla rectangular de la siguiente manera:

Se leen dos numeros NI, NJ que son simplemente el número de intervalos que tendrá la malla rectangular en (x, y) respectivamente.

Suponiendo por el momento que estamos utilizando el método de elemento finito con malla regular (es decir al método de elementos finitos con elementos regulares, sin tomar en cuenta el caso de las singularidades) lo que hacemos es simplemente construir la malla rectangular y dividir cada rectángulo en dos triángulos rectángulos cuya hipotenusa común es una diagonal del rectángulo. Se escogió la diagonal con pendiente $\frac{\partial y}{\partial x} < 0$. Así, dividimos cada rectángulo en un triángulo "superior" y un triángulo "inferior".

Aquí $p = p(x, y, t, u)$, $q = q(x, y, t, u)$, $f = f(x, y, t, u)$ o sea que como estas pueden ser funciones de la solución, no estamos restringidos a resolver la ecuación lineal, sino que en general, y es este el caso que nos interesa, es posible tener una ecuación no-lineal.

La estructura es muy similar a la del caso unidimensional, pero desde luego hay diferencias importantes en varios aspectos, que son precisamente las que se analizan a continuación.

seccion 9.1 Transformación isoparamétrica

La transformación entre un triángulo en el espacio (x, y) y el triángulo unitario standard en el espacio (ξ_1, ξ_2) se define como:

$$(x, y) = \tau(\xi_1, \xi_2) = \sum_{i=1}^3 \gamma_i P_i(\xi_1, \xi_2, \xi_3)$$

donde los P_i son las coordenadas (x, y) de los seis nodos del triángulo unitario cuadrático standard ordenados como se ve en la figura 4b.

Por la forma de construir la malla, en el plano (x, y) hay dos tipos distintos de triángulos, inferiores y superiores. Cada uno de estos tipos esta numerado de forma distinta pero compatible con respecto al mapeo al triángulo unitario cuadrático standard en (ξ, ζ) , (η, ζ) .

Las coordenadas locales (ξ, ζ) se definen en términos de (x, y) mediante la inversa de la transformación isoparamétrica. La necesidad de que esta este bien definida impone varias condiciones sobre la disposición de los puntos intermedios, pues al igual que en el caso unidimensional, en ciertas partes el determinante jacobiano J de la transformación isoparamétrica se anula (recordemos que precisamente esta propiedad era aprovechada en los elementos de puntos cuartos).

seccion 9.2 La inversa a la transformación isoparamétrica

Mediante la rutina EVAL, dados (x, y) y sabiendo que se esta en el triángulo determinado por las coordenadas x_1, y_1, x_2, y_2 donde $x \in [x_1, x_2]$ se quieren determinar las coordenadas locales ξ, ζ .

Si se tiene alguna estimacion inicial $\xi^{(0)}, \zeta^{(0)}$ de ξ, ζ (se utilizó 0.3 como valor inicial de ambas); entonces como $x = x(\xi, \zeta)$ y $y = y(\xi, \zeta)$ son las transformaciones isoparamétricas, se tiene que $x^{(0)} = x(\xi^{(0)}, \zeta^{(0)})$ y $y^{(0)} = y(\xi^{(0)}, \zeta^{(0)})$. El proceso de Newton consiste en estimar $\xi^{(k)}, \zeta^{(k)}$ usando de manera iterativa (si k es el índice de iteración):

$$\begin{pmatrix} \xi^{(k+1)} - \xi^{(k)} \\ \zeta^{(k+1)} - \zeta^{(k)} \end{pmatrix} = - \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \zeta} \end{pmatrix}^{-1} \begin{pmatrix} x^{(k)} - x \\ y^{(k)} - y \end{pmatrix}$$

En EVAL se hace esto de manera explícita, es decir:

- 1) Calcular $\frac{\partial x}{\partial z_1}$, $\frac{\partial x}{\partial z_2}$, $\frac{\partial y}{\partial z_1}$ y $\frac{\partial y}{\partial z_2}$. En otras palabras, se tiene la matriz jacobiana (de la transformación isoparamétrica):

$$J = \begin{pmatrix} \frac{\partial x}{\partial z_1} & \frac{\partial x}{\partial z_2} \\ \frac{\partial y}{\partial z_1} & \frac{\partial y}{\partial z_2} \end{pmatrix}$$

- 2) A partir de estos valores se obtiene el determinante jacobiano:

$$J = \begin{vmatrix} \frac{\partial x}{\partial z_1} & \frac{\partial x}{\partial z_2} \\ \frac{\partial y}{\partial z_1} & \frac{\partial y}{\partial z_2} \end{vmatrix}$$

- 3) Se utiliza CALC para obtener a partir de un valor inicial para z_1 y z_2 valores correspondientes (usando la transformación isoparamétrica) $x_0 = x(t_0)$ y $y_0 = y(t_0)$.

Para calcular la inversa de la matriz jacobiana, se puede utilizar el siguiente resultado:

$$\begin{pmatrix} \frac{\partial x}{\partial z_1} & \frac{\partial x}{\partial z_2} \\ \frac{\partial y}{\partial z_1} & \frac{\partial y}{\partial z_2} \end{pmatrix}^{-1} = \begin{pmatrix} \frac{\partial z_1}{\partial x} & \frac{\partial z_1}{\partial y} \\ \frac{\partial z_2}{\partial x} & \frac{\partial z_2}{\partial y} \end{pmatrix} = \frac{1}{J} \begin{pmatrix} \frac{\partial y}{\partial z_2} & -\frac{\partial x}{\partial z_2} \\ -\frac{\partial y}{\partial z_1} & \frac{\partial x}{\partial z_1} \end{pmatrix}$$

Este resultado es sustituido en los lugares adecuados del proceso de Newton para que finalmente tener:

$$r_1 = x - x^{(k)}$$

$$r_2 = y - y^{(k)}$$

$$f_1 = ((\partial y / \partial z_2) r_1 - (\partial x / \partial z_2) r_2) / J$$

$$f_2 = ((-\partial y / \partial z_1) r_1 + (\partial x / \partial z_1) r_2) / J$$

$$z_1^{(k+1)} = z_1^{(k)} + f_1$$

$$z_2^{(k+1)} = z_2^{(k)} + f_2$$

y aproximar de manera iterativa a los valores de β_1, β_2 cuidando evitar que estos queden por error fuera del triángulo.

Se hace esto hasta cuatro veces, o hasta que tanto $|f_1| < \epsilon$ como $|f_2| < \epsilon$. En el programa (ver el listado en el apéndice PROGRAMA/20):

- 1) $\beta_1, \beta_2, \beta_3$ se llaman T1, T2, T3.
- 2) RJAC significa J , el determinante Jacobiano.
- 3) DXT1, DXT2, DYT1 y DYT2, significan respectivamente $\partial x/\partial \beta_1, \partial x/\partial \beta_2, \partial y/\partial \beta_1$ y $\partial y/\partial \beta_2$.
- 4) A $x^{(k)}, y^{(k)}$ resultantes de la transformación isoparamétrica se les llama XX, YY, mientras que a x, y se les llama XIMP, YIMP respectivamente. XI, YI son vectores de seis elementos donde se guardan las coordenadas de los nodos del triángulo en el cual se está trabajando en ese momento.
- 5) Además RES1, RES2, FAC1, FAC2 son variables temporales utilizadas para cálculos; las últimas dos son usadas también para comparar con ϵ , o sea EPS.
- 6) Como para una dimensión, las UIJ son las funciones base o sus derivadas según el valor de ISP, que corresponde a lo que era ID en una dimensión. La convención en este caso es que ISP=1 significa que se desea la función base, que si ISP=2 es que se requiere la parcial respecto a β_1 , y si ISP=3 es que interesa la parcial respecto a β_2 . La propiedad arriba ilustrada acerca de la matriz Jacobiana es utilizada en el programa ya que se tienen las funciones base y sus derivadas en función de β_1, β_2 y no de x, y .
- 7) El parámetro que detiene la iteración, EPS se fija externamente.

De manera similar al caso unidimensional, se utiliza Integración de Gauss, pero recordando que aquí se necesitan mas puntos para obtener la precisión deseada, (i. e. siete y no cuatro). El tipo de integral deseada es:

$$\int_{-1}^1 \left(\int_{-1}^{1-z_1} \frac{z_1 z_2}{\rho(z_1, z_2)} dz_2 \right) dz_1$$

la aproximación usada es la formula de Hammer:

$$\frac{1}{2} \sum_{k=1}^7 \left(f^k \frac{\partial(x, z)}{\partial(z_1, z_2)} \right) \Big|_{(z_1^{(k)}, z_2^{(k)})} w_k$$

Si utilizamos los pesos adecuados, w_k que se pueden consultar en Strang y Fix (pags. 183-184), Tong & Rossetto (pags. 109-190), siendo f^k la función adecuada valuada en los puntos x^k, y^k allí listados. entonces se puede esperar una precisión del orden de h^2 . En este contexto $O(h^2)$ significa que la integración numérica es exacta si la función integrada es un polinomio de orden ≤ 1 en z_1, z_2, z_3 .

seccion 9.4 Elementos singulares con punto cuarto

Ya se mencionó que si ciertos nodos laterales de un elemento cuadrático triangular se ponen en los puntos cuartos, entonces las funciones base pueden representar funciones de la forma \sqrt{r} alrededor de la singularidad, donde r es la distancia a esta singularidad.

Esto se ha hecho ya para elementos que tienen una singularidad en un nodo (ver [4]) pero aquí queremos además el caso en que todo un lado es singular. Supongamos un triángulo P1P2P3 con P4 en P2P3, con P5 en P3P1 y con P6 en P1P2 como muestra la figura 5.

Hay varios casos; supongamos primero que se tiene la disposición P6 en el punto cuarto mas cercano a P1 sobre P1P2 y P5 en el punto cuarto mas cercano a P1 sobre P1P3, (fig 4b).

Se tiene entonces que para cualquier punto sobre las líneas que van de P1 a cualquier punto en P2P3 si r es la distancia a P1 entonces:

$$x - x_1 = [(x_3 - x_1)z_3 + (x_2 - x_1)z_2] (z_2 + z_3)$$

$$y - y_1 = [(y_3 - y_1)z_3 + (y_2 - y_1)z_2] (z_2 + z_3)$$

Por ejemplo, cuando $z_2 = 0$, se tiene que si l_{13} es la longitud del lado P1P3 entonces $r = (z_3)^2 l_{13}$. Este es el caso con una singularidad alrededor de P1.

Otro caso es cuando, por ejemplo, P4 está en el punto cuarto cercano a P2 sobre P2P3 y P5 esta en el punto cuarto cercano a P1 sobre P1P3. En este caso suponemos la distancia a P1P2 para cualquier punto a lo largo de rectas que van de P3 a puntos sobre P1P2.

Así, si (x_c, y_c) es la proyección del punto (x, y) desde P3 sobre P1P2 entonces:

$$x - x_c = (x_3 - x_c) z_3^2$$

$$y - y_c = (y_3 - y_c) z_3^2$$

En tal caso, si h_3 es la distancia perpendicular de P3 a P1P2 que es el lado singular, resulta que $r = h_3^2$. En ambos casos el comportamiento es como se deseaba para el problema a tratar, i.e. como \sqrt{r} .

Ya se habia comentado que usar el metodo de puntos cuartos es de cierta manera equivalente a usar funciones especiales pues en vez de transformar la funcion base, se transforma el triángulo, de manera que expresando las funciones base cuadráticas como función de x, y y no de z_1, z_2 , de hecho tiene otra base para el mismo espacio (el formado al incluir las funciones base especiales).

Por cuestiones de tiempo se decidió no programar el caso de funciones especiales para el caso bidimensional. Es de suponer, sin embargo, que no debe haber mayor problema en esta extensión. Queda por ver si esto es cierto en un estudio posterior.

Viendo globalmente la malla con elementos singulares en el caso de singularidades puntuales contra el caso de singularidades en partes de la frontera, resulta claro que en este segundo caso se está perturbando la regularidad y simetría de la malla de una manera mucho mayor. Por lo cual, a pesar de que estamos aproximando mejor en la dirección en que está la singularidad, en la otra dirección (y justamente este efecto es mayor cerca de las singularidades) estamos aproximando mucho peor.

Es por esto que, al menos para mallas gruesas pueden presentarse serios problemas en cuanto a este método. Veamos de todas maneras que sucede con él en la práctica. Hay que aclarar que este tipo de perturbaciones geométricas no se presentarían en el caso de funciones especiales, por lo cual, de hecho podría resultar ser un mejor método en problemas de este tipo.

seccion 9.5 Refinamiento local

En cuanto al refinamiento local se aplica el siguiente esquema: consiste en dividir en la dirección correcta según la singularidad de manera similar a como se hizo en el caso unidimensional, pero en vez de dividir en 4, cada subintervalo, se divide en 2, es decir, como se indica en la figura número 6, cumpliendo con el criterio enunciado en el caso general.

No debe haber mayores problemas en extender este método,

En este capítulo se aplican los métodos anteriormente descritos a problemas específicos en dos dimensiones. Se escogieron problemas semejantes a los de una dimensión que más que un estudio de los casos representativos es una simple muestra de los resultados encontrados para una dimensión que siguen siendo válidos y las dudas que hay acerca de la validez de los otros. No es un estudio tan sistemático como en el caso unidimensional, ya que solo se trata de observar que resultados se extienden al caso bidimensional.

Además de analizar algunos de los problemas que se habían escogido para una dimensión con fines de comparación, se tomaron otros problemas parecidos. Revisémoslos entonces en orden: Simplemente hay que recordar que la ecuación es

$$\frac{\partial u}{\partial t} = \nabla \cdot (\rho \nabla u) - q u + f$$

y ver que son ρ, q, f , cuales son las condiciones de frontera aplicables y desde luego cual es u en cada caso.

Cada problema es resuelto mediante tres métodos, a saber: el de elemento finito con elementos regulares, el de elementos con puntos cuartos, y el de refinamiento local. El de funciones base especiales no se analizó más pero no hay razón para suponer que los resultados vistos para una dimensión, que indicaban que en la forma en que fue implementado de hecho era parecido al método de puntos cuartos pero con la transformación hecha de manera explícita y sin perturbaciones geométricas (por lo cual tal vez no fue la mejor decisión posible el no analizarlo) dejan de ser válidos.

Además se muestra que el efecto observado para distintas mallas también puede extenderse en este caso.

sección 10.1

El problema $u = \sqrt{\sin(\pi y)}$

Este fue el problema que se analizó más a fondo; en lo sucesivo le llamaremos problema "a".

Fue tratado por distintos métodos y mallas; debido a que los resultados para el método de punto cuarto no fueron tan como se esperaban en un problema de este tipo (en el que debería mejorar la aproximación debido a que es un problema singular que se comporta como \sqrt{x} en la vecindad de la singularidad) se enfatizó este, el caso más sencillo de este tipo, independiente del tiempo.

En este problema

$$p = 0, \quad q = 0, \quad f = \frac{\pi^2 \sin(\pi^2 y)}{2}$$

y las condiciones de frontera son $u(x, 0) = u(x, 1) = 0$

Los resultados descritos en la tabla 8 (ver también la gráfica 6) muestran que, al menos para las mallas estudiadas (que por problemas de tiempo de proceso no pueden ser mayores sin hacer un esfuerzo considerable para hacer el álgebra lineal de las matrices esparcidas de manera más eficiente) el método de elementos con puntos cuartos no parece ser muy bueno. Es peor que el método standard, contrariamente al resultado encontrado en una dimensión (el que esperábamos se extendiera con relativa facilidad a dos dimensiones).

Parece que efectivamente se puede atribuir el comportamiento relativamente malo de este método a que la malla modificada para el punto cuarto resulta asimétrica en x (como habíamos observado).

Aunque este problema no depende de x , la aproximación si; supongamos que el error E tiene dos componentes: El error introducido

por esta asimetría en x , es decir E_x , y el debido a la singularidad en y es decir E_y . No es posible tampoco olvidar el efecto del truncamiento, pero en este caso no parece ser importante.

Debido a que, como ya vimos en una dimensión (y en teoría el comportamiento es semejante) E_y disminuye al mejorar la aproximación a \sqrt{y} cerca de la singularidad por usar el método del punto cuarto; sin embargo es posible que para estas mallas E_x sea grande y oscurezca la mejora en la aproximación a \sqrt{y} .

Todo esto es válido tanto para el error máximo como para el error L2. En una dimensión, un problema comparable era el problema 4; en ese caso los puntos cuartos no mejoraron la solución cuando $NI=5$ pero sí con $NI=10$. Si $NI=20$ nuestra aproximación era mejor, pero ya con $NI=40$, el efecto se hacía menos notorio debido a que la iteración ya no convergía (en el tiempo de proceso permitido en la máquina 87800).

En dos dimensiones los errores son bastante mayores, sobre todo cuando $NJ=5$, aunque las cosas mejoran bastante cuando $NJ=10$, por lo que resulta bastante probable que para mallas más finas en y el análisis asintótico del error resulte válido y que los elementos con puntos cuartos mejoren la aproximación. (hay que aclarar que a pesar de que el método no mejoró la aproximación e inclusive la empeoró, seguimos teniendo soluciones aproximadas, es decir, no se observan perturbaciones tan terribles que destruyan la forma de la solución en forma global). Este estudio queda pendiente, pues es necesario contar con rutinas más eficientes para la factorización de matrices con una estructura esparcida muy particular.

Para probar el funcionamiento de el programa para problemas en dos dimensiones, se hizo una selección con un criterio idéntico al usado para los problemas unidimensionales, es decir se escogieron casos representativos. Para estos casos veamos cuanto valen p , q , f y las condiciones de frontera para

$$\frac{\partial u}{\partial t} = \nabla \cdot (p \nabla u) - q u + f$$

que es nuestro problema a resolver en general; tenemos entonces:

1) un problema independiente del tiempo (problema 1), lineal, no singular, con una solución $u = \sin(\pi x)$ donde $p=1, q=0, f = \pi^2 \sin(\pi x)$ y $u(0, y) = u(1, y) = 0$.

No se espera que los métodos estudiados impliquen una mejoría sobre el método standard.

2) un problema independiente del tiempo (problema 2), lineal, no singular, con una solución $u = \sin(\pi y)$ donde $p=1, q=0, f = \pi^2 \sin(\pi y)$ y $u(x, 0) = u(x, 1) = 0$.

Tampoco se espera que los métodos estudiados impliquen una mejoría sobre el método standard. Se seleccionaron estos dos problemas equivalentes para verificar que el programa no introduzca efectos asimétricos con respecto a x o y .

3) un problema dependiente del tiempo (problema 3), singular, que es de la forma que nos interesa, es decir con una solución singular de la forma \sqrt{t} específicamente:

$$u = e^{-t} \left(1 + \frac{1}{2} (\cos(\pi x)) \sqrt{\sin(\pi y)} \right)$$

donde $p = u, q = 1, f = \frac{\pi^2}{8} e^{-2t} \sin(\pi y) (2 + 8 \cos(\pi x) + 5 \cos^2(\pi x))$

$$y \quad u(x,0) = u(x,1) = 0$$

Aquí si se espera que los métodos estudiados impliquen una mejoría sobre el método standard debido a la presencia de la singularidad.

También se tiene un problema similar pero independiente del tiempo (problema 6) donde

$$u = (1 + \frac{1}{2} \cos(\pi x)) \sqrt{\sin(\pi y)}$$

$$p = u, \quad q = 0, \quad f = \pi^2 \sin(\pi y) \frac{1}{8} (2 + 8 \cos(\pi x) + 5 \cos^2(\pi x))$$

con iguales condiciones de frontera en el cual se espera un comportamiento similar.

4) un problema dependiente del tiempo (problema 4), no-lineal, singular y con una solución no singular

$$u = (1 + \frac{1}{2} \cos(\pi x)) \sin(\pi y) / (t - t_0)$$

donde

$$t_0 = -1, \quad p = u, \quad q = \sqrt{(t - t_0)}, \quad f = -(a + b + (c+d)e) / (t - t_0)$$

$$y \quad a = (\pi \cos(\pi y) (1 + \frac{1}{2} \cos(\pi x)))^2$$

$$b = \frac{1}{2} \pi \sin(\pi y) \sin(\pi x)$$

$$c = -\pi \sin(\pi y) (1 + \frac{1}{2} \cos(\pi x))$$

$$d = -\frac{1}{2} \pi \sin(\pi y) \cos(\pi x)$$

$$e = \sin(\pi y) (1 + \frac{1}{2} \cos(\pi x))$$

$$y \quad \text{donde } u(x,0) = u(x,1) = 0$$

No se espera que los métodos mejoren al método standard.

También un problema similar pero independiente del tiempo (problema 7) donde $u = (1 + \frac{1}{2} \cos(\pi x)) \sin(\pi y)$ y donde $p = u$, $q = 0$, $f = -(a+b+(c+d)e)$ con los mismos valores de frontera y en el cual se espera un comportamiento similar.

5) un problema dependiente del tiempo (problema 5), lineal y con una solución

$$u = \sin(\pi y) \cos(\pi x) e^{-2\pi^2 t}$$

donde $p = 1$, $q = 0$, $f = 0$

$$u(x, 0) = u(x, 1) = 0$$

Por último un problema similar pero independiente del tiempo (problema 8) donde $u = \sin(\pi y) \cos(\pi x)$ y $p = 1$, $q = -2x^2$, $f = 0$ con iguales condiciones de frontera

Las siguientes tablas describen sucintamente estos problemas (tabla 9) y los resultados para el error máximo y el error cuadrático (respectivamente las tablas 10 y 11), además de la tabla 14 y la grafica 7 (que resumen la convergencia observada en problemas bidimensionales). De ellas se puede concluir:

1) Al refinar uniformemente la malla en y es decir al dejar crecer N_y , se puede decir que si

$$h = (y_f - y_p) / N_y$$

entonces el error disminuye en forma al menos proporcional a h^2 si la solución es función de y y no de x . Algo similar sucedría con las x si se refinara en ese sentido. Se escogió trabajar sobre y pues para las mallas usadas resultan sistemas de ecuaciones más pequeños al refinar en y que al refinar en x .

Todo esto sucede cuando la ecuación es lineal y sin singularidades, en este caso el segundo problema. Esto implica que es conveniente trabajar con mallas suficientemente finas en y cuando la solución depende de esta.

2) El método de refinamiento local se comporta tal como se esperaba, es decir, trae consigo un mejoramiento considerable, comparable con un refinamiento uniforme grande en aquellos casos en que hay una solución singular (e inclusive donde no la hay pero si hay una singularidad), y un mejoramiento poco significativo donde no suceda esto. La ventaja del refinamiento local sobre el refinamiento uniforme es que en aquellos casos en que ambos funcionan se obtienen resultados similares a menor costo.

3) El método de elementos con puntos cuartos, que era el que mas hubieramos deseado que funcionara debido a su costo relativamente bajo, no parece tener el efecto esperado para ninguno de los problemas aqui analizados.

Se puede usar un argumento similar, es decir la existencia de perturbaciones geometricas, al visto para

$$u = \sqrt{\sin(\pi y)}$$

en los casos que esperabamos que mejorara, es decir

$$u = e^{-t} \left(1 + \frac{1}{2} \cos(\pi x) \right) \sqrt{\sin(\pi y)}$$

y

$$u = \left(1 + \frac{1}{2} \cos(\pi x) \right) \sqrt{\sin(\pi y)}$$

que son los que tienen un comportamiento de la forma \sqrt{y} cuando $y \rightarrow 0$ y $\sqrt{1-y}$ cuando $y \rightarrow 1$.

En los demas casos no es sorprendente que empeore, siendo $u = \sin(\pi x)$ y $u = \sin(\pi y)$ buenos ejemplos de lo que esperabamos sucediera.

No solo no nos sirve tener \sqrt{r} representada de alguna manera en la base, pues ya no se tiene r^2 sino que ademas estamos introduciendo perturbaciones geometricas.

seccion 11.1

Resumen de convergencia

De las tablas y graficas anteriores se puede concluir que

1) Tal como lo predice la teoría encontramos ordenes de convergencia de (N^3) o más en aquellos problemas en que no hay soluciones singulares tanto para el error máximo como para el L2 (encontrando mas debido a los factores causantes de superconvergencia ya mencionados, i.e. el que la malla en que se calcula el error contenga en su mayoría nodos de los elementos finitos). La gráfica número 1 así como la tabla número 12 son muy claras en este aspecto. Ver tambien los problemas 2 y 3 segun la tabla 13.

2) Se confirma que en presencia de soluciones singulares el método original (es decir el que no toma medidas especiales en estos casos) tiene un orden de convergencia bastante bueno, aunque su precisión es menor a la de los metodos modificados. Vease la convergencia del problema 4 (metodo sin cambios) según la tabla 13.

3) Como era deseado, con los métodos modificados que toman en cuenta la existencia de la singularidad, cuando esta efectivamente existe, se recupera en parte la convergencia perdida. En las gráficas es bastante claro cuando tengo problemas como el anotado acerca de que el criterio usado para detener el método puede hacerlo prematuramente. La grafica numero 5 así como la parte acerca de la convergencia en el problema 4 (tabla 13), son claras en este aspecto.

4) Desde luego, cuando se aplican elementos con puntos cuartos y funciones base especiales donde no se necesitan se obtiene un orden de convergencia bastante menor. No ocurre lo mismo para refinamiento local (pues seguimos teniendo representaciones compatibles). Las gráficas 1,2,3 y 4 son ilustrativas en este aspecto. Ver también los problemas 1, 2 y 3 segun las tablas 12 y 13.

5) Los problemas bidimensionales 1, 2 y 'a' es decir $u = \sqrt{\sin(\pi y)}$ muestran claramente el comportamiento esperado cuando se refina la malla en y para una función que no depende de y , para una que depende exclusivamente de ella, y para una que depende exclusivamente de ella pero tiene una singularidad. Los otros problemas bidimensionales me indican que el hecho de depender de x y y simultáneamente introduce ruido (al menos para el caso de mallas tan gruesas en x ya que al refinar en esta dirección el error disminuye ligeramente). Esto es claro en las graficas 6 y 7.

El orden de convergencia que esperamos para casos donde no hay solución singular es de la forma

$$(\Delta x)^r + (\Delta y)^r$$

Los números en la tabla 14 solo toman en cuenta el cambio en y es decir suponen la forma $(\Delta y)^r$. Como la parte en x no cambia, los números solo tienen sentido si el problema no depende de x . Este es el caso del problema 2, donde si se observa un número parecido a 3 en la tabla. Lo único que puedo decir acerca de los demás problemas es que muestran una convergencia que parece consistente con lo esperado.

seccion 11.2 Conclusiones

De los desarrollos teóricos, los programas hechos y los resultados encontrados tanto para una como para dos dimensiones es conveniente anotar lo siguiente como lo mas importante, quedando ademas descrito de manera concisa en las tablas 14 y 15 (todo referido a los problemas que nos interesaban en un principio, es decir a los no-lineales con singularidades del tipo \sqrt{x} o similares, tanto dependientes de tiempo como independientes, unidimensionales y bidimensionales):

subseccion 11.2.1 El caso unidimensional

1) En los problemas unidimensionales, el método de refinamiento local es el mejor para mallas gruesas, en las que además es muy caro. Para mallas finas las diferencias entre este y el método de elementos con puntos cuartos así como el método con funciones base especiales dejan de ser importantes. Tanto el método de elementos con puntos cuartos como el método con funciones base especiales funcionan de la forma esperada (tanto en los casos independientes del tiempo como en los casos dependientes del tiempo), es decir producen una mejor aproximación en aquellos casos en que se tiene una solución singular de tipo \sqrt{x} que es el caso de interés.

Los métodos de elementos con puntos cuartos y funciones base especiales, son tan buenos como se esperaba (ya que tanto la solución como la aproximación se comportan en ese caso como \sqrt{x} cerca de las fronteras), recuperándose en parte la razón de convergencia de un problema suave. En general para mallas finas es mejor el método de elementos con puntos cuartos que el de funciones base especiales; para las mallas gruesas (esto también se observó en los problemas anteriores) la diferencia es menor pero a favor de el método de funciones base especiales, ya que si bien hay mas errores de redondeo para estas, para las mallas gruesas los errores de redondeo son poco importantes; por otra parte el tener una malla irregular produce errores grandes sobre todo en mallas gruesas. De hecho se tiene una malla no uniforme en el caso de elementos con puntos cuartos y no en el de las funciones base especiales.

2) Debido a que ninguna de las soluciones analíticas de los problemas escogidos tiene una expansión asintótica que contenga términos en \sqrt{x} el comportamiento del método de elementos con punto cuarto fue bueno aun en aquellos casos en que se esperaba que fuera muy malo pues no tenemos aproximaciones mejores, pero tampoco peores. Lo mismo ocurre con el método con funciones base especiales.

subseccion 11.2.2 El caso bidimensional

1) En cuanto al problema bidimensional 'a' donde $u = \sqrt{(x^2 + y^2)}$ puede decirse que a pesar de que en cuanto a orden magnitud del error se esperaba que los métodos modificados mejoraran al método standard esto no se pudo observar en el caso del método de puntos cuartos. Se dieron argumentos geométricos y referentes al grosor de la malla usada para explicar esta desviación del comportamiento esperado. Se concluye que es necesario probar mallas mas finas para poder decir algo mas definitivo.

2) En el caso bidimensional se tiene el comportamiento esperado en el caso del refinamiento local, al igual que en el caso unidimensional, hay una importante mejoría de la aproximación y de la razón de convergencia. Sin embargo, y esto es muy importante, el método de elementos con puntos cuartos no muestra el comportamiento que se esperaba: ya se discutieron posibles explicaciones, pero una verdadera respuesta solo podra obtenerse en una indagación mas detallada y haciendo pruebas extensas con mallas mas finas que las limitaciones del programa y la máquina no permitan hacer de manera inmediata.

Las medidas que deben tomarse para hacer una prueba mas satisfactoria del metodo de punto cuarto en el caso bidimensional son:

Necesitamos un programa que en vez de recalcular infinidad de veces ciertas cosas, como la malla, para ahorrar memoria, lo haga una sola vez y guarde los resultados, claro esta, a expensas de la memoria.

Ademas se deben resolver de manera mas eficiente (por ejemplo algún método iterativo) los sistemas de ecuaciones lineales (que son bastante grandes) en vez de tener que factorizar matrices enormes con la consiguiente perdida de tiempo. (En esto no me puda meter para no alargar aun mas el trabajo).

3) En ambos casos (uni y bi-dimensionales) se observó que al resolver problemas independientes del tiempo (es decir problemas elípticos) como si fueran parabólicos, no se puede asegurar (con los resultados obtenidos) si se está convergiendo a la solución como teóricamente se esperaba; sin embargo, hay que tener presente que con dos o tres pasos en el tiempo sería mucha suerte el que efectivamente observáramos tal convergencia de manera inmediata.

Es de esperarse que los resultados del análisis asintótico no resulten evidentes en los primeros pasos de tiempo (en que inclusive se pueden presentar oscilaciones grandes sin invalidar el análisis).

Se puede sin embargo ver que al menos no se tienen grandes oscilaciones ni se nos disparan los resultados. Para hacer un análisis mejor necesitamos correr el programa mucho más a lo largo del tiempo; otra vez nos enfrentamos a las limitaciones y soluciones arriba delineadas acerca de la ineficiencia del programa y las limitaciones de tiempo de procesamiento.

En resumen, aunque no toda la evidencia es del todo concluyente, es claro el camino para encontrar una solución a los detalles que quedan sin respuesta por el momento: Seguir por el mismo camino pero cuidando más la eficiencia en tiempo del programa.

BIBLIOGRAFIA (referencias directas)

- 1) England R., Hennart J.P. et.al: 2-dimensional finite element calculational methods for plasma diffusion in low beta devices. Nuclear Science Engineering: 64 132-140 (1977).
- 2) Hennart J.P., England R. et.al: A finite element model for plasma simulation in multipoles. International Journal for numerical methods in Engineering: vol. 14 pags. 1449-1460 (1979).
- 3) Whiteman J.R.: Numerical solution for steady state diffusion problems containing singularities. in vol. 2 of Gallagher R. H., Taylor C. & Zienkiewicz O.C. (EDS) Finite Element Methods in Flow Problems Wiley (1974).
- 4) Wait R.: Singular Isoparametric finite elements. J. Inst. Math. Applics. (1977) 20, 133-141.
- 5) Strang G. & Fix G.: An analysis of the finite element method. Prentice Hall (1973).
- 6) Mitchell A.R. & Wait R.: The finite element method in partial differential equations. Wiley (1977).
- 7) Tong P. and Rossettos J.N.: Finite Element Method: Basic Technique and Implementation. The MIT Press, Boston (1977).
- 8) Whiteman J.R., Akin J.E.: Finite Elements, Singularities and Fracture. MAFELAP III (1978) Edited by Whiteman J.R. Academic Press 1975
- 9) Barnhill R.E., and Whiteman J.R.: Error Analysis of Galerkin methods for Dirichlet problems containing boundary singularities. J. Inst. Math. Applics. 15, 121-125 (1975)

- 10) Papamichael N. and Whiteman J.R.: A numerical conformal transformation method for harmonic mixed boundary value problems in polygonal domains. Z. angew. Math Phys. 24,304-316(1973).
- 11) Elsgoltz L.: Ecuaciones diferenciales y cálculo variacional Editorial MIR Moscu 1977
- 12) Hennart J. P., Mund E. H.: Singularities in the Finite Element Approximation of Two-Dimensional Diffusion Problems Nuclear Science and Engineering:62, 55-68(1977)
- 13) Krasnov M. L. , Makarenko C I. , Kiseliou A. I.: Cálculo Variacional (Ejemplos y Problemas) MIR (1976) Moscu.
- 14) Gram J.G.: Numerical solution of Partial differential equations Proceedings of NATO Advanced Study Institute, Kjeller, Norway, August 20-24 1973 see articles by Collatz L. & Mitchell A.R.
- 15) Wait R.: Finite Element methods for elliptic problems with singularities. Internal Communications, Department of Computational and Statistical Science, University of Liverpool, England.
- 16) Whitemann J.R.: A Bibliography for Finite Elements, Academic Press (1975) London.

Bibliografía

Bibliografía adicional consultada:

- 1) Oden J.T., Zienkiewicz D.C., Gallagher R.H., Taylor C. (ed).
Finite Element Methods in Flow Problems
UAH Press (1974)
 - i) Whiteman J.R.
Numerical Solution of Steady State Diffusion Problems Containing Singularities.
 - ii) Oden J.T., Wellford Jr. L. C.
Mathematical features of Finite Element approximation of certain flow problems with emphasis on convection and diffusion.
 - iii) Hutton A.G.
On flow near Singular points of a wall boundary.
 - iv) Apostol R.T., Charlton J.A.
Solution of a two dimensional diffusion-convection equation describing mass transfer of a neutrally buoyant tracer.
- 2) Oden J.T., Zienkiewicz D.C., Gallagher R.H., Taylor C. (ed).
Finite Element Methods in Fluids
Wiley (1975)
 - i) Zienkiewicz D.C.
Why Finite Elements?
 - ii) Oden J.T., Wellford Jr. L. C.
Accuracy and Convergence of Finite Element Galerkin approximation of time dependent problems with emphasis on diffusion.
 - iii) Whiteman J.R.
Numerical Solution of Steady State Diffusion Problems Containing Singularities.

Bibliografia

- 3) Oden J. T.
Finite Elements of nonlinear continua
Mc-Graw-Hill (1972)
- 4) Lederman W., Churchouse R. F.,
Handbook of Applicable Mathematics, Volume III, Numerical Methods
Wiley (1981)
- 5) Bennet A. W., Vichnevetsky R. (ed)
Numerical Methods for Differential Equations and Simulation,
proceedings of IMACS(AICA)
North Holland (1978)
 - i) Baker A. J., Soliman M. O.
On the accuracy and convergence of implicit numerical integration
of finite element generated ordinary differential equations.
 - ii) Csendes Z. J.
A novel finite element method for two point boundary value
problems.
 - iii) Hulsey J. L., Emanuel J. H.
Finite Element modelling of climatically induced heat flow.
- 6) De Boor Carl (ed)
Mathematical aspects of Finite Elements in Partial Differential
Equations.
proceedings of a symposium conducted by the Mathematics Research
Center, The University of Wisconsin, Madison April 1-3 1974
Academic Press (1974)
 - i) Thomee V
Some Convergence Results for Galerkin Methods for Parabolic
Boundary problems.
 - ii) Babuska I
Solution of problems with interfaces and singularities.

Bibliografia

iii) Swartz B.K.

The construction and comparison of finite difference analogs of some finite element schemes.

iv) Dupont T.

L2 Error estimates for projection methods for parabolic equations in Approximating Domains.

7) Ames W.F.

Numerical Methods for Partial Differential Equations.
Academic Press (1977)

8) De Boor C., Golub G.H.,

Recent advances in numerical Analysis
Academic Press (1978)

1) Nitsche J.A.

Finite Element Approximation to one dimensional Stefan Problem

9) Temam R.

Numerical Analysis
Reidel Publishing Company
Dordrecht Holland (1973).

10) Grove W.E.

Brief numerical methods.
Prentice Hall (1966)

11) Bramble J.H. (ed)

Numerical solution of Partial Differential Equations.
proceedings Symposium College Park Md. 1965
Academic Press (1966)

1) Varga R.

Hermite Interpolation-Type Ritz Methods for two point boundary value problems.

Bibliografia

- 12) Bliss G. A.
Lectures on the Calculus of Variations.
Phoenix Books (1948 printed 1968)
- 13) Stoer J., Bulirsch R.
Introduction to Numerical Analysis.
Springer Verlag (1980)
- 14) Henrici P.
Elements of Numerical Analysis.
Wiley (1964)
- 15) Blum E. K.
Numerical Analysis and Computation: theory and practice.
Addison Wesley (1972)
- 16) Aziz A. K. (ed)
The Mathematical Foundations of the Finite Element Method with
Applications to Partial Differential Equations.
Academic Press (1972)
- i) Babuska I. and Aziz A. K.
Survey Lectures on the Mathematical Foundations of the Finite
Element Method.
- ii) Ciarlet P. G. and Raviart P. A.,
The combined effect of curved boundaries and Numerical Integration
in Isoparametric Finite Element Methods.
- iii) Dupont T.
Some L2 error estimates for Parabolic Galerkin Methods.
- iv) Eisenstat S. C., Schultz M. H.
Computational Aspects of the Finite Element Method.
- v) Fix G. J.
Effects of Quadrature Errors in Finite Element Approximation of
Steady State, Eigenvalue and Parabolic Problems.

Bibliografia

- vi) Strang G.
Variational Crimes in the Finite Element Method.

- 17) Hubbard B.
Numerical Solution of Partial Differential Equations -II
Academic Press (1971)

- i) Babuska I.
The Finite Element Method for Elliptic Differential Equations.

- ii) Marcal P.V.
On General Purpose Programs for Finite Element Analysis, with
special reference to geometric and material nonlinearities.

- iii) Strang G.
The Finite Element Method and Approximation theory.

Descripción general de las tablas y gráficas.

La tabla 0 contiene la descripción de los problemas unidimensionales independientes del tiempo. Las tablas 1 a 5 contienen los errores (máximo y L_2) encontrados para estos problemas. Estos errores encontrados para cada método con distintas mallas para elemento finito (mallas asociadas a NI elementos), usando una malla uniforme de $NIMP$ puntos para calcularlos, son tabulados contra el valor de NI .

La tabla 12 contiene información acerca de la proporción de los errores entre mallas cada vez más finas, o sea el orden de convergencia observado, es decir las pendientes de las rectas que resultarían al graficar estos datos en escala log-log. Estos resultados fueron obtenidos a partir de los datos presentados en las tablas anteriores (es decir 1 a 5) de la siguiente manera: si el error asociado a la malla que tiene $NI=5$ es E_1 y el asociado a $NI=10$ es E_2 , entonces en el renglón marcado 5-10 pongo $(1/(L_2) \ln(E_1/E_2))$. Se procede de igual manera para las demás mallas.

De las tablas 1 a 5 se obtuvieron además las gráficas 1 a 5. Son simplemente las mismas tablas presentadas de manera gráfica, con escalas log-log, las que nos dan la misma información que la tabla 12. Se incluyeron pues facilitan el extraer conclusiones.

Para el caso parabólico unidimensional se utilizaron las tablas 6 y 7. En la primera se resume los problemas probados (es decir los problemas 5 a 8). La tabla 7 resume los resultados encontrados para una malla de elementos finitos de $NI=10$, a lo largo del tiempo. Se utilizó la tabla 7 se presenta de manera gráfica en la gráfica 8 (utilizo escalas distintas para cada problema).

En cuanto a los problemas bidimensionales, tenemos la tabla 8, que analiza el problema a para todos los métodos y para varias mallas de elemento finito. La tabla 13 y la gráfica 6 complementan esta información tal como teníamos para una dimensión. La tabla 9 describe los demás problemas bidimensionales estudiados; los resultados están contenidos en las tablas 10 y 11. Complementa la información la gráfica 7 y la tabla 13 (con los grados de convergencia observados).

Las tablas 14 y 15 resumen las conclusiones obtenidas.

Descripción de los problemas independientes del tiempo en una dimensión.

num.	u	p	q	f	descripción
1	$\sin(\pi x)$	1	0	$\pi^2 \sin(\pi x)$	L-NS-NM
2	$\sin(\pi x)$	$1-u^2$	0	$\pi^2 \sin(\pi x) (1-3 \sin^2(\pi x))$	NL NS-NM
3	$\sin(\pi x)$	u	0	$-\pi^2 (1-2 \sin^2(\pi x))$	NL-S-NM
4	$\sqrt{2 \sin(\pi x)}$	u	0	$\pi^2 u^2 / 2$	L-S M

Si la descripción incluye los siguientes símbolos, significa que el problema tiene las siguientes características: L es lineal NL no-lineal, NS es problema no-singular y S singular; M implica que las soluciones tienen singularidad por tanto se espera que los resultados mejoraran con los metodos especiales; NM implica soluciones no singulares. u es la solución analítica, p, q, f los coeficientes en la ecuación.

Descripción de los problemas independientes del tiempo en una dimensión

Tabla 0

Problema uno Método uno Error Máximo				
$\Delta t = 0.1$				
tiempo	$N_I = 5$	$= 10$	$N_I = 20$	$N_I = 40$
$t = 0.0$	1.840×10^{-3}	4.604×10^{-6}	2.985×10^{-7}	1.924×10^{-8}
$t = 0.1$	1.845×10^{-3}	2.009×10^{-6}	1.200×10^{-7}	8.442×10^{-9}
$t = 0.2$	1.842×10^{-3}	2.882×10^{-6}	1.852×10^{-7}	1.313×10^{-8}
$t = 0.3$	1.842×10^{-3}	2.683×10^{-6}	1.205×10^{-7}	6.005×10^{-9}
$\Delta t = 0.05$				
tiempo	$N_I = 5$	$N_I = 10$	$N_I = 20$	$N_I = 40$
$t = 0.00$	1.840×10^{-3}	4.604×10^{-6}	2.985×10^{-7}	1.924×10^{-8}
$t = 0.05$	1.842×10^{-3}	4.692×10^{-6}	3.059×10^{-7}	1.950×10^{-8}
$t = 0.10$	1.845×10^{-3}	2.403×10^{-6}	1.395×10^{-7}	8.004×10^{-9}
$t = 0.15$	1.844×10^{-3}	3.197×10^{-6}	2.082×10^{-7}	1.378×10^{-8}
$t = 0.20$	1.843×10^{-3}	2.044×10^{-6}	1.009×10^{-7}	1.037×10^{-8}
$t = 0.25$	1.843×10^{-3}	2.694×10^{-6}	1.717×10^{-7}	1.196×10^{-8}
$t = 0.30$	1.843×10^{-3}	2.512×10^{-6}	1.544×10^{-7}	1.098×10^{-8}

Tabla 1.
Error Máximo para el problema uno utilizando el método uno

Problema uno Método uno				
Error L_2				
$\Delta t = 0.1$				
tiempo	$N_I = 5$	$N_I = 10$	$N_I = 20$	$N_I = 40$
$t = 0.0$	0.355×10^{-4}	2.059×10^{-6}	2.060×10^{-7}	1.335×10^{-8}
$t = 0.1$	0.376×10^{-4}	1.352×10^{-6}	8.270×10^{-8}	5.884×10^{-9}
$t = 0.2$	0.372×10^{-4}	1.638×10^{-6}	1.278×10^{-7}	8.143×10^{-9}
$t = 0.3$	0.377×10^{-4}	1.670×10^{-6}	8.935×10^{-8}	6.054×10^{-9}

$\Delta t = 0.05$				
tiempo	$N_I = 5$	$N_I = 10$	$N_I = 20$	$N_I = 40$
$t = 0.00$	0.355×10^{-4}	2.059×10^{-6}	2.060×10^{-7}	1.335×10^{-8}
$t = 0.05$	0.350×10^{-4}	2.610×10^{-6}	2.111×10^{-7}	1.352×10^{-8}
$t = 0.10$	0.371×10^{-4}	1.432×10^{-6}	0.629×10^{-7}	6.262×10^{-9}
$t = 0.15$	0.300×10^{-4}	1.922×10^{-6}	1.437×10^{-7}	6.623×10^{-9}
$t = 0.20$	0.371×10^{-4}	1.689×10^{-6}	1.110×10^{-7}	7.580×10^{-9}
$t = 0.25$	0.372×10^{-4}	1.779×10^{-6}	1.185×10^{-7}	6.320×10^{-9}
$t = 0.30$	0.374×10^{-4}	1.736×10^{-6}	1.005×10^{-7}	7.003×10^{-9}

Tabla 2.
Error L_2 para el problema uno utilizando el método uno

Problema uno por todos los métodos

Error Máximo				
$\Delta t = 0.1, t = 0.3$				
Malla	sin cambios	punto cuarto	refinamiento local	funciones especiales
$NI = 5$	1.842×10^{-3}	1.181×10^{-3}	1.168×10^{-3}	1.205×10^{-3}
$NI = 10$	2.663×10^{-6}	1.511×10^{-6}	2.681×10^{-6}	1.499×10^{-6}
$NI = 20$	1.205×10^{-7}	1.750×10^{-7}	1.293×10^{-7}	5.918×10^{-8}
$NI = 40$	0.995×10^{-9}	4.076×10^{-9}	0.721×10^{-9}	3.440×10^{-7}
$\Delta t = 0.05, t = 0.3$				
Malla	sin cambios	punto cuarto	refinamiento local	funciones especiales
$NI = 5$	1.843×10^{-3}	1.180×10^{-3}	1.170×10^{-3}	1.170×10^{-3}
$NI = 10$	2.512×10^{-6}	1.526×10^{-6}	2.529×10^{-6}	1.533×10^{-6}
$NI = 20$	1.544×10^{-7}	0.388×10^{-7}	1.541×10^{-7}	2.382×10^{-6}
$NI = 40$	1.098×10^{-9}	2.583×10^{-9}	1.137×10^{-9}	1.805×10^{-7}
Error L_2				
$\Delta t = 0.1, t = 0.3$				
Malla	sin cambios	punto cuarto	refinamiento local	funciones especiales
$NI = 5$	0.377×10^{-4}	5.330×10^{-5}	4.936×10^{-4}	5.464×10^{-5}
$NI = 10$	1.670×10^{-6}	4.665×10^{-4}	1.671×10^{-6}	4.835×10^{-4}
$NI = 20$	8.935×10^{-8}	7.050×10^{-7}	3.923×10^{-8}	2.008×10^{-6}
$NI = 40$	6.954×10^{-9}	2.200×10^{-8}	0.696×10^{-9}	3.446×10^{-7}
$\Delta t = 0.05, t = 0.3$				
Malla	sin cambios	punto cuarto	refinamiento local	funciones especiales
$NI = 5$	0.374×10^{-4}	5.327×10^{-5}	4.935×10^{-4}	5.363×10^{-5}
$NI = 10$	1.736×10^{-6}	4.710×10^{-4}	1.736×10^{-6}	4.740×10^{-4}
$NI = 20$	1.065×10^{-7}	5.305×10^{-7}	1.063×10^{-7}	1.203×10^{-6}
$NI = 40$	7.663×10^{-9}	2.042×10^{-8}	7.842×10^{-9}	0.325×10^{-6}

Tabla 3.
Error máximo y L_2 para el problema uno utilizando los cuatro métodos

Problema dos por todos los métodos
Error Máximo y L_2

$$\Delta t = 0.1, t = 0.3$$

Error máximo

Malla	sin cambios	punto cuarto	refinamiento local	funciones especiales
$NI = 5$	1.071×10^{-2}	1.108×10^{-2}	1.480×10^{-2}	1.050×10^{-2}
$NI = 10$	4.028×10^{-3}	1.402×10^{-2}	4.041×10^{-2}	1.447×10^{-2}
$NI = 20$	2.723×10^{-3}	1.106×10^{-2}	2.760×10^{-2}	2.375×10^{-2}
$NI = 40$	1.973×10^{-3}	1.170×10^{-2}	1.000×10^{-2}	1.017×10^{-2}

Error L_2

Malla	sin cambios	punto cuarto	refinamiento local	funciones especiales
$NI = 5$	1.003×10^{-2}	5.280×10^{-2}	0.114×10^{-1}	5.470×10^{-2}
$NI = 10$	2.773×10^{-3}	4.803×10^{-2}	2.795×10^{-2}	4.510×10^{-2}
$NI = 20$	1.756×10^{-3}	0.007×10^{-1}	1.790×10^{-2}	1.121×10^{-2}
$NI = 40$	1.070×10^{-3}	0.851×10^{-2}	1.102×10^{-2}	1.118×10^{-2}

Tabla 4.

Errores máximo y L_2 problema dos utilizando los cuatro métodos

Error L_2				
$\Delta t = 0.1$, $t = 0.3$				
Problema tres por todos los métodos				
Malla	sin cambios	punto cuarto	refinamiento local	funciones especiales
$NI = 5$	1.042×10^{-3}	4.240×10^{-3}	5.431×10^{-4}	2.877×10^{-3}
$NI = 10$	3.836×10^{-5}	2.059×10^{-4}	1.652×10^{-5}	2.158×10^{-4}
$NI = 20$	1.102×10^{-6}	1.400×10^{-6}	1.042×10^{-6}	1.443×10^{-6}
$NI = 40$	6.526×10^{-8}	6.283×10^{-8}	6.856×10^{-8}	1.590×10^{-7}
Problema cuatro por todos los métodos				
Malla	sin cambios	punto cuarto	refinamiento local	funciones especiales
$NI = 5$	2.428×10^{-2}	2.397×10^{-2}	2.804×10^{-4}	2.226×10^{-2}
$NI = 10$	1.103×10^{-2}	2.802×10^{-4}	2.533×10^{-5}	2.753×10^{-4}
$NI = 20$	1.425×10^{-3}	3.600×10^{-5}	6.505×10^{-6}	1.418×10^{-5}
$NI = 40$	3.936×10^{-5}	2.684×10^{-6}	6.735×10^{-7}	1.800×10^{-5}

Tabla 5.
Error L_2 para problemas tres y cuatro utilizando los cuatro métodos

Descripción de los problemas
dependientes de tiempo en una dimensión

$u_{analítica}$	p	q	f	descripción	prob. num.
$\sin(\pi x) \cdot e^{-x^2 t}$	1	0	0	L-NS-NM	5
$\sin(\pi x) \cdot e^{-x^2 t}$	$1 + u^2$	$2\pi^2 \cos(\pi x) e^{-x^2 t}$	$\pi^2 u^3$	NL-NS-NM	0
$\sin(\pi x) / (\pi^2 (t - t_0)^2)$	π	$1 / (t - t_0)$	$-\cos(2\pi x) / (\pi(t - t_0)^2)$	NL-S-NM	7
$\sqrt{\sin(\pi x)} u^{-t}$	π	1	$(\pi u)^2 / 2$	NL-S-M	8

Aquí $t_0 = -1$, L es lineal NL, no-lineal, NS es no-singular S singular.

Si las singularidades son de tipo $\sqrt{\sin(\pi y)}$ puse M, lo que implica que se espera que mejorará con los métodos vistos y NM que no;

Tabla 6.
Descripción de los problemas cinco a ocho en una dimensión.

Descripción de los problemas dependientes del tiempo en una dimensión.

	μ	p	q	f	descripción
5	$\sin(\pi x) e^{-\pi^2 t}$	1	0	0	L-US-NM
6	$\sin(\pi x) e^{-\pi^2 t}$	$1+\mu^2$	$2\pi^2 \cos^2(\pi x) e^{-2\pi^2 t}$	$\pi^2 \mu^3$	NL-NS-NM
7	$\frac{\sin(\pi x)}{\pi^2 (t-t_0)^2}$	μ	$\frac{1}{(t-t_0)}$	$\frac{-\cos(2\pi x)}{\pi (t-t_0)^2}$	NL-S-NM
8	$\sqrt{\sin(\pi x)} e^{-t}$	μ	1	$\frac{1}{2} \pi^2 \mu^2$	L-S-M

Aquí $t_0 = -1$

Además, si la descripción incluye los siguientes símbolos, significa que el problema tiene las siguientes características: L es lineal, NL no-lineal, NS es problema no-singular y S singular; M implica que las soluciones tienen singularidad por tanto se espera que los resultados mejorarán con los métodos especiales; NM implica soluciones no singulares. μ es la solución analítica, p, q, f los coeficientes en la ecuación.

Descripción de los problemas dependientes del tiempo en una dimensión
Tabla 6

Problemas cinco a ocho todos los Métodos

Error L_2

$$\Delta t = 0.1, n_i = 10, t = 0.3$$

Problema cinco				
tiempo	sin cambios	punto cuarto	refinamiento local	funciones especiales
$t = 0.0$	9.120×10^{-4}	1.024×10^{-3}	9.124×10^{-4}	1.102×10^{-3}
$t = 0.1$	1.853×10^{-3}	1.071×10^{-3}	1.853×10^{-3}	1.691×10^{-3}
$t = 0.2$	2.248×10^{-3}	2.257×10^{-3}	2.240×10^{-3}	2.260×10^{-3}
$t = 0.3$	2.717×10^{-3}	2.713×10^{-3}	2.718×10^{-3}	2.729×10^{-3}
Problema seis				
tiempo	sin cambios	punto cuarto	refinamiento local	funciones especiales
$t = 0.0$	9.763×10^{-4}	1.082×10^{-3}	9.782×10^{-4}	1.158×10^{-3}
$t = 0.1$	1.821×10^{-3}	1.034×10^{-3}	1.821×10^{-3}	1.850×10^{-3}
$t = 0.2$	2.525×10^{-3}	2.540×10^{-3}	2.525×10^{-3}	2.563×10^{-3}
$t = 0.3$	3.098×10^{-3}	3.102×10^{-3}	3.098×10^{-3}	3.119×10^{-3}
Problema siete				
tiempo	sin cambios	punto cuarto	refinamiento local	funciones especiales
$t = 0.0$	2.353×10^{-6}	2.831×10^{-6}	2.190×10^{-6}	2.974×10^{-6}
$t = 0.1$	4.370×10^{-6}	2.779×10^{-6}	4.223×10^{-6}	5.593×10^{-6}
$t = 0.2$	6.309×10^{-6}	2.744×10^{-6}	6.180×10^{-6}	7.000×10^{-6}
$t = 0.3$	8.163×10^{-6}	2.724×10^{-6}	8.055×10^{-6}	1.013×10^{-5}
Problema ocho				
tiempo	sin cambios	punto cuarto	refinamiento local	funciones especiales
$t = 0.0$	1.124×10^{-3}	2.700×10^{-4}	3.943×10^{-6}	2.806×10^{-4}
$t = 0.1$	1.053×10^{-3}	2.720×10^{-4}	7.389×10^{-6}	2.623×10^{-4}
$t = 0.2$	1.123×10^{-3}	2.744×10^{-4}	6.630×10^{-6}	2.729×10^{-4}
$t = 0.3$	1.128×10^{-3}	2.752×10^{-4}	7.536×10^{-6}	2.722×10^{-4}

Tabla 7.

Error L_2 para los problemas cinco a ocho

$$\text{Problema } u = \sqrt{\sin(\pi y)}$$

$$\Delta t = 0.001, t = 0.001$$

Error Máximo				
Red (M, N)	(3, 5)	(6, 5)	(3, 10)	(6, 10)
solo (x+d)	.13008030	.12382007	.10303337	88434151×10^{-1}
cuartos	.29066412	.31302556	.21200853	.21100703
refinamiento	$.018522255 \times 10^{-1}$	$.712256240 \times 10^{-1}$	$.436110450 \times 10^{-1}$	$.315000310 \times 10^{-1}$

Error L_2				
Red (M, N)	(3, 5)	(6, 5)	(3, 10)	(6, 10)
solo (x+d)	$.42780070 \times 10^{-1}$	$.41361471 \times 10^{-1}$	$.22492171 \times 10^{-1}$	$.20165301 \times 10^{-1}$
cuartos	$.72492275 \times 10^{-1}$	$.84652047 \times 10^{-1}$	$.30608348 \times 10^{-1}$	$.42502088 \times 10^{-1}$
refinamiento	$.13385706 \times 10^{-1}$	$.07380623 \times 10^{-2}$	$.75370200 \times 10^{-2}$	$.48407014 \times 10^{-2}$

Tabla 8.
 Problema $u = \sqrt{\sin(\pi y)}$

Descripción de los problemas en dos dimensiones.

núm.	μ	p	q	f	Descripción
1	$\sin(\pi x)$	1	0	$\pi^2 \sin(\pi x)$	I-E-L-US-NA
2	$\sin(\pi y)$	1	0	$\pi^2 \sin(\pi y)$	I-E-L-NS-NA
3	$e^t (1 + \frac{1}{2} \cos(\pi x)) \sqrt{\sin(\pi y)}$	μ	1	$\pi^2 e^{-2t} \sin(\pi y) F/g$	D-P-NL-S-M
4	$\frac{(1 + \frac{1}{2} \cos(\pi x)) \sin(\pi y)}{(t - t_0)}$	μ	$1/(t - t_0)$	$\frac{-(a + b + (c + d)e)}{(t - t_0)}$	D-P-NL-S-NA
5	$\sin(\pi y) \cos(\pi x) e^{-2\pi^2 t}$	1	0	0	D-P-L-NS-NA
6	$(1 + \frac{1}{2} \cos(\pi x)) \sqrt{\sin(\pi y)}$	μ	0	$\pi^2 \sin(\pi y) F/g$	I-E-NL-S-M
7	$(1 + \frac{1}{2} \cos(\pi x)) \sin(\pi y)$	μ	0	$-(a + b + (c + d)e)$	I-E-NL-S-NA
8	$\sin(\pi y) \cos(\pi x)$	1	$-2\pi^2$	0	I-E-L-US-NA

Aquí $t_0 = -1$ $F = 2 + 8 \cos(\pi x) + 5 \cos^2(\pi x)$

$a = (\pi \cos(\pi y) (1 + \frac{1}{2} \cos(\pi x)))^2$ $d = -\pi^2 \sin(\pi y) \cos(\pi x) / 2$

$b = (\frac{\pi \sin(\pi y) \sin(\pi x)}{2})^2$ $e = \sin(\pi y) (1 + \frac{1}{2} \cos(\pi x))$

$c = -\pi^2 \sin(\pi y) (1 + \frac{1}{2} \cos(\pi x))$

Además, si la descripción incluye los siguientes símbolos, significa que el problema tiene las siguientes características: I es independiente del tiempo, D dependiente; E es elíptico P parabólico, L es lineal NL no-lineal, NS es problema no-singular y S singular. M implica que las soluciones tienen singularidad por tanto se espera que los resultados mejorarán con los métodos especiales; NA implica soluciones no singulares. Mas la solución analítica, e, q, f los coeficientes en la ecuación.

Descripción de los problemas en dos dimensiones

Tabla 9

Problemas en dos dimensiones.

$$\Delta t = .001, t = .001$$

Error Máximo				
Prob.	NI= 3, NJ= 5	NI= 3, NJ= 10	refinamiento	cuartos
1	$.84897376 \times 10^{-2}$	$.86039533 \times 10^{-2}$	$.70531781 \times 10^{-3}$	$.34071451 \times 10^{-1}$
2	$.14068035 \times 10^{-2}$	$.19276979 \times 10^{-3}$	$.16414939 \times 10^{-2}$	$.59481544 \times 10^{-2}$
3	$.74050805 \times 10^{-1}$	$.18686150 \times 10^{-1}$	$.16133378 \times 10^{-1}$	$.04115059 \times 10^{-1}$
4	.41859357	.23484619	.23206010	.23214772
5	$.14398218 \times 10^{-1}$	$.57098102 \times 10^{-2}$	$.13972937 \times 10^{-1}$	$.01851040 \times 10^{-2}$
6	$.75161831 \times 10^{-1}$	$.10376605 \times 10^{-1}$	$.10244658 \times 10^{-1}$	$.01866053 \times 10^{-1}$
7	$.75610990 \times 10^{-1}$	$.19537958 \times 10^{-1}$	$.15835930 \times 10^{-1}$	$.90311238 \times 10^{-1}$
8	$.15141368 \times 10^{-1}$	$.50613895 \times 10^{-2}$	$.14687788 \times 10^{-1}$	$.10094493 \times 10^{-1}$

Tabla 10.
Problemas en dos dimensiones; Error Máximo.

Problemas en dos dimensiones.

$$\Delta t = .001, t = .001$$

Error L_2				
Prob.	NI= 3, NJ= 5	NI= 3, NJ= 10	refinamiento	cuartos
1	$.40107429 \times 10^{-2}$	$.46220010 \times 10^{-2}$	$.28406452 \times 10^{-3}$	$.11739792 \times 10^{-1}$
2	$.77478455 \times 10^{-3}$	$.67108059 \times 10^{-4}$	$.75755960 \times 10^{-3}$	$.20183379 \times 10^{-2}$
3	$.16937067 \times 10^{-1}$	$.60298054 \times 10^{-2}$	$.53725574 \times 10^{-2}$	$.24693678 \times 10^{-1}$
4	.11391684	$.63259449 \times 10^{-1}$	$.83977569 \times 10^{-1}$	$.62538786 \times 10^{-1}$
5	$.46171801 \times 10^{-2}$	$.35013441 \times 10^{-2}$	$.45038197 \times 10^{-2}$	$.30431123 \times 10^{-2}$
6	$.17053921 \times 10^{-1}$	$.60724878 \times 10^{-2}$	$.54059374 \times 10^{-2}$	$.24847221 \times 10^{-1}$
7	$.17455610 \times 10^{-1}$	$.65233709 \times 10^{-2}$	$.61144734 \times 10^{-2}$	$.20648045 \times 10^{-1}$
8	$.47783356 \times 10^{-2}$	$.22820189 \times 10^{-2}$	$.47074300 \times 10^{-2}$	$.41287132 \times 10^{-2}$

Tabla 11.
Problemas en dos dimensiones; Error L_2 .

Convergencia
Problema uno por todos los métodos

Error M ^{aximo} $\Delta t=0.1, t=0.3$				
Mallas sin cambios		p. cuarto	ref. local	f. esp.
NI=5 → 10	9.434	2.976	6.747	3.007
NI=10 → 20	4.342	9.746	4.373	7.984
NI=20 → 40	3.696	5.431	3.753	4.102
$\Delta t=0.05, t=0.3$				
Mallas sin cambios		p. cuarto	ref. local	f. esp.
NI=5 → 10	9.519	2.950	8.854	2.943
NI=10 → 20	4.024	11.222	4.036	9.529
NI=20 → 40	3.813	4.623	3.760	3.722
Error L ₂ $\Delta t=0.1, t=0.3$				
Mallas sin cambios		p. cuarto	ref. local	f. esp.
NI=5 → 10	9.116	3.878	8.854	2.943
NI=10 → 20	4.224	9.870	4.227	7.850
NI=20 → 40	3.693	3.002	3.736	2.543
$\Delta t=0.05, t=0.3$				
Mallas sin cambios		p. cuarto	ref. local	f. esp.
NI=5 → 10	9.076	3.499	8.151	3.500
NI=10 → 20	4.026	9.794	4.029	8.622
NI=20 → 40	3.797	4.699	3.760	4.249

Ordenes de convergencia observados para el problema uno utilizando los cuatro métodos

Tabla 12.

Convergencia

Problemas dos a cuatro

Error L ₂ $\Delta t=0.1$, $t=0.3$					
Problema 2					
Gallas sin cambios	p. cuarto	ref. local	f. esp.		
NI=5 \rightarrow 10	3.265	4.491	3.530		
NI=10 \rightarrow 20	3.981	3.757	3.656		
NI=20 \rightarrow 40	4.024	3.269	3.324		
Problema 3					
Gallas sin cambios	p. cuarto	ref. local	f. esp.		
NI=5 \rightarrow 10	4.743	3.937	3.734		
NI=10 \rightarrow 20	5.121	3.924	3.181		
NI=20 \rightarrow 40	4.077	3.926	3.181		
Problema 4					
Gallas sin cambios	p. cuarto	ref. local	f. esp.		
NI=5 \rightarrow 10	4.388	3.468	3.015		
NI=10 \rightarrow 20	4.351	-1.373	4.279		
NI=20 \rightarrow 40	-1.092	-0.037	-0.035		

Ordenes de convergencia observados para los problemas dos a cuatro

Convergencia Problemas en dos dimensiones

Error M^{aximo}

Problema

NJ=5 \rightarrow 10 (NI=6)

2	(m. regular)	.345
3	(p. cuartos)	.454
4	(ref.)	.917
5	(m. regular)	.003
6	(m. regular)	2.880
7	(m. regular)	1.923
8	(m. regular)	1.833
9	(m. regular)	1.312
10	(m. regular)	1.923
11	(m. regular)	1.956
12	(m. regular)	1.418

Ordenes de convergencia observados para los problemas en dos dimensiones

Tabla 13.

conclusiones:

problemas id

problema:	elementos regulares	puntos cuartos	resonancias local	funciones especiales
1				
2				
3				
4		M→M	MM→MM	M→M
5				
6				
7				
8		M→M	MM→MM	M→M

tabla 14

En esta tabla:

M→M indica que mejoró la aproximación a la solución tal como se esperaba.

MM→MM indica que mejoró mucho la aproximación a la solución tal como se esperaba.

En los demás no se esperaba mejora y así sucedió.

conclusiones:

problemas 2d

Problema:	elementos regulares	puntos cuartos	Refinamiento local
1		M->NM	M->M
2			
3		M->NM	M->M
4			
5			
6		M->NM	M->M
7			
8			

tabla 15

En esta tabla:

M->NM indica que mejoró la aproximación a la solución tal como se esperaba

M->NM indica que NO mejoró la aproximación a la solución tal como se esperaba.

En los demás no se esperaba mejora y así sucedió.

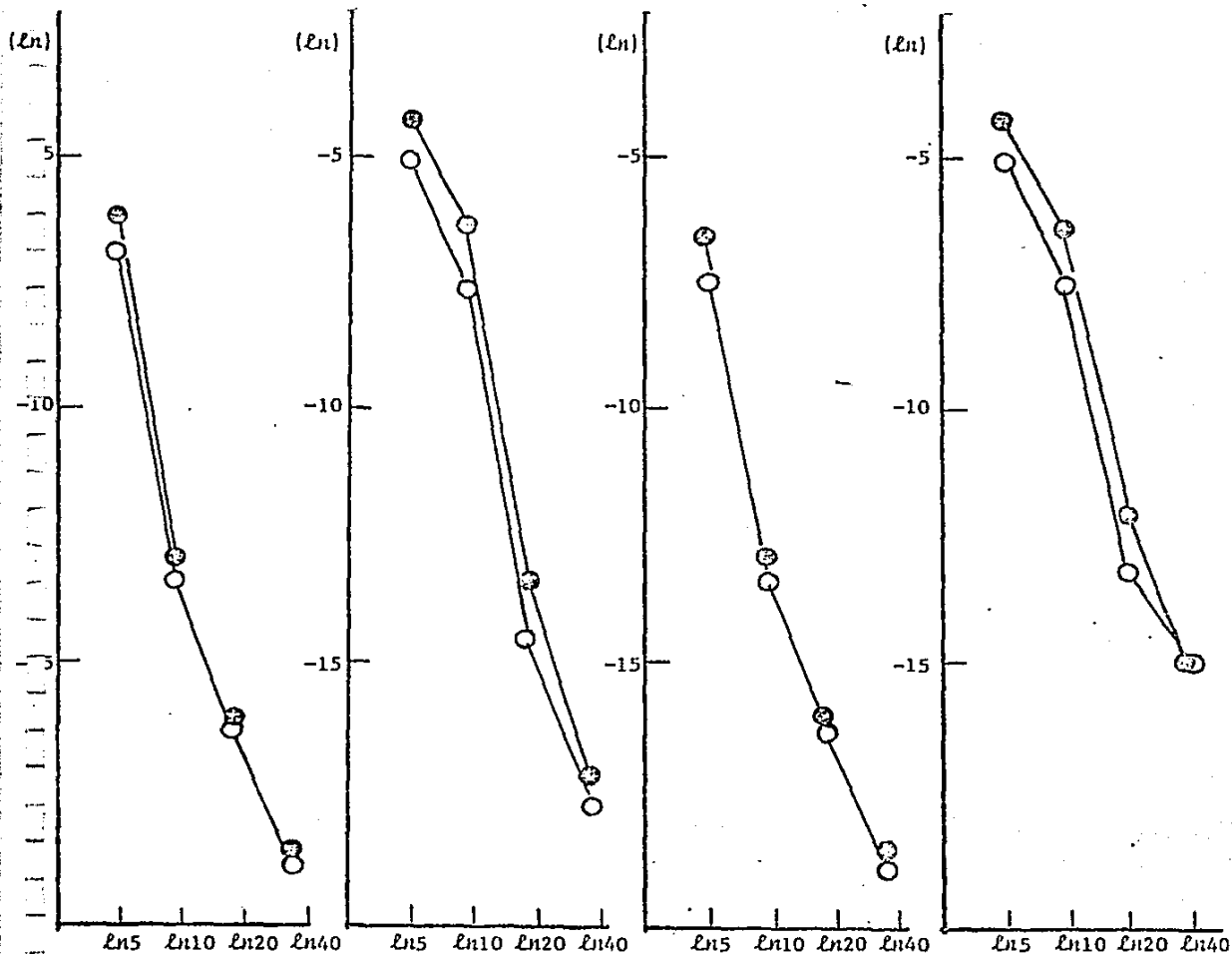
ANALISIS DEL PROBLEMA 1 PARA LOS SIGUIENTES METODOS

ELEMENTOS REGULARES

PUNTO CUARTO

REFINAMIENTO LOCAL

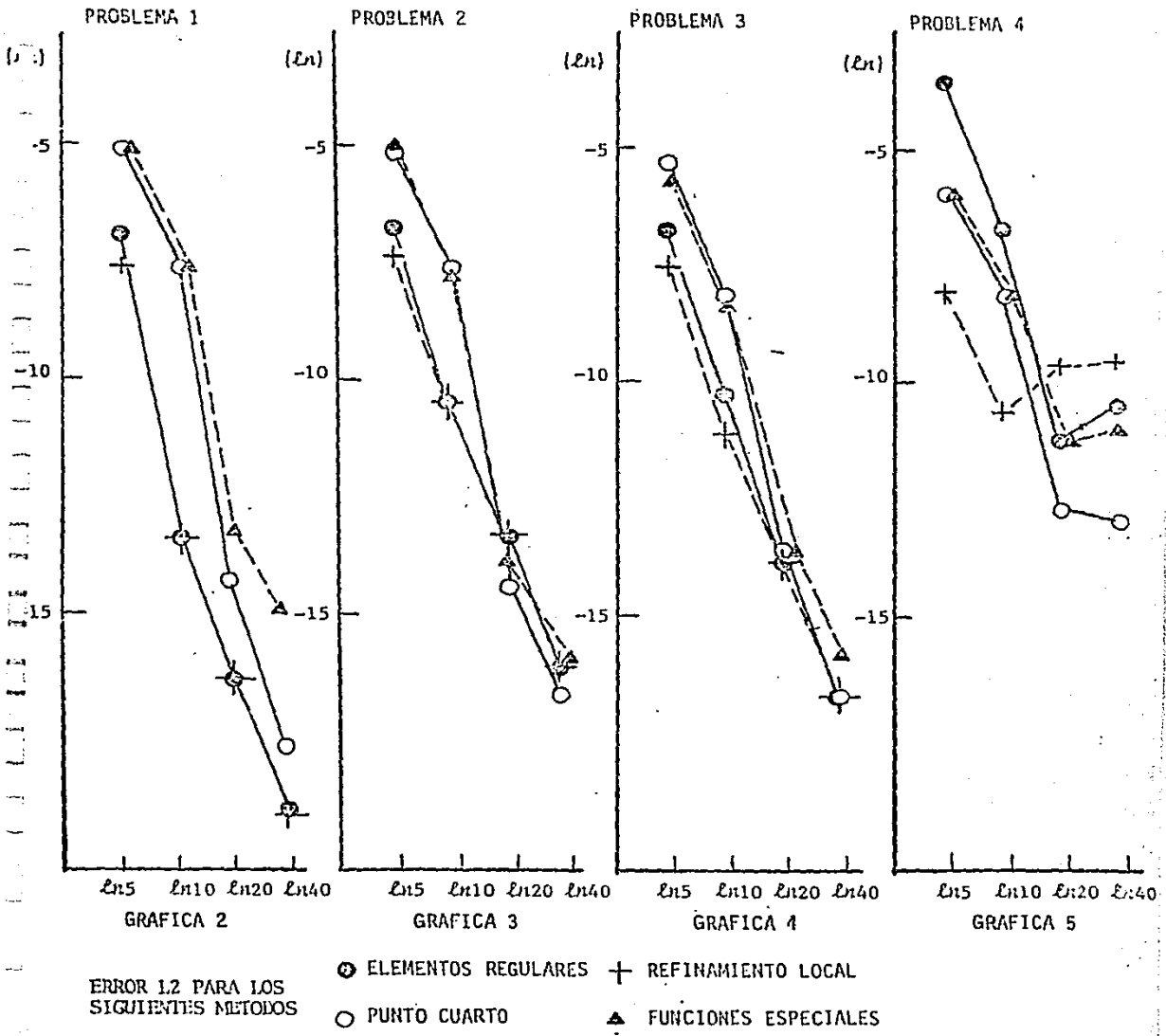
FUNCIONES ESPECIALES



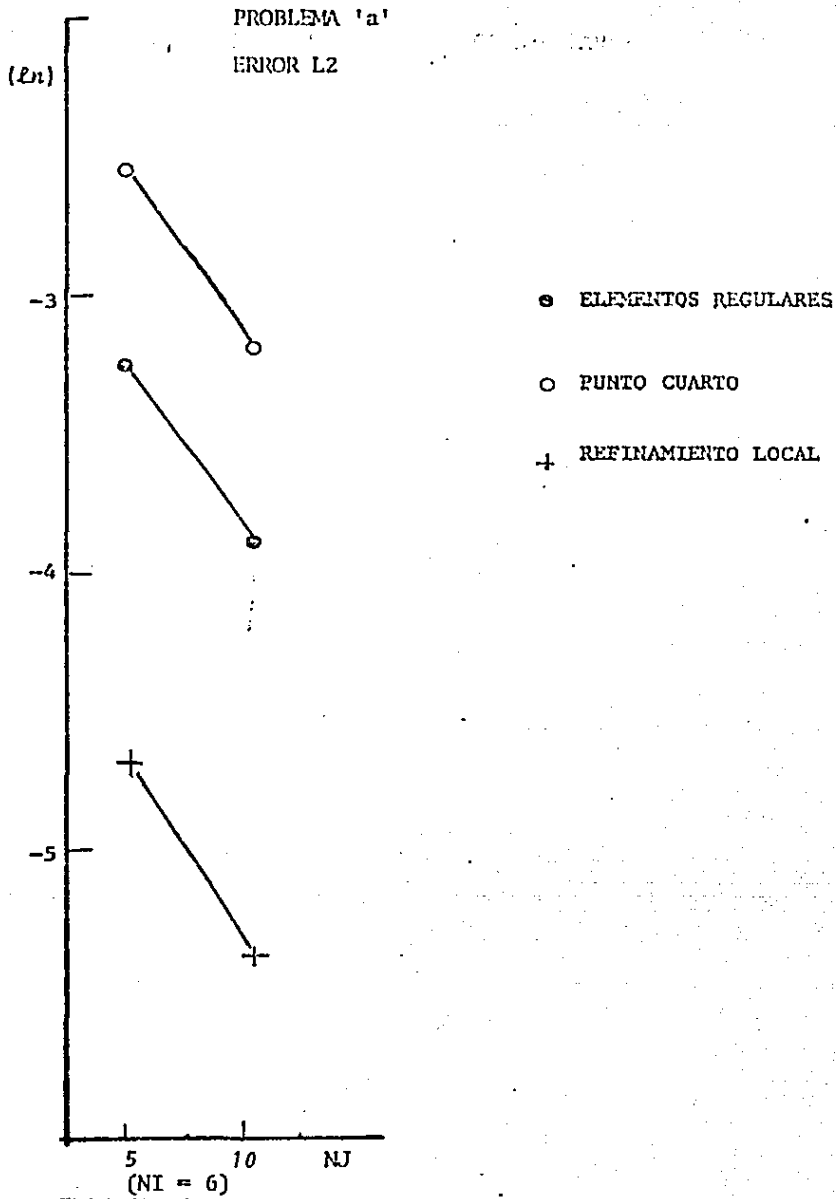
⊙ ERROR MAXIMO

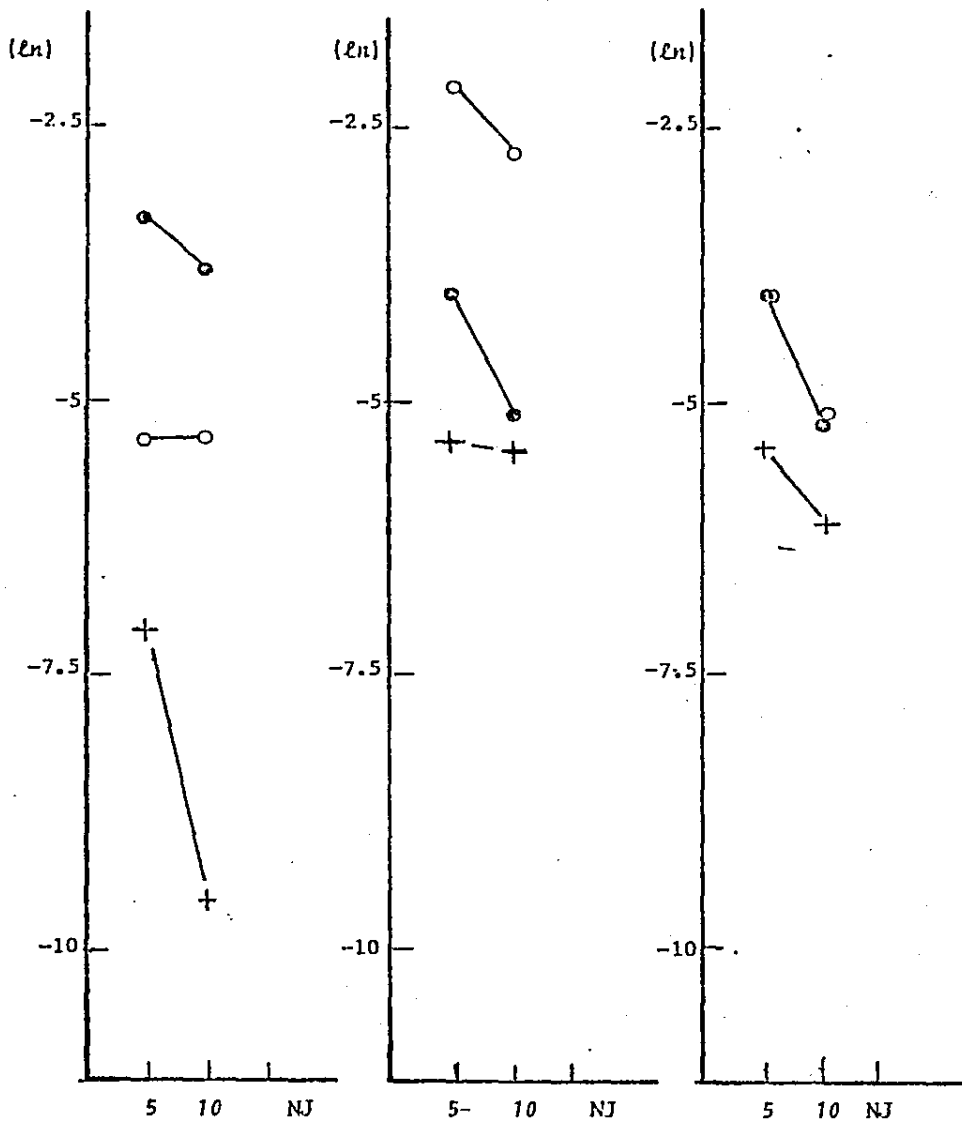
○ ERROR L2

GRAFICA 1



ERROR 1.2 PARA LOS SIGUIENTES METODOS





NI = 6

ERROR L2 EN PROBLEMAS BIDIMENSIONALES CON METODO DE ELEMENTO REGULAR

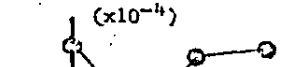
- | | | |
|----------------|--------------|--------------|
| ● PROBLEMA 'a' | ● PROBLEMA 3 | ● PROBLEMA 6 |
| ○ PROBLEMA 1 | ○ PROBLEMA 4 | ○ PROBLEMA 7 |
| + PROBLEMA 2 | + PROBLEMA 5 | + PROBLEMA 8 |

GRAFICA 7

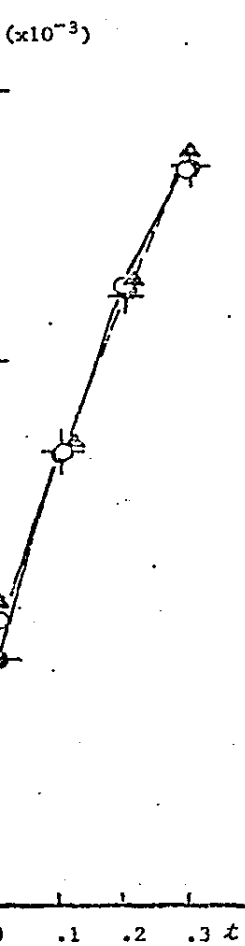
ERROR L2 PARA LOS PROBLEMAS DEPENDIENTES DEL TIEMPO
UTILIZANDO LOS SIGUIENTES METODOS

- ⊙ ELEMENTOS REGULARES
- ⊕ REFINAMIENTO LOCAL
- PUNTO CUARTO
- △ FUNCIONES ESPECIALES

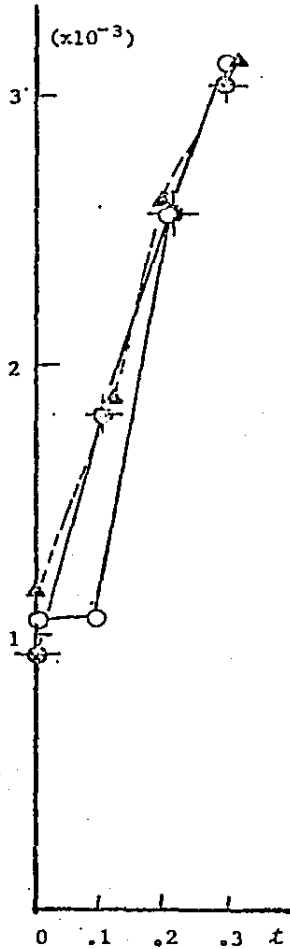
PROBLEMA 8



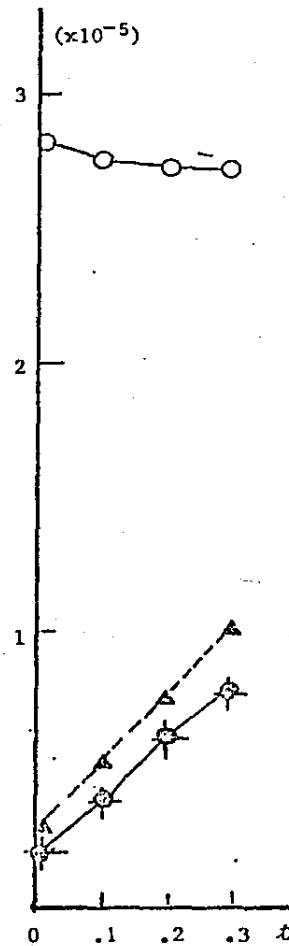
PROBLEMA 5



PROBLEMA 6



PROBLEMA 7



PROBLEMA 8

PROBLEMA 8

PROBLEMA 8

GRAFICA 8

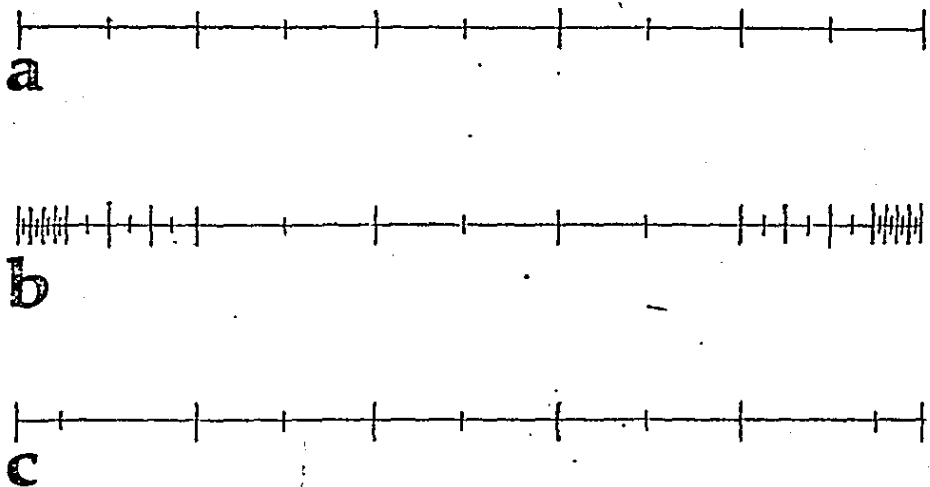
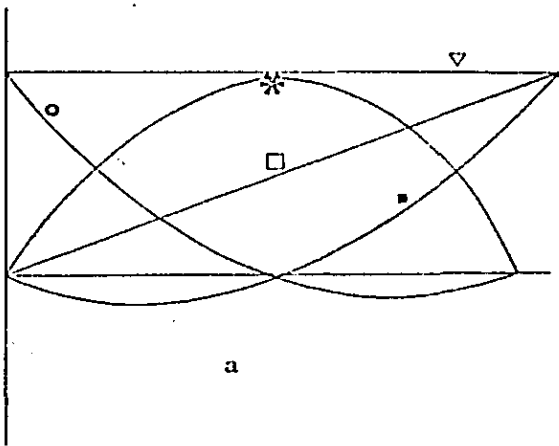


figura (1) Mallas de elementos finitos unidimensionales:

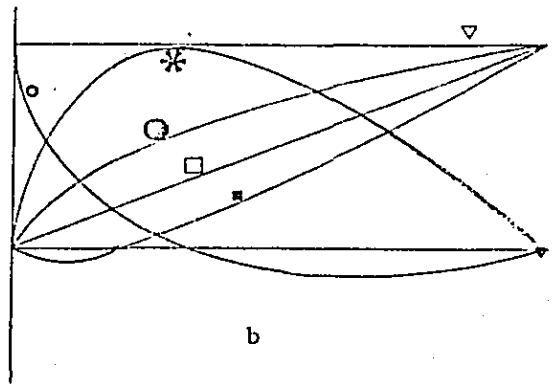
(a) malla regular con $N=5$ intervalos.

(b) refinamiento local dividiendo el primer (y el último) intervalo en cuatro subintervalos, y en seguida lo mismo con el primer (y último) subintervalo.

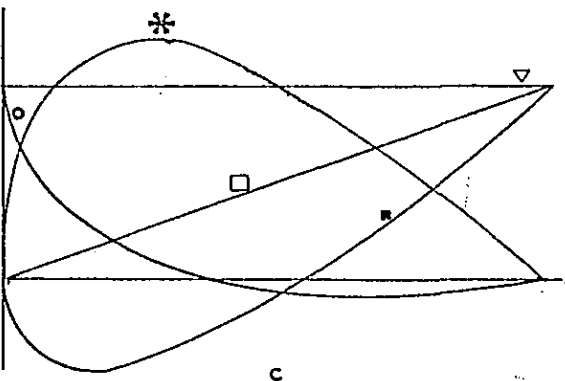
(c) malla con elementos singulares con puntos cuartos en el primer (y último) intervalo.



a



b



c

- $\nabla y=1$
- $\square y=x$
- $\odot y=\sqrt{x}$
- $\circ \gamma_1$
- $* \gamma_2$
- $\blacksquare \gamma_3$

figura (2)

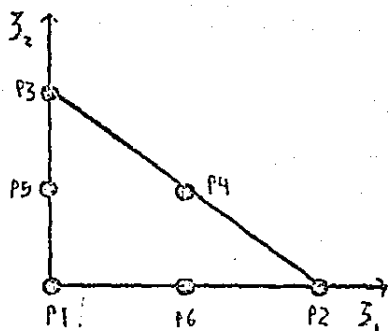
(a) funciones base cuadráticas en $[0, 1]$ cuando tenemos x_1, x_2, x_3 son $\gamma_1, \gamma_2, \gamma_3$ y valen respectivamente 1 en $x_1=0, x_2=0.5$ y $x_3=1$. Como referencia se trazaron $y=1$ y $y=x$.

figura (2)

(b) funciones base cuadráticas en $[0, 1]$ cuando tenemos x_1, x_2 y x_3 son $\gamma_1, \gamma_2, \gamma_3$ y valen respectivamente 1 en $x_1=0, x_2=0.25$ y $x_3=1$. Como referencia se trazaron $y=1, y=x$ y $y=\sqrt{x}$.

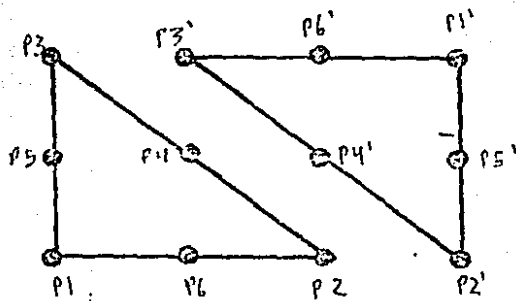
figura (2)

(c) funciones base especiales en $[0, 1]$. Es de notarse que se alcanzan valores que no cumplen $-1 \leq y \leq 1$. como referencia se trazaron $y=1$ y $y=x$.



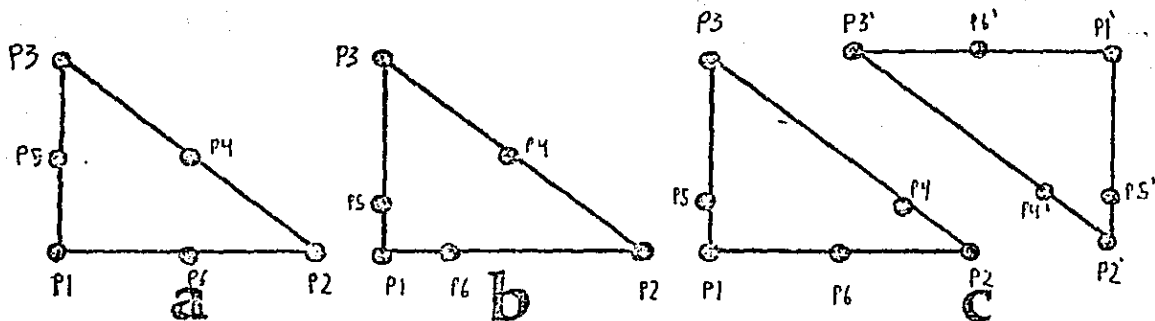
figura(3)

(a) Triángulo Cuadrático estándar en el plano (z_1, z_2) .



figura(3)

(b) Cada rectángulo en el plano (x, y) se divide en los dos triángulos mostrados.

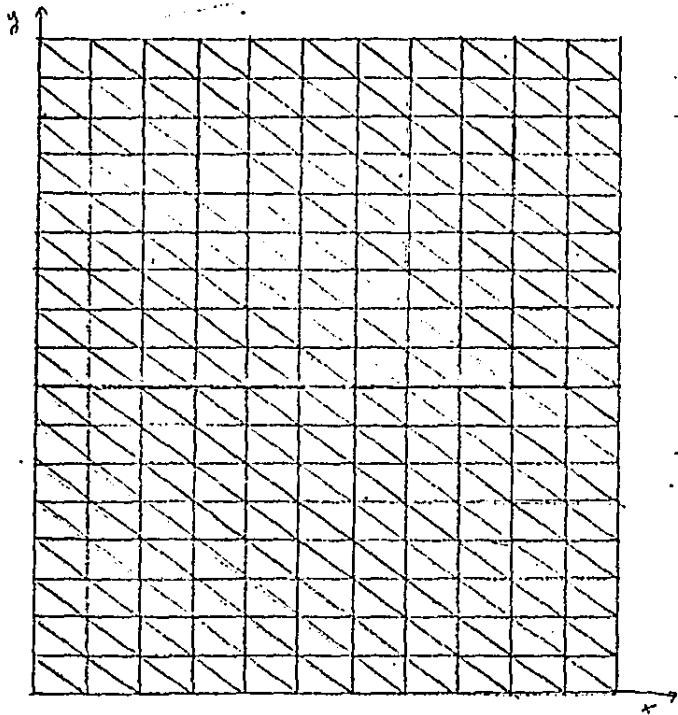


figura(4)

(a) un triángulo cuadrático usual.

(b) el triángulo singular con puntos cuartos usado cuando tenemos una singularidad en P_1 .

(c) el par de triángulos que se usarán para singularidades en el lado P_1P_2 .



figura(5) Malla uniforme bidimensional con $NI=11$ y $NJ=17$.

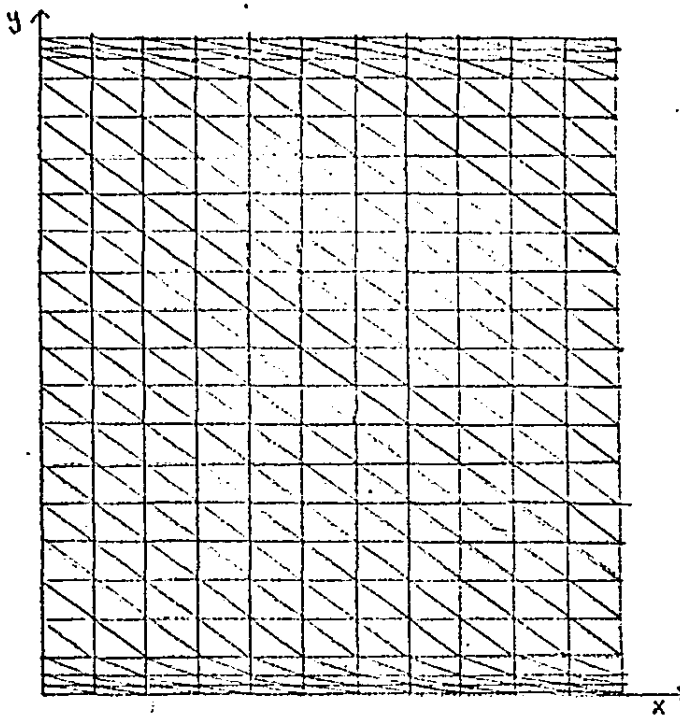


Figura (6) Malla localmente refinada para tratar singularidades en $y=y_f$ y $y=y_f$.

Programas:

ARCHIVO: (ISA) PROGRAMA ORIGINAL ON I.MAS.
13 PAGINAS, 313 LINEAS.
452 REGISTRUS.

FECHA: (10/21/83); 13:41 HRS.
ULTIMO CAMBIO: (10/13/83).

PAGINAS: 1-END.

** E SPOOLER ** VERSION 3
DEPTO. DE COMPUTACION
I. I. M. A. S. U.N.A.M.

DIRECCIONARIO DE PAGINAS.

PAGINA

LINEAS

CONTENIDO DE LA PRIMERA LINEA

1	19	C* DECLARACIONES PROGRAMA/ORIGINAL
2	28	C* PROGRAMA PRINCIPAL
3	29	SUBROUTINE STURM (NI, ILBC, IRBC, NP, A, U, V, AF)
4	49	C* STIFF=====
5	17	FUNCTION F(X, IPR)
6	17	FUNCTION G(X, IPR)
7	19	FUNCTION F(X, IPR)
8	19	FUNCTION FEX(X, IPR)
9	47	SUBROUTINE ERROR (U, NI, ILBC, XI, XF, IPK, NIMP, H, Nr)
10	18	FUNCTION FINI(X1, X2, F, I, J, ID, IPR)
11	28	FUNCTION UIJ (J, ID, X1, X2, Xc)
12	44	SUBROUTINE CHOINT (N, M, A, L)
13	39	SUBROUTINE CHOENS (N, M, L, XT8)

```
1 C* DECLARACIONES PROGRAMA/ORIGINAL
2 FILE 5=PVF/FEIND7UNIT=DISK,RECORD=14,BLOCKING=420
3 FILE 6=IMP,UNIT=PRINTER,RECORD=24
4 FILE 7=MSG,UNIT=MEMOIE
5 DIMENSION A(1200)
6 C*
7 C* ESTE PROGRAMA RESUELVE LA ECUACION  $-(U')^2+U= F$ 
8 C* CON CONDICIONES EN LA FRONTERA  $U=0$  O  $U'=0$ 
9 C* EMPLEANDO ELEMENTOS FINITOS CUADRATICOS SOBRE  $(X_I, X_F)$ 
10 C* Y CUADRATURA GAUSIANA CON CUATRO PUNTOS PARA CALCULAR
11 C* LOS COEFICIENTES DE LA MATRIZ Y DEL VECTOR SEGUNDO MIEMBRO
12 C* EN  $X=X_I$  Y  $X=X_F$ 
13 C*
14 C* NI = NUMERO DE INTERVALOS
15 C* ILBC = CONDICION DE FRONTERA EN  $X=X_I$ 
16 C* =0 U=0, =1 U'=0
17 C* IRBC = CONDICION DE FRONTERA EN  $X=X_F$ 
18 C* =0 U=0, =1 U'=0
19 C*
```



```

1  C*  PROGRAMA PRINCIPAL
2  C*  10  READ(5,7) NI,ILBC,IRBC
3  C*      WRITE (6,900) NI,ILBC,IRBC
4  C*      WRITE (7,1000)
5  C*      IF (NI.EQ.0)CALL EXIT
6  C*
7  C*      NP=NI*2-1+ILBC+IRBC
8  C*
9  C*      LA=NP*3
10 C*
11 C*      IA=1
12 C*      IU=IA+LA
13 C*      IB=IU+NP
14 C*      IAF=IB+NP
15 C*      IFI=IAF+NP*3-1
16 C*      IF (IFI.LE.1500) GO TO 20
17 C*      WRITE(7,1100) IFI
18 C*      WRITE(6,1100) IFI
19 C*      CALL EXIT
20 C*  20  CALL STORM(NI,ILBC,IRBC, NP,A(IA),A(IU),A(IB),A(IAF))
21 C*      GO TO 10
22 C*
23 C*  900  FORMAT (I11,"ELEMENTOS CUADRAICOS = GAUSS 4 PUNTOS"//2X,"NI=",I5,
24 C*        * 2X,"ILBC=",I5,2X,"IRBC=",I5)
25 C*  1000 FORMAT (" DATA OK")
26 C*  1100 FORMAT(I11,"A ES DEMASIADO PENUENO, IFI=",I5)
27 C*
28 C*      END

```

```

1 SUBROUTINE STURN (NI, ILBC, IRBC, NP, A, U, B, AF)
2 C*
3 DIMENSION A(NP,3), U(NP), B(NP), AF(NP,3)
4 C*
5 X1 = ABCISA INICIAL
6 C* XF = ABCISA FINAL
7 C* IPR = INDICE DEL PROBLEMA
8 C* NIMP = NUMERO DE PUNTOS DE IMPRESION
9 C*
10 READ(5,/) XI, XF, IPR, NIMP
11 WRITE (6,1000) XI, AF, IPR, NIMP
12 WRITE (7,1100)
13 H=(XF-XI)/NI
14 C*
15 CALL STIFF(A, U, XI, H, NI, ILBC, IRBC, NP, IPR)
16 WRITE(7,/) A, B
17 C*
18 CALL CHOBNT (NP,3,A,AF)
19 C*
20 CALL CHOBNS (NP,3,AF,U,B)
21 C*
22 CALL ERROR (U, NI, ILBC, XI, XF, IPR, NIMP, H, NP)
23 C*
24 1000 FORMAT(1H0,2X,"XI=",1PE10.3,2X,"XF=",1E10.3,2X,"IPR=",I3,2X,
25 * "NIMP=",I5/77)
26 1100 FORMAT(" DATA 2 ON")
27 C*
28 RETURN
29 END

```

```

1  C* STIFF=====
2  SUBROUTINE STIFF (A,H,XI,H,NI,ILBC,IRBC,NP,IPR)
3  C*
4  C*
5  C*  ESTA RUTINA SE ENCARGA DE CALCULAR A Y B, SIENDO
6  C*  ENTONCES LA PARTE EN QUE SE UTILIZA FEM.
7  C*
8  C*  IPR ES UN INDICE QUE INDICA DE QUE PROBLEMA SE TRATA
9  C*  ILBC=0 SI U(XI)=0, 1 SI SU DERIVADA ES 0
10 C*  IRBC=0 SI U(XI)=0, 1 SI SU DERIVADA ES 0
11 C*
12 C*
13 DIMENSION A(NP,3),B(NP),IND(3)
14 EXTERNAL P,U,F
15 C*
16 DO 10 N=1,NP
17   B(N)=0.
18   DO 10 M=1,3
19     A(N,M)=0.
20 10 CONTINUE
21 C*
22 DO 100 N=1,NI
23   XI=XI+(N-1)*H
24   X2=X1+H
25   IND(1)=(N-1)*4+ILBC
26   IND(2)=IND(1)+1
27   IND(3)=IND(2)+1
28   IMIN=1
29   IMAX=3
30   IF (.EQ.1 .AND. ILBC.EQ.0) IMIN=2
31   IF (.EQ.NI .AND. IRBC.EQ.0) IMAX=2
32   DO 30 I=IMIN,IMAX
33     II=IND(I)
34     DO 20 J=IMIN,1
35       JJ=3-I+J
36       RRR=FINT(X1,A2,U,I,J,0,IPR)
37       SSS=FINT(X1,A2,P,I,J,1,IPR)
38       A(II,JJ)=A(II,JJ) + RRR + SSS
39       WRITE(7,/) RRR,SSS,A(II,JJ)
40 20 CONTINUE
41   TTT=FINT(X1,A2,F,I,0,0,IPR)
42   B(II)=B(II) + TTT
43   WRITE(7,/) TTT,B(II)
44 30 CONTINUE
45 100 CONTINUE
46 C*
47 RETURN
48 END
49 C*

```

```
1      FUNCTION P(X, IPR)
2      C*
3      C*      GO TO(100,200,300,400), IPR
4      C*
5      100  P=1.
6           RETURN
7      C*
8      200  P=1.+X
9           RETURN
10     C*
11     300  P=EXP(X)
12          RETURN
13     C*
14     400  P=EXP(X)
15          RETURN
16     END
17     C*
```

```
1      FUNCTION D(X, IPR)
2      C*
3      GO TO (100, 200, 300, 400), IPR
4      C*
5      100  D=0.
6          RETURN
7      C*
8      200  D=100.
9          RETURN
10     C*
11     300  D=EXP(X)
12         RETURN
13     C*
14     400  D=EXP(X)
15         RETURN
16     END
17     C*
```

```
1      FUNCTION F(X, IPR)
2      C*
3      DATA PI/3.1415926535898/
4      C*
5      GO TO (100, 200, 300, 400), IPR
6      C*
7      100 F=SIN(3.*PI*X)
8          RETURN
9      C*
10     200 F=-PI*COS(PI*X)+(100+PI**2*(1+X))*SIN(PI*X)
11         RETURN
12     C*
13     300 F=EXP(X)*(X**2-2.*X-1.)
14         RETURN
15     C*
16     400 F=EXP(X)*(X**2-5*X**4-15*X**3+12*X**2)
17         RETURN
18     END
19     C*
```

```
1      FUNCTION FEA(A, IPR)
2      C*
3      DATA PI/3.1415926535898/
4      C*
5      GO TO (100,200,300,400), IPR
6      C*
7      100 FEX=SIN(3.*PI*X)/(9.*PI**2)
8          RETURN
9      C*
10     200 FEX=SIN(PI*A)
11         RETURN
12     C*
13     300 FEX=X*(X-1.)
14         RETURN
15     C*
16     400 FEX=X**4*(X-1.)
17         RETURN
18     END
19     C*
```

```

1  SUBROUTINE ERROK (U, NI, ILBC, XI, XF, IPR, NIMP, H, NP)
2  C*
3  DIMENSION U(NP)
4  C*
5  E=0.
6  E2=0.
7  NMAX=1
8  NI=(XF-XI)/FLUAI(NIMP-1)
9  DO 100 N=1, NIMP
10 XIMP=XI+(N-1)*H
11 DO 10 M=1, NI
12 XI=XI+(M-1)*H
13 IF (XI.LE.XIMP) GO TO 10
14 II=M-1
15 GO TO 20
16 10 CONTINUE
17 II=NI
18 20 IND1=(II-1)*2*ILBC
19 IND2=IND1+1
20 IND3=IND2+1
21 X1=XI+(II-1)*H
22 X2=X1+H
23 U1=0.
24 IF (IND1.GE.1) U1=U(IND1)
25 U2=U(IND2)
26 U3=0.
27 IF (IND3.LE.NP) U3=U(IND3)
28 UA=U1*U1J(1,0;X1,A2;XIMP)+U2*U1J(2,0,X1,X2,XIMP)
29 * +U3*U1J(3,0;X1,A2,XIMP)
30 UE=FE(XIMP,IPR)
31 ER=UE-UA
32 IF (ABS(ER).GT.E) NMAX=N
33 E=AMAX1(ABS(ER),E)
34 E2=E2+ER**2
35 WRITE (6,1100) XIMP,UE,UA,ER
36 100 CONTINUE
37 E2=SQRT(E2/NIMP)
38 XMAX=XI+(NMAX-1)*H
39 WRITE(6,1200)E,XMAX,E2
40 WRITE(7,/)E,XMAX,E2
41 1100 FORMAT(2X,"XIMP=",1PE10.3,2X,"UE=",E15.8,2X,"UA=",E15.8,3X,
42 * "ER=",E10.3)
43 1200 FORMAT(//," EL ERROK MAXIMO ES",1PE10.3,2X,"EN X=",E10.3//
44 * " EL ERROR L ES",E10.3)
45 C*
46 RETURN
47 END

```



```
1 FUNCTION FINT(X1,X2,F,I,J,IO,IPC)
2 C* INTEGRACION DE GAUSS DE 4 PUNTOS
3 DIMENSION X(4),W(4)
4 DATA X/-0.861136311594053,-0.339981043584056,
5 * +0.339981043584056,+0.861136311594053/
6 DATA W/0.347854845137454,0.652145154862540,
7 * 0.652145154862540,0.347854845137454/
8 C*
9 FINT=0.
10 DO 10 N=1,4
11 XC=(X2-X1)*(X(N)+1.)/2.+X1
12 FINT=FINT+W(N)*F(XC,IPC)
13 *UIJ(J,IO,X1,XC,XC)
14 10 CONTINUE
15 FINT=FINT*(X2-X1)/2.
16 C*
17 RETURN
18 END
```

```
1 FUNCTION UIJ (J, IU, X1, X2, XC)
2 C* FUNCIONES BASE
3 IOP=IU+1
4 JP=J+1
5 H2=(X2-X1)**2
6 GO TO (10,50),IOP
7 C*
8 10 GO TO (20,30,35,40),JP
9 C*
10 20 UIJ=1.
11 RETURN
12 30 UIJ=(2.*XC-X1-X2)*(XC-X2)/H2
13 RETURN
14 35 UIJ=4.*(XC-X1)*(XC-XC)/H2
15 RETURN
16 40 UIJ=(XC-X1)*(2.*XC-X1-X2)/H2
17 RETURN
18 C*
19 50 GO TO (60,60,65,70),JP
20 C*
21 60 UIJ=(4.*XC-X1-3.*X2)/H2
22 RETURN
23 65 UIJ=4.*(X1+X2-2.*XC)/H2
24 RETURN
25 70 UIJ=(4.*XC-3.*X1-X2)/H2
26 RETURN
27 C*
28 END
```

```

1  SUBROUTINE CHUNT (N,M,A,L)
2  C*  CHOLESKY
3  DIMENSION A(N,M),L(N,M)
4  REAL A
5  LOGICAL FAIL
6  INTEGER P,Q,R,S
7  FAIL=.FALSE.
8  DT=1.
9  ID=0
10 DO 100 I=1,M
11   P=I
12   IF(I.LT.M) P=M-I+1
13   R=I-M+P
14   DO 50 J=P,M
15     S=J-1
16     Q=M-J+P
17     Y=A(I,J)
18     IF(P.GT.S) GO TO 20
19     DO 10 K=P,S
20       Y=Y-L(I,K)*L(K,Q)
21     Q=Q+1
22 10 CONTINUE
23 20   IF(J.EQ.M) GO TO 30
24     L(I,J) = Y*L(K,Q)
25     R=R+1
26     GO TO 50
27 30   DT=DT*Y
28     IF(Y.EQ.0.) GO TO 900
29 35   IF(ABS(DT).LT.1.) GO TO 40
30     DT=DT*0.1
31     ID=ID+1
32     GO TO 35
33 40   IF(ABS(DT).GE.0.1) GO TO 45
34     DT=DT*10.
35     ID=ID-1
36     GO TO 40
37 45   L(I,J)=1./SQRT(DT)
38 50 CONTINUE
39 100 CONTINUE
40 110 RETURN
41 900 PRINT 1000
42 1000 FORMAT(10H0CHH1HVAUA)
43 CALL EXIT
44 END

```

```

1  SUBROUTINE CHURNS (N,H,L,X,H)
2  C*  CHOLESKY PARTE 2
3  DIMENSION L(N,1),A(1),B(1)
4  REAL L
5  INTEGER P,Q
6  C*
7  C*FORWARD SWEEP
8  C*
9  M1=M-1
10 DO 25 I=1,N
11   P=1
12   IF(I.LT.M) P=M-I+1
13   Q=-M+I-1
14   Y=B(I)
15   IF(P.GT.M1) GO TO 20
16   DO 10 K=P,M
17     Q=Q+1
18     Y=Y-L(I,K)*A(Q)
19   10 CONTINUE
20   X(I)=Y*L(I,M)
21   25 CONTINUE
22  C*
23  C* BACKWARD SUBSTITUTION
24  C*
25  DO 50 I=1,N
26   IB=N-I+1
27   P=1
28   IF(I.LT.M) P=M-I+1
29   Y=X(IB)
30   M=IB+1-P
31   IF(P.GT.M1) GO TO 40
32   DO 30 K=P,M
33     Y=Y-L(Q,K)*A(Q)
34   30 CONTINUE
35   40 X(IB)=Y*L(IB,M)
36   50 CONTINUE
37  RETURN
38  END
39

```

ARCHIVO: (ISAX)PROGRAMA/10/1 ON IIMAS:
1^a PAGINA, 109^o LINEAS.
1147 REGISTROS.

FECHA: (10/21/83); 15:42 HRS.
ULTIMO CAMBIO: (10/15/83).

PAGINAS: 1-END.

** E SPOOLER ** VERSION 3
DEPIO. DE COMPUTACION
I. J. M. A. S. - U.N.A.M.

DIRECCIONARIO DE PAGINAS.

PAGINA

LINEAS

CONTENIDO DE LA PRIMERA LINEA

1	56	C* COMMENT=====PROGRAMA/10/1=====
2	101	C* MAIN=====
3	109	C* SIOHM=====
4	79	C* STIFF=====
5	43	C* FUNCIONES=====
6	30	FUNCTION G(X1,X2,X3,U1,U2,U3,I,TIME,IO,IPR,IC)
7	35	FUNCTION F(X1,X2,X3,U1,U2,U3,I,TIME,IO,IPR,IC)
8	48	FUNCTION FEX(XA,TIME,IO,IPR)
9	81	C* ERROR=====
10	52	C* FIN=====
11	98	C* BASE=====
12	17	C* CALL=====
13	47	C* CHECK=====
14	51	C* FACION=====
15	57	C* SOL=====
16	77	C* ERROR2=====
17	73	C* INICIA=====
18	7	C* FUNCTION UNO=====
19	32	C* FUNCTION TIM=====

```

1  C* COMMENT=====PROGRAMA/10/1=====
2  SSET LINEIN,0
3  C*
4  C*
5  C*
6  C*
7  C*
8  C*
9  C*
10 C*
11 C*
12 FILE 5(KIND=DISK,MAXRECSIZE=14,BLOCKSIZE=420,TITLE="PVF/P1M7")
13 FILE 6(KIND=DISK,MAXRECSIZE=14,BLOCKSIZE=420,TITLE="UBA/A1")
14 FILE 7(KIND=DISK,MAXRECSIZE=14,BLOCKSIZE=420,TITLE="UBB/A1")
15 FILE 8(PVF/DATE,UNIT=DISK,RECORD=14,BLOCKING=420)
16 FILE 9(MSH,UNIT=MEMOIE)
17 FILE 10(PVF/DATE,UNIT=DISK,RECORD=14,BLOCKING=420)
18 DIMENSION RHA(100),A(100),PU(3,500),RIM(500)
19 C* =====
20 C*
21 C*
22 C*
23 C* ESTE PROGRAMA RESUELVE LA ECUACION DU/DI= (U')'-NU*F
24 C*
25 C* EMPLEANDO ELEMENTOS FINITOS CUADRATICOS SOBRE (X1,XF)
26 C* UTILIZANDO UNA MALLA REGULAR CALCULADA AUTOMATICAMENTE,
27 C* O BIEN LEYENDO DE UN FILE UNA MALLA IRREGULAR SI ASI SE
28 C* DESEA, TODO ESTO PARA LOS TIEMPOS TIME0 A TIMEF CON INCRE-
29 C* MENTOS EN TIEMPO DELTAT.
30 C* USO CUADRATURA GAUSIANA CON CUATRO PUNTOS PARA CALCULAR
31 C* LOS COEFICIENTES DE LA RAIZ Y DEL VECTOR SEGUNDO MIEMBRO
32 C* EN X=XI Y X=XF.
33 C* P, Q Y F PUEDEN SER FUNCIONES DE LA SOLUCION, Y POR LO
34 C* TANTO LA ECUACION PUEDE SER NO LINEAL.
35 C*
36 C* NI = NUMERO DE INTERVALOS
37 C* ILBC = CONDICION DE FRONTERA EN X=XI
38 C*      = 0 U=0, =1 U'=0
39 C* IRBC = CONDICION DE FRONTERA EN X=XF
40 C*      = 0 U=0, =1 U'=0
41 C* LEE ME DICE COMO CALCULAR AUTOMATICAMENTE LA MALLA (=1,...,4)
42 C*
43 C* 1.- MALLA REGULAR AUTOMATICA
44 C* 2.- PUNTOS CUARTOS AUTOMATICOS EN EXTREMOS, REGULAR LO DEMAS;
45 C* 3.- PUNTOS CUARTOS PERO LO DEMAS VARIA LENTAMENTE.
46 C* 4.- REFINAMIENTO EN LOS EXTREMOS.
47 C*
48 C* O BIEN LEER DE PVF/DATE (=5)
49 C*      DE PVF/DATE2 (=6)
50 C* O FUNCIONES ESPECIALES CON MALLA REGULAR (=7)
51 C* TERMINAR SI >7;
52 C*
53 C* =====
54 C*
55 C*
56 C*

```

```

59          PU(3,IY)=PU(1,IY) + III
60 265      CONTINUE
61          GO TO 280
62 27       FIL=R
63          GO TO 271
64 28       FIL=10
65          NI=VI+12
66 271      DO 15 IY=1,NI
67          READ(FIL,/) PU(1,IY),PU(2,IY),PU(3,IY)
68 15       CONTINUE
69 280      CONTINUE
70          NP=VI*2-1+ILBC*IRBC
71          LA=VP*3
72 C*
73          IA=1
74          IU=IA+LA
75          ID=IU+NP
76          IA=IR+NP
77          IFI=IAF+NP*3-1
78          IF (IFI.LE.1000) GO TO 20
79          WRITE(6,1100) IFI
80          CALL EXII
81 20       CALL STORML(E,NI,PU,ILBC,IRBC,NP,HMA,
82          * A(IA),A(IU),A(ID),A(IAF),X1,Xr,H,M,IESr)
83 1        CONTINUE
84          GO TO 10
85 11       CONTINUE
86          LUCK 6
87          LUCK 7
88          CALL EXIT
89 C*
90 900      FORMAT (I11,"ELEMENTOS CUADRATICOS = GAUSS 4 PUNTOS="//2X,"NI=",I5,
91          * 2X,"ILBC=",I3,2X,"IRBC=",I3)
92 1100     FORMAT(I11,"A ES DEMASIADO PEQUENO,IFI=",I5)
93 C*
94          END
95
96
97
98
99
100
101

```



```

1  C* MAIN=====
2  C*
3  C*
4  C*   PROGRAMA PRINCIPAL:
5  C*
6  C* =====
7  M=3
8  10 CONTINUE
9  READ(5,/) LEE,NUMV,XI,XF
10 C* WRITE(6,*) LEE,NUMV,XI,XF
11 C* WRITE(7,*) LEE,NUMV,XI,XF
12 IF (LEE.GT.1) GO TO 11
13 DO 1 IY=1,NUMV
14 READ(5,/) NI,ILBC,IRBC
15 H=(XF-XI)/NI
16 C* WRITE(6,*) H,NI,ILBC,IRBC
17 C* WRITE(7,*) H,NI,ILBC,IRBC
18 C*
19 IESP=0
20 GO TO (23,24,25,26,27,28,21)LEE
21 IFSP=1
22 RIM(1)=0.5
23 RIM(NI)=0.5
24 FI=0.
25 GO TO 250
26 RIM(1)=0.25
27 RIM(NI)=0.75
28 FI=0.
29 GO TO 250
30 RIM(1)=0.25
31 RIM(NI)=0.75
32 FI=0.5/FLOAT(NI-2)
33 CONTINUE
34 NI1=NI-1
35 DO 256 I=2,NI1
36 RIM(I)=RIM(I-1)+FI
37 IF (FI.LE.0) RIM(I)=0.5
38 CONTINUE
39 DO 257 N=1,NI
40 PU(1,N)=X1+(N-1)*H
41 PU(2,N)=PU(1,N)+RIM(N)*H
42 PU(3,N)=PU(1,N)+H
43 CONTINUE
44 GO TO 280
45 26 CONTINUE
46 H1=H/16.
47 H2=H/4.
48 NI=NI+12
49 NI2=NI-3
50 PU(1,1)=X1
51 DO 265 IY=1,NI
52 HH=H
53 IF( (IY.LE.4) .OR. (IY.GE.NI2) ) HH=H1
54 IF( (IY.GE.4) .AND. (IY.LE.7) ) HH=H2
55 IF( (IY.GE.(NI2-3)) .AND. (IY.LI.NI4) ) HH=H2
56 IF (IY.GE.1) PU(1,IY)=PU(3,IY-1)
57 PU(2,IY)=PU(1,IY) + (0.5*HH)
58

```

```

1 C* STURM=====
2 C*
3 C*
4 C*
5 C*   ESTA RUTINA CONTROLA EL PROCESO EN GENERAL;
6 C*   LLAMA A TODAS LAS DEMAS RUTINAS CUANDO SE
7 C*   REQUIEREN. CASO PARABOLICO NO LINEAL ISOPARAMETRICO
8 C*   IPR INDICE DE PROBLEMA
9 C*
10 C* =====
11 C* SUBROUTINE STURM (LEL, NI, PU, ILBC, IRBC, NP, RMA, A, U, B, AF,
12 C*   * XI, XF, H, N7IESP)
13 C*
14 C*   DIMENSION PU(3, NI), NPVT(500), U(NP), A(NP, M), B(NP, M)
15 C*   DIMENSION UAU(500), UNEN(500), AF(NP, M)
16 C*   IPR = INDICE DEL PROBLEMA
17 C*   NIMP = NUMERO DE PUNTOS DE IMPRESION
18 C*
19 C*   READ (5, /) TIMEU, TIMEF, DELTAT, THETA
20 C*   WRITE (7, *) TIMEU, TIMEF, DELTAT, THETA
21 C*   WRITE (6, *) / TIMEU, TIMEF, DELTAT, THETA
22 C*   READ (5, /) IPR, NIMP, EPSI, EPSK, LIMIT
23 C*   WRITE (7, *) IPR, NIMP, EPSI, EPSK, LIMIT
24 C*   WRITE (6, *) IPR, NIMP, EPSI, EPSK, LIMIT
25 C*
26 C*   DO 11 N=1, NP
27 C*     U(N)=0.51
28 C*
29 C* 11 CONTINUE
30 C*   TIME = TIMEU
31 C*   TU=-1.0
32 C*   DA UNA APROXIMACION INICIAL A LA SOLUCION
33 C*   CALL INICIA(U, PU, TIME, TU, LEE, RMA, NI, ILBC, IRBC, NP, IPR, M, IESP)
34 C*
35 C* 80 IF (TIME.EQ.TIMEU) GO TO 81
36 C*   IF (IESP.EQ.'0') GO TO 801
37 C*   RUTINAS DE ERROR
38 C*
39 C*   CALL ERROR2(TIME, TU, U, NI, ILBC, XI, XF, PU, IPR, NIMP, H, NP, LEE)
40 C*   GO TO 81
41 C* 801 CALL ERROR(TIME, TU, U, NI, ILBC, XI, XF, PU, IPR, NIMP, H, NP, LEE)
42 C*   CONTINUE
43 C*   IER = 0
44 C*   IF (TIME.GT.TIMEF) GO TO 100
45 C*   AVANCE EN TIEMPO
46 C*
47 C*
48 C*   DT=DELTAT*THETA
49 C*   DIT=DELTAT*(1.0-THETA)
50 C*   CALL STIFF(U, PU, TIME, TO, LEE, A, B, NI, ILBC, IRBC,
51 C*     * NP, IPR, M, IESP)
52 C*
53 C*   DO 1020 I = 1, NP
54 C*     MM = A*MAX1(I-2, 1)
55 C*     UAU(I) = 0.1*U(I)
56 C*     DO 1010 J = MM, I
57 C*       METODO THETA
58 C*
59 C*     JIM=J-I*M
60 C*     UAU(I) = UAU(I) - DIT*A(I, JIM)*U(J) + RMA(I, JIM)*U(J)
61 C*     IF (J.LT.I) UAU(J) = UAU(J) - DIT*A(I, JIM)*U(I) +
62 C*       RMA(I, JIM)*U(I)
63 C*
64 C* 1010 CONTINUE
65 C* 1020 CONTINUE

```

```

59     TIME = TIME + DELIAT
60 C*     WRITE(9,*) TIME
61     WRITE(9,104) ((A(LI,LJ),LI=1,MP),LJ=1,M), (B(LK),LK=1,MP)
62 104 FORMAT("  A,0"//G15.8)
63 C*     ITERACION
64 C*
65 330 DO 1030 I = 1, MP
66     MM=AMAX1(I,-2,1)
67     DO 1040 J=MM,I
68     JJ=J-ITM
69     A(I,JIM)=KMA(I,JIM) + DI*A(I,JIM)
70 1040 CONTINUE
71     B(I)=DAMX(I) + DI*B(I)
72 1030 CONTINUE
73     WRITE(9,104) ((A(LI,LJ),LI=1,MP),LJ=1,M), (B(LK),LK=1,MP)
74     CALL FACTOR(A,MP,M,K)           %FACTOR(LA)
75     IF (K.LE.NP) GO TO 334
76     CALL SOL(A,F7B7INEM,MP,M)      %RESUELVE
77     CALL CHECK(U,ONEA,EPSI,EPSP,ITLAG,MP,LIMIT,ITER)
78 C*     PRUEBA CONVERGENCIA
79 C*
80     ITER=ITER+1
81     IF (IFLAG.EQ. 1) GO TO 80
82     CALL STIFF(U,CU,TIME,TO,LEE,A,B,NI,ILBC,INBC,
83 *           NP,ITRIM,IESP)
84     GO TO 330
85 334 WRITE(6,1111)
86 C*     WRITE(6,*) A,AF,K,MP
87 1111 FORMAT("  HURONK ")
88     WRITE(7,1111)
89     CALL EXIT
90 100 CONTINUE
91 C*
92     RETURN
93     END
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109

```

```

1 C* STAFF=====
2 C*
3 C*
4 C*
5 C* ESTA RUTINA SE ENCARGA DE CALCULAR A Y H, SIENDO
6 C* ENTONCES LA PARTE EN QUE SE UTILIZA FEM.
7 C*
8 C* IPR ES UN INUTE QUE INDICA DE QUE PROBLEMA SE TRATA
9 C* ILBC=0 SI U(AI)=U, 1 SI SU DERIVADA ES 0
10 C* IRBC=0 SI U(AF)=U, 1 SI SU DERIVADA ES 0
11 C*
12 C*
13 C* =====
14 C* SUBROUTINE STAFF (U,PU,TIME,TO,LEE,A,B,NI,ILBC,IRBC,
15 C* NP,IPR,M,IESP)
16 C*
17 C* DIMENSION UNF(3),PU(3,NP),A(NP,N),B(NP),ANU(3)
18 C* EXTERNAL P,Q,r
19 C* WRITE(9,111) T(ALL),LL=1,M)
20 111 FORMAT(' ',G15.8)
21 DO 20 N=1,NP
22 B(N)=0.
23 DO 10 MN=1,M
24 A(N,MN)=0.
25 10 CONTINUE
26 20 CONTINUE
27 C*
28 DO 101 N=1,NI
29 X1=PU(1,N)
30 X2=PU(2,N)
31 X3=PU(3,N)
32 IND(1)=(N-1)*2+ILBC
33 IND(2)=IND(1)+1
34 IND(3)=IND(2)+1
35 IND1=IND(1)
36 IND2=IND(2)
37 IND3=IND(3)
38 INTN=1
39 IMAX=3
40 IF (N.EQ.1.ANU,ILBC.EQ.0) IMIN=2
41 IF (N.EQ.NI.AND,IRBC.EQ.0) IMAX=2
42 U1=0.
43 IF (IND1.GE.1) U1=U(IND1)
44 U2=U(IND2)
45 U3=0.
46 IF (IND3.LE.NP) U3=U(IND3)
47 IF (IESP.EQ.0) GO TO 200
48 IC=1
49 IF (N.EQ.1) IC=2
50 IF (N.EQ.NI) IC=3
51 200 DO 30 I=IMIN,IMAX
52 II=IND(I)
53 DO 201 J=IMIN,1
54 JJ=M-1+J
55 RRR=FINT(TIME,TO,U1,U2,U3,X1,X2,X3,U,1,J,0,IPR,IC)
56 SSS=FINT(TIME,TO,U1,U2,U3,X1,X2,X3,r,1,J,1,IPR,IC)
57 A(II,JJ)=A(II,JJ) + RRR + SSS
58 201 CONTINUE
59 TTT=FINT(TIME,TO,U1,U2,U3,X1,X2,X3,F,I,0,0,IPR,IC)

```

```
59      B(IJ)=B(IJ) + III
60      30 CONTINUE
61      100 CONTINUE
62      C* =====
63      C*
64      101 CONTINUE
65      C* WRITE(9,104) (A(LJ,LJ),LJ=1, NP),LJ=1,MJ (B(LK),LK=1, NP)
66      RETURN
67      END
68
69
70
71
72
73
74
75
76
77
78
79
```

```

1  C* FUNCIONES=====
2  C*
3  C* SIGUEN LAS FUNCIONES P,Q,F Y FEX:
4  C* P, Q Y F CORRESPONDEN A TALES EN LA ECUACION
5  C*  $DU/DT = (P(U)) - QU + F$ 
6  C*
7  C* FEX CALCULA LA SOLUCION EXACTA POR MOTIVOS DE
8  C* COMPARACION.
9  C*
10 C*
11 C* =====
12 C* FUNCION P(X1,X2,X3,U1,U2,U3,T,TIME,TU,IPR,IC)=====
13 C*
14 C* CALL CALC(X1,X2,X3,U1,U2,U3,T,TIME,10,XA,UA,IC)
15 C* GO TO(100,200,300,400,500,600,700,800,900),IPR
16 C*
17 C* 100 P=1.
18 C* RETURN
19 C*
20 C* 200 P=1.+UA**2
21 C* RETURN
22 C*
23 C* 300 P=UA
24 C* RETURN
25 C*
26 C* 400 P=UA
27 C* RETURN
28 C* 500 CONTINUE
29 C* P=1.
30 C* RETURN
31 C* 600 CONTINUE
32 C* P=1.+UA**2
33 C* RETURN
34 C* 700 CONTINUE
35 C* P=UA
36 C* RETURN
37 C* 800 CONTINUE
38 C* P=UA
39 C* RETURN
40 C* 900 CONTINUE
41 C* RETURN
42 C* END
43 C*

```

```
1 FUNCTION D(A1,X2,AS,U1,U2,U3,I,TIME,TU,IPK,IC)
2 DATA PI/3.1415926535898/
3 CALL CALC(X1,A2,X3,U1,U2,U3,T,TIME,I0,XA,UA,IC)
4 C*
5 GO TO (100,200,300,400,500,600,700,800,900),IPK
6 C*
7 100 N=0.
8 RETURN
9 C*
10 200 N=0.
11 RETURN
12 C*
13 300 N=0.
14 RETURN
15 C*
16 400 N=0.
17 RETURN
18 500 N=0.
19 RETURN
20 600 QN=2.*(PI**C)
21 Q=QN**COS(PI**AJ**2)*EXP(-QN*TIME)
22 RETURN
23 700 D=1./(TIME-10)
24 RETURN
25 800 Q=1.
26 RETURN
27 900 RETURN
28 END
29 C*
30 C* =====
```

```

1  FUNCTION F(X1,X2,X3,U1,U2,U3,T,TIME,TU,IPR,IC)
2  DATA PI/3.1415926535898/
3
4  C*
5  CALL CALC(X1,X2,X3,U1,U2,U3,T,TIME,IO,XA,UA,IC)
6  GO TO (100,200,300,400,500,600,700,800,900),IPR
7
8  C*
9  100 F=(PI**2)*UA
10 RETURN
11
12 C*
13 200 CONTINUE
14 F=-(PI**2)*UA*(1. - 3*(UA**2))
15 RETURN
16
17 C*
18 300 F=-(PI**2)*(1. - 2.*(UA**2))
19 RETURN
20
21 C*
22 400 F=((PI*UA)**2)/2.
23 RETURN
24
25 500 CONTINUE
26 SEN=SIN(3.*PI*XA)
27 F=0.
28 RETURN
29
30 600 CONTINUE
31 F=(PI**2)*(UA**2)
32 RETURN
33
34 700 F=-COS(2*PI*XA)/(PI*(TIME-TU))**2
35 RETURN
36
37 800 F=((PI*UA)**2)/2.
38 RETURN
39
40 900 CONTINUE
41 RETURN
42 END
43
44 C* =====
45 C*

```



```
1      FUNCTION FEX(AA, TIME, TO, IPR)
2
3      C*
4      DATA PI/3.1415926535898/
5
6      C*
7      GO TO (100, 200, 300, 400, 500, 600, 700, 800, 900), IPR
8      RETURN
9
10     C*
11     100 FEX=SIN(PI*AA)
12         RETURN
13
14     C*
15     200 FEX=SIN(PI*AA)
16         RETURN
17
18     C*
19     300 FEX=SIN(PI*AA)
20         RETURN
21
22     C*
23     400 SEN=SIN(PI*AA)
24         FEX=SQRT(SEN)
25         RETURN
26
27     C*
28     500 FEX=SIN(PI*AA)*LEAP(-(PI**2)*TIME)
29         RETURN
30
31     C*
32     600 FEX=SIN(PI*AA)*EXP(-(PI**2)*TIME)
33         RETURN
34
35     C*
36     700 FEX=SIN(PI*AA)/((PI**2)*(TIME-TO))
37         RETURN
38
39     C*
40     800 SEN=SIN(PI*AA)
41         FEX=EXP(-TIME)*SQRT(SEN)
42         RETURN
43
44     C*
45     900 CONTINUE
46         RETURN
47         END
48
```

```

1  C* ERROR=====
2  C*
3  C*
4  C* AQUI VEO EL VALOR DE LA SOLUCION EN LOS PUNTOS DADOS
5  C* PARA LA APROXIMACION POR FEM7 LOS COMPARO CON LOS
6  C* CALCULADOS DE MANERA EXACTA, OBTENIENDO EL MAXIMO
7  C* ERROR Y TAMBIEN EL ERROR L2.
8  C*
9  C* =====
10 C*
11 C* SUBROUTINE ERKOK (TIME, IO, U, N1, ILBC, X1, XF, PU, IPR, NIMP, H, NP, LEE)
12 C*
13 C* DIMENSION U(NP), UM(S), PU(S, NP)
14 C*
15 C* IC=1
16 C* E=0.
17 C* E2=0.
18 C* NMAX=1
19 C* NI=(XF-XI)/FLUAI(NIMP-1)
20 C* DO 100 N=1, NIMP
21 C* XIMP=XI+(N-1)*NI
22 C* DO 11 MM=1, NI
23 C* XX=PU(3, MM)
24 C* IF (XIMP.GT.XX) GO TO 10
25 C* IJ=MM
26 C* GO TO 20
27 C* CONTINUE
28 C*
29 C* 10 CONTINUE
30 C* II=NI
31 C* 20 CONTINUE
32 C* IND1=(II-1)*2+ILBC
33 C* IND2=IND1+1
34 C* IND3=IND2+1
35 C* X1=PU(1, II)
36 C* X2=PU(2, II)
37 C* X3=PU(3, II)
38 C* U1=0
39 C* IF (IND1.LE.1) U1=U(IND1)
40 C* U2=U(IND2)
41 C* U3=0
42 C* IF (IND3.LE.NP) U3=U(IND3)
43 C* TIMP=TIME-X1-X2-X3-XIMP
44 C* CALL CALC(X1, X2, X3, U1, U2, U3, TIMP, TIME, IO, XA, UA, IC)
45 C* UE=FEX(XA, TIME, IO, IPR)
46 C* UJ=FEX(XIMP, TIME, IO, IPR)
47 C* ER=UE-UA
48 C* IF (ABS(ER).GT.E) NMAX=N
49 C* E=AMAX1(ABS(ER), E)
50 C* E2=E2+ER**2
51 C* WRITE (6, 111) AA, UE, UA, ER
52 C* 111 FORMAT(' XA, UE, UA, ER', 4E15.8)
53 C* WRITE (9, 111) AA, UE, UA, ER
54 C* 100 CONTINUE
55 C* E2=SQRT(E2/NIMP)
56 C* XMAX=XI+(NMAX-1)*NI
57 C* WRITE(6, 1200) E, XMAX, E2
58 C* WRITE(7, 1200) E, XMAX, E2
59 C* 1100 FORMAT(2X, 'XIMP=', F1PE10.3, 3X, 'UE=', E15.8, 2X, 'UA=', E15.8, 3X,

```

```
59 * "ER=",E10.3)
60 1200 FORMAT(// " EL ERROR MAXIMO ES",1PE10.3,2X,"EN X=",E10.3//
61 * " EL ERROR L ES",E10.3)
62 C*
63 RETURN
64 END
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
```

```

1  C* FINT=====
2  C**
3  C**
4  C**  ESTA RUTINA INTEGRA MEDIANTE CUADRATURA GAUSSIANA DE
5  C**  CUATRO PUNTOS.
6  C**
7  C**
8  C**
9  C**  =====
10 C**  FUNCION FINT(TIME,T0,U1,U2,U3,X1,X2,X3,F,I,J,IO,IPR,IC)
11 C*
12 DIMENSION X(4),U(3),V(3),W(4)
13 DATA X/-0.861136311594053,-0.539981043584056,
14 * +0.339781043584056,+0.061136311594053/
15 DATA W/0.347824043137454,0.452145154862540,-
16 * 0.652145154862540,0.347824043137454/
17 C*
18 FINT=0.
19 DO 13 N=1,4
20   IX(N)
21   IF(IC.EQ.2) I=1+((T+1.0)*2)/2.0 - 1.0
22   IF(IC.EQ.3) I=1.0 - ((1.0+T)*2)/2.0
23   DO 11 IYX=1,3
24     V(IYX)=0.5*(IYX,1,-1.0,1.0,1,IC)
25 11 CONTINUE
26   RJAC=V(1)*X1 + V(2)*X2 + V(3)*X3
27   IF(IC.EQ.2) RJAC=RJAC*(1.0+X(N))
28   IF(IC.EQ.3) RJAC=RJAC*(1.0-X(N))
29   IF (RJAC.EQ. 0) GO TO 17
30   RMULT=RJAC*(1-2*IO)
31   IF(IC.EQ.2) RMULT=RMULT*(1.0+X(N))
32   IF(IC.EQ.3) RMULT=RMULT*(1.0-X(N))
33   FINT=FINT+R(N)*F(X1,X2,X3,01,U2,U3,T,TIME,T0,IPR,IC)*
34   * UIJ(I,IO,-1.0,1.0,T,IC)*UIJ(J,IO,-1.0,1.0,1,IC)*RMULT
35 13 CONTINUE
36 C*
37 RETURN
38 END
39
40
41
42
43
44
45
46
47
48
49
50
51
52

```

```

1  C* BASE=====
2  C**
3  C**
4  C**      AQUI TENGO LAS FUNCIONES BASE Y SUS DERIVADAS:
5  C**      SI IC ES 1 TENGO FUNCIONES BASE USUALES;
6  C**      SI IC ES 2 TENGO FUNCIONES BASE ESPECIALES IZQUIERDAS
7  C**      SI IC ES 3 TENGO FUNCIONES BASE ESPECIALES DERECHAS
8  C**
9  C* =====
10 FUNCTION UIJ(J, I0, X1, X2, XC, IC)
11 C*
12     IUP=I0+1
13     RA2=SQRT(2, J)
14     JP=J+1
15     H=X2-X1
16     H2=H**2
17     GO TO (1, 100, 200), IC
18     1 GO TO (10, 50), IUP
19 C*
20     10 GO TO (20, 30, 35, 40), JP
21 C*
22     20 UIJ=1.
23     RETURN
24     30 UIJ=(2.*XC-X1-X2)*(XC-X2)/H2
25     RETURN
26     35 UIJ=4.*(XC-X1)*(X2-XC)/H2
27     RETURN
28     40 UIJ=(XC-X1)*(4.*X2-X1-X2)/H2
29     RETURN
30 C*
31     50 GO TO (60, 60, 65, 70), JP
32 C*
33     60 UIJ=(4.*XC-X1-3.*A2)/H2
34     RETURN
35     65 UIJ=4.*(X1+X2-2.*AC)/H2
36     RETURN
37     70 UIJ=(4.*XC-5.*X1-A2)/H2
38     RETURN
39 C*
40     100 TP1=XC + 1.
41     GO TO (110, 150), IUP
42     110 GO TO (120, 130, 135, 140), JP
43     120 UIJ=1.
44     RETURN
45     130 UIJ=(TP1 - (1.+RA2)*SQRT(TP1) + RA2)/RA2
46     RETURN
47     135 UIJ=- (1.+RA2)*(TP1 - RA2*SQRT(TP1))
48     RETURN
49     140 UIJ=((1.+RA2)/RA2)*(TP1 - SQRT(TP1))
50     RETURN
51 C*
52     150 GO TO (160, 160, 165, 170), JP
53     160 UIJ=(1.- (1.+RA2)/(2*SQRT(TP1)))/RA2
54     RETURN
55     165 UIJ=- (1.+RA2)*(1.-1./SQRT(2*TP1))
56     RETURN
57     170 UIJ=((1.+RA2)/RA2)*(1.-1./(2*SQRT(TP1)))
58     RETURN

```

```
59 C*
60 C*
61 200 TM1=1.-XC
62 GU TO (210,250),IUP
63 210 GU TO (220,230,230,240),JP
64 220 UIJ=1.
65 RETURN
66 230 UIJ=((1.+RA2)/RA2)*(TM1*SQRT(IM1))
67 RETURN
68 235 UIJ=-RA2+1.)*(IM1*RA2*SQRT(TM1))
69 RETURN
70 240 UIJ=(IM1 + RA2 - (1.+RA2)*SQRT(TM1))/RA2
71 RETURN
72 C*
73 C*
74 250 GU TO (260,260,260,270),JP
75 260 UIJ=((1.+RA2)/RA2)*(-1.+1./(2*SQRT(IM1)))
76 RETURN
77 265 UIJ=-RA2+1.)*(-1.+1./SQRT(2*IM1))
78 RETURN
79 270 UIJ=(-1.+((1.+RA2)/(2.*SQRT(TM1))))/RA2
80 RETURN
81 END
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
```

```
1 C* CALC=====
2 C*
3 C*
4 C*   AQUI, DADAS LAS FUNCIONES BASE Y LOS TRES PUNTOS EN EL
5 C*   INTERVALO PERTINENTE, CALCULO XA Y UA.
6 C*   IC INDICA QUE TENGO FUNCIONES BASE ESPECIALES
7 C*
8 C* =====
9 SUBROUTINE CALC(X1,X2,X3,U1,U2,U3,T,TIME,I0,XA,UA,IC)
10 DIMENSION UO(3)
11 DO 11 IYX=1,3
12   UU(IYX)=OIJ(IYA,U,-1.0,1.0,T,IC)
13 11 CONTINUE
14 XA=UU(1)*X1 + UU(2)*X2 + UU(3)*X3
15 UA=UU(1)*U1 + UU(2)*U2 + UU(3)*U3
16 RETURN
17 END
```

```

1 C* CHECK=====
2 C*
3 C*
4 C* COMPARA U CON UNEW ... ES DELIX LA ESTIMACION ANTERIOR
5 C* CON LA NUEVA; Y SI LA MAYOR DIFERENCIA ES MENOR QUE
6 C* EPSI, EPSR O SI YA DICE EL NUMERO MAXIMO DE ITERACIONES
7 C* DESEADO, DETIENE EL PROCESO;
8 C*
9 C*
10 C* =====
11 SUBROUTINE CHECK(U, UNEW, EPSI, EPSR, I, LAG, NP, LIMIT, ITER)
12 DIMENSION U(NP), UNEW(NP)
13 IFLAG=0
14 SMAX=0.
15 RMAX=0.
16 DO 10 N=1, NP
17 TEMP=ABS(U(N) - UNEW(N))
18 WRITE (9,111) TEMP, U(N), UNEW(N), N, ITER
19 111 FORMAT("TEMP",D, UNEW,N",3F8.5," ",2I2)
20 IF (TEMP .GT. RMAX) RMAX=TEMP
21 ISMP=ABS(UNEW(N))
22 IF (ISMP .GT. SMAX) SMAX = TSMP
23 10 CONTINUE
24 EPS=EPSI+EPSR*SMA
25 WRITE(9,112) RMAX, SMAX, EPS, ITER, LIMIT
26 112 FORMAT("RMAX, SMAX, EPS, ITER, LIMIT",3G15.8,2I2)
27 IF ( (RMAX .LT. EPS) .OR. (ITER .GE. LIMIT) ) GOTO 20
28 GOTO 30
29 IFLAG=1
30 CONTINUE
31 DO 15 J=1, NP
32 15 U(J)=UNEW(J)
33 RETURN
34 END
35
36
37
38
39
40
41
42
43
44
45
46
47

```



```

1 C* FACTOR=====
2 C*
3 C*
4 C*
5 C*   ESTA RUTINA FACTORIZA MATRICES SINGULARES BANDADAS.
6 C*
7 C*
8 C* =====
9 SUBROUTINE FACTOR(A,AF,N,M,K)
10 DIMENSION A(N,M),AF(N,M)
11 DO 4 K=1, N
12   DO 2 J=1, M
13     AF(K,J) = A(K,J)
14   CONTINUE
15 2 CONTINUE
16 DO 90 K=1, N
17   IF (A(K,M) .EQ. 0.0) GO TO 90
18   IF (N .LE. K) GO TO 90
19   L = K + M - 2
20   IF (L .GE. N) L = N - 1
21   DO 9 J = K, L
22     DO 8 I = A(J,K)
23       IF (I .EQ. 0) GO TO 8
24       AF(J+1, I-J+M) = AF(J+1, I-J+M) - AF(I+1, K+M-I-1)*
25         * AF(J+1, K+M-J-1)
26     CONTINUE
27     AA = AF(J+1, K+M-J-1)/AF(K,M)
28     AF(J+1, M) = AF(J+1, M) - AA*AF(J+1, K+M-J-1)
29     AF(J+1, K+M-J-1) = AA
30 9 CONTINUE
31 K = N+1
32 RETURN
33 END
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51

```



```

1 C* ERROR2=====
2 C*
3 C* AQUI VEO EL VALOR DE LA SOLUCION EN LOS PUNTOS VADOS
4 C* PARA LA APROXIMACION POR FEM? LOS COMPARO CON LOS
5 C* CALCULADOS DE MANERA EXACTA, OBTENIENDO EL MAXIMO
6 C* ERROR2 Y TAMBIEN EL ERROR2 L2.
7 C*
8 C* =====
9 C*
10 C* SUBROUTINE ERROR2 (TIME,TO,U,NI,ILBC,X1,X2,PU,IPR,NIMP,H,NP,LEE)
11 C*
12 C* DIMENSION U(NP),UU(3),PU(3,NP)
13 C*
14 XMAX = 0.0
15 E=0.
16 E2=0.
17 NIME=((NIMP - 1)/NI) + 1
18 IF (NIME.NE.1) GO TO 11
19 NIM=NIMP+1
20 NIME=NIME+1
21 11 CONTINUE
22 HI=2.0/FLOAT(NIME - 1)
23 DO 100 N=1,NI
24 IND1=(N-1)*2+ILBC
25 IND2=IND1+1
26 IND3=IND2+1
27 X1=PU(1,N)
28 X2=PU(2,N)
29 X3=PU(3,N)
30 U1=0.
31 IF (IND1.LE.1) U1=U(IND1)
32 U2=U(IND2)
33 U3=0.
34 IF (IND3.LE.NP) U3=U(IND3)
35 IC=1
36 IF (N.EQ.1) IC=2
37 IF (N.EQ.NI) IC=3
38 DO 101 M=1,NIMP
39 TIMP=-1.0 + (M - 1)*HI
40 CALL CACC(X1,X2,X3,U1,U2,U3,TIMP,TIME,TO,XA,UA,IC)
41 UE=FE(XA,TIMP,TO,IPR)
42 ER=UE-UA
43 IF (ABS(ER).GT.E) XMAX=XA
44 E=AMAX1(ABS(ER),E)
45 IF ((TIMP.LE.-1.0).AND.(IND1.NE.1)) GO TO 99
46 E2=E2+ER**2
47 WRITE(6,112) XA,UE,UA,ER
48 112 FORMAT("ERR",4F8.3)
49 99 CONTINUE
50 101 CONTINUE
51 100 CONTINUE
52 E2=SQRT(E2/NIMP)
53 WRITE(7,1200) E,XMAX,E2
54 WRITE(6,1200) E,XMAX,E2
55 1200 FORMAT(/" EL ERROR2 MAXIMO ES",1PE10.3,2A,"EN X=",E10.3//
56 * " EL ERROR2 L2 ES",E10.3)
57 C*
58 RETURN

```

110/21/85) (ISAX)PROGRAMA/AD/1 UN IIMAS.

PAGINA: 12-2

END

59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77

```

1 C* INICIA=====
2 C*
3 C*
4 C* AQUI CALCULO EL VALOR INICIAL DE MI SOLUCION BASANDOME
5 C* EN LA SOLUCION EXACTA, ADEMÁS DE LA MATRIZ DE MASA
6 C* LA CUAL ES CONSTANTE:
7 C*
8 C*
9 C*
10 C*
11 C* =====
12 C* SUBROUTINE INICIA(U, NU, TIME, TU, LEE, NMA, NI, ILBC, IRBC,
13 C* NP, IPR, M, IESP)
14 C*
15 C* DIMENSION U(NP), PU(3, NP), RMA(NP, M), IND(3)
16 C* EXTERNAL UNU
17 C*
18 C* DO 20 N=1, NP
19 C* DO 10 MM=1, M
20 C* RMA(N, MM)=0.
21 C* 10 CONTINUE
22 C* 20 CONTINUE
23 C*
24 C* DO 101 N=1, NI
25 C* X1=PU(1, N)
26 C* X2=PU(2, N)
27 C* X3=PU(3, N)
28 C* IND(1)=(N-1)*2+ILBC
29 C* IND(2)=IND(1)+1
30 C* IND(3)=IND(2)+1
31 C* IND1=IND(1)
32 C* IND2=IND(2)
33 C* IND3=IND(3)
34 C* IMIN=1
35 C* IMAX=3
36 C* IF (N.EQ.1 .AND. ILBC.EQ.0) IMIN=2
37 C* IF (N.EQ.NI .AND. IRBC.EQ.0) IMAX=2
38 C* IF (IND1.EQ.1) U(IND1)=FEX(X1, TIME, T0, IPR)
39 C* U(IND2)=FEX(X2, TIME, T0, IPR)
40 C* IF (IND3.LE.NP) U(IND3)=FEX(X3, TIME, T0, IPR)
41 C* IF (IESP.EQ.0) GO TO 200
42 C* IC=1
43 C* IF (N.EQ.1) IC=2
44 C* IF (N.EQ.NI) IC=3
45 C* 200 DO 30 J=IMIN, IMAX
46 C* II=IND(J)
47 C* DO 201 J=IMIN, 1
48 C* JJ=M-1+J
49 C* RRR=FINF(TIME, T0, U1, U2, U3, X1, X2, X3, UNU, 1, J, 0, IPR, IC)
50 C* RMA(II, JJ)=RMA(II, JJ)+RRR
51 C* 201 CONTINUE
52 C* 30 CONTINUE
53 C* 100 CONTINUE
54 C* =====
55 C*
56 C* 101 CONTINUE
57 C* RETURN
58 C* END

```

59
60
61
62
63
64
65
66
67
68
69
70
71
72
73

1
2
3
4
5
6
7

C*
C*
C*

```

FUNCTION UNU=====
=====
FUNCTION UNU(X1,X2,X3,U1,U2,U3,T,TIME,TU,PA,IC)
UNO=1
RETURN
END

```

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
31
32

```

C* FUNCTION TIM=====
C*
C*
=====
FUNCTION TIM(A1,X2,X3,XIMP)
  XX1=XIMP-X2
  XX2=X3-X1
  XX3=X1-2*X2+A3
  IOMP=XX1*0.5*XA3 + (XX2*2)/16.
  IF (TOMP .EQ. 0.) TOMP=0.
  TOMP=SQRT(TOMP) + XX2*0.25
  TIM=XX1/IOMP
RETURN
END

```

ARCHIVO: (ISAX)PROGRAMA/2D ON IIMAS.
20 PAGINAS, 1460 LINEAS.
1170 REGISTROS.

FECHA: (10/21/83); 15:44 HRS.
ULTIMO CAMBIO: (10/13/83).

PAGINAS: 1-ENU.

** E SPOOLER ** VERSION 3
DEPTO. DE COMPUTACION
I- I. M. A. S. " U. N. A. M.

DIRECCIONARIO DE PAGINAS.

CONTENIDO DE LA PRIMERA LINEA

PAGINA

LINEAS

1 47
 2 127
 3 130
 4 152
 5 44
 6 29
 7 52
 8 55
 9 123
 10 52
 11 80
 12 63
 13 20
 14 87
 15 53
 16 54
 17 241
 18 7
 19 18
 20 26

C* COMMENT=====PROGRAMA/20 ABRIL 28 1983=====

C* MAIN=====

C* STURN=====

C* SIFF=====

C* FUNCIONES=====

 FUNCTION W(X1,Y1,U1,T1,T2,TIME,IO,IPR)

 FUNCTION F(X1,Y1,U1,T1,T2,TIME,IO,IPR)

 FUNCTION FEX(XA,YA,TIME,IO,IPR)

C* ERROR=====

C* SUBROUTINE EVAL=====

C* FIBI=====

C* BASE=====

C* CALC=====

C* CHECK=====

C* FACTOR=====

C* SOL=====

C* INICIA=====

 FUNCTION IXX(NX,LM)

 FUNCTION IYY(NX,LM)

C* FUNCTION UNO=====


```

1 C* MAIN=====
2 C*
3 C*
4 C* PROGRAM PRINCIPAL:
5 C*
6 C*
7 C*
8 C* =====
9 10 CONTINUE
10 READ(5,/) LEE,NUMV,XI,YI,XF,YF
11 C* WRITE(6,*) LEE,NUMV,XI,YI,XF,YF
12 C* WRITE(7,*) LEE,NUMV,XI,YI,XF,YF
13 C* WRITE(6,*) LEE,NUMV,XI,YI,XF,YF,IT,JI,ITN,JTN
14 712 FORMAT(" LEE",2I2)
15 WRITE(6,712) LEE,NUMV
16 713 FORMAT(" XI",4("G15.8",))
17 IF (LEE.GT.7) GO TO 11
18 DO 1 IYZ=1,NUMV
19 READ(5,/) NI,NJ,ICO,ICC
20 HI=(XF-XI)/FLOA(NI)
21 HJ=(YF-YI)/FLOA(NJ)
22 C* WRITE(6,*) HI,HJ,NI,NJ,ICO
23 C* WRITE(7,*) HI,HJ,NI,NJ,ICO
24 WRITE(6,723) NI,NJ
25 723 FORMAT(" NI NJ",2," ",I2)
26 WRITE(6,732) ICO(1),ICO(2),ICO(3),ICO(4)
27 732 FORMAT(" ICO",4(" ",I2))
28 WRITE(6,733) ICC(1),ICC(2),ICC(3),ICC(4)
29 733 FORMAT(" ICC",4(" ",I2))
30 C*
31 GO TO (25,66)LEE
32 23 DO 237 N=1,NI
33 RIM(N)=0.0
34 237 CONTINUE
35 DO 238 N=1,NJ
36 RJM(N)=0.0
37 238 CONTINUE
38 WRITE(6,660)
39 666 FORMAT(" PUNTO CUARTO")
40 IF (ICO(1).EQ.V) RIM(1)=0.25
41 IF (ICO(2).EQ.V) RIM(1)=0.75
42 IF (ICO(3).EQ.V) RJM(1)=0.25
43 IF (ICO(4).EQ.V) RJM(1)=0.75
44 239 DO 257 N=1,NI
45 PX(1,N)=XI+(N-1)*HI
46 PX(2,N)=PA(1,N)+RIM(N)*HI
47 PX(3,N)=PA(1,N)+HI
48 257 CONTINUE
49 DO 258 N=1,NJ
50 PY(1,N)=XI+(N-1)*HJ
51 PY(2,N)=PY(1,N)+RJM(N)*HJ
52 PY(3,N)=PT(1,N)+HJ
53 258 CONTINUE
54 WRITE(6,321) (PX(1,LJ),PA(2,LJ),PX(3,LJ),LJ,LJ=1,NI)
55 WRITE(6,322) (PY(1,LJ),PT(2,LJ),PY(3,LJ),LJ,LJ=1,NJ)
56 321 FORMAT(" PA",3G15.8," ",I2)
57 322 FORMAT(" PT",3G15.8," ",I2)
58 GO TO 280

```

```

59      26      CONTINUE
60      HI1=HI/4.
61      HI2=HI/2.
62      IF (IC0(1) .EQ.0) NI=NI+4
63      I2=I1-1
64      I3=I2-1
65      IF (IC0(5) .EQ.0) NJ=NJ+4
66      J2=NJ-1
67      J3=J2-1
68      PX(1,1)=XI
69      HJ1=HJ/4.
70      HJ2=HJ/2.
71      PY(1,1)=TI
72      WRITE(6,661)
73      667      FORMAT(" *EFINAMIENIO")
74      DO 265 IX=1,NI
75      HHI=HI
76      IF ( (IX.LE.2) .AND. (IC0(1).EQ.0) ) HHI=HI1
77      IF ( (IX.GE.12) .AND. (IC0(2).EQ.0) ) HHI=HI1
78      IF ( (IX.EQ.3) .AND. (IC0(1).EQ.0) ) HHI=HI2
79      IF ( (IX.EQ.13) .AND. (IC0(2).EQ.0) ) HHI=HI2
80      IF (IX.GE.1) PX(1,IX)=PX(3,IX-1)
81      PX(2,IX)=PA(1,IX) + (0.5*HHI)
82      PX(3,IX)=PA(1,IX) + HHI
83      265      CONTINUE
84      DO 266 IY=1,NJ
85      HHJ=HJ
86      IF ( (IY.LE.2) .AND. (IC0(3).EQ.0) ) HHJ=HJ1
87      IF ( (IY.GE.12) .AND. (IC0(4).EQ.0) ) HHJ=HJ1
88      IF ( (IY.EQ.3) .AND. (IC0(3).EQ.0) ) HHJ=HJ2
89      IF ( (IY.EQ.13) .AND. (IC0(4).EQ.0) ) HHJ=HJ2
90      IF (IY.GE.1) PY(1,IY)=PY(3,IY-1)
91      PY(2,IY)=PT(1,IY) + (0.5*HHJ)
92      PY(3,IY)=PT(1,IY) + HHJ
93      266      CONTINUE
94      WRITE(6,321) (PX(1,LJ),PA(2,LJ),PX(3,LJ),LJ,LJ=1,NI)
95      WRITE(6,322) (PY(1,LJ),PT(2,LJ),PY(3,LJ),LJ,LJ=1,NJ)
96      280      CONTINUE
97      NX=NI*2-1+IC0(1)+IC0(2)
98      LA=NX+3
99      NY=NJ*2-1+IC0(3)+IC0(4)
100     NP=NX*NY
101     M=2*NX+1
102     CALL STUKM(L,E,NI,NJ,PX,PY,IC0,ICC,MP,NX,NY,MA,
103     * A,U,B,AF,AP,AP,XP,YI,YF,PI,HJ,H,IT,JI,ITN,JTN)
104     1      CONTINUE
105     GO TO 10
106     11     LOCK 6
107     LOCK 7
108     CALL EXIT
109
110     C*
111     C*
112     END
113
114
115
116

```

117
118
119
120
121
122
123
124
125
126
127

```

1 C* STURM=====
2 C*
3 C*
4 C*
5 C*   ESTA RUTINA CONTROLA EL PROCESO EN GENERAL;
6 C*   LLAMA A TODAS LAS DEMAS RUTINAS CUANDO SE
7 C*   REQUIEREN. ES CASI IDENTICA A LA DEL PROG. EN 10
8 C*
9 C*
10 C* =====
11 C*   SUBROUTINE STURM (LEE,NI,NJ,IPK,PI,ICO,ICC,N,NA,NT,HA,A,U,B,AF,AP,
12 C*   * XI,XF,YI,YF,HI,HJ,W,I,JI,IT,ITN)
13 C*
14 C*   DIMENSION PX(2,NI),PY(3,NJ),U(NP),A(NP,M),B(NP),ICC(4),UX(2500)
15 C*   REAL MA(NP,M)
16 C*   DIMENSION UAU(2000),UNEW(5000),AF(NP,NP),AP(NP,NP),ICO(4)
17 C*   DIMENSION II(9),JI(6),IIN(6),JTN(6),C(5000),UNE(5000)
18 C*
19 C*
20 C*   READ (5,/) TIMEU,TIMEF,DELTAI,THETA
21 C*   WRITE(7,*) TIMEU,TIMEF,DELTAI,THETA
22 C*   WRITE (6,521) TIMEU,TIMEF
23 C*   521 FORMAT(" TIMEU, TIMEF ",G15.8)
24 C*   WRITE (6,522) DELTAI,THETA
25 C*   522 FORMAT(" DELTAI, THETA ",G15.8)
26 C*   WRITE (6,*) TIMEU,TIMEF,DELTAI,THETA
27 C*   READ (5,/) IPR,NIMP,MIMP,EPSP,EPSPR,LIMI!
28 C*   WRITE(6,937) IPR
29 C*   937 FORMAT(6 PRUEBA NUMERO ",14)
30 C*   WRITE(7,*) IPR,NIMP,EPSP,EPSPR,LIMIT
31 C*   WRITE (6,*) IPR,NIMP,EPSP,EPSPR,LIMIT
32 C*   * WRITE (6,*) IPR,NIMP,EPSP,EPSPR,LIMI!
33 C*   DO 11 N=1, NP
34 C*     U(N)=0.51
35 C*   11 CONTINUE
36 C*   TIME = TIMEU
37 C*   TU=-1
38 C*   CALL INICIALU,PA,PY,TIME,T0,LEE,MA,NI,NJ,ICO,ICC,
39 C*   * HF,NX,NY,IPK,M,IT,JI,IIN,JTN)
40 C*   WRITE(6,9991)
41 C*   9991 FORMAT(" INICIO ")
42 C*   DO 79 I=1,NP
43 C*     UNE(I)=U(I)
44 C*   79 CONTINUE
45 C*   IF (TIME.EQ.TIME0) GO TO 81
46 C*   WRITE(6,9952)
47 C*   9952 FORMAT (" ENTRE A ERROR ")
48 C*   121 CONTINUE
49 C*   DO 10 N=1, NP
50 C*     WRITE(6,321) N,UNE(N),UNEW(N)
51 C*   10 CONTINUE
52 C*   321 FORMAT(" N,U,UNEW,TEMP",14,2(" ",G15.8))
53 C*   CALL ERROR(TIME,TU,U,NI,NJ,ICO,ICC,AF,XI,XF,II,YF,PX,PY,IPR,
54 C*   * NIMP,MIMP,NI,HJ,NP,NX,NY,LEE,EPSP,IT,JI,IIN,JTN)
55 C*   81 CONTINUE
56 C*   IER = 0
57 C*   WRITE(6,123) TIME
58 C*   123 FORMAT(" TIME ",G15.8)
59 C*   IF (TIME.GE. TIMEF) GO TO 100

```

```

59      DI=DELTAT*THEIA
60      DIT=DELTAT*(1-THETA)
61      CALL STIFF(UNEN,PA,PY,TIME,TO,LEE,A,B,
62      * HI,HJ,N1,N2,ICO,ICL,NI,NA,NY,IPR,M,I,IT,JT,ITN,JTN)
63      WRITE(6,9992)
64      9992  FORMAT(" STIFF UNO ")
65      DO 1020 I = 1,NP
66      MM = AMAX1(I-(M-1),1)
67      UAUX(I) = D11*P(I)
68      DO 1010 J = MM,I
69      JIM=J+TH
70      UAUX(I) = UAUX(I) - D11*A(I,JIM)*U(J) + MA(1,JIM)*U(J)
71      IF (J.LT=I) UAUX(J) = UAUX(J) - DIT*A(I,JIM)*U(I) +
72      *      ME(1,JIM)*U(I)
73      1010  CONTINUE
74      1020  CONTINUE
75      IFLAG=0
76      TIME = TIME + DELTAT
77      330  CONTINUE
78      DO 1030 I = 1,NP
79      MM=MAX1(I-(M-1),1)
80      DO 1040 JIM=1,M
81      AIJ=A(I,JIM)
82      A(I,JIM)=MA(I,JIM) + DT*AIJ
83      1040  CONTINUE
84      B(I)=UAUX(I) + DT*B(I)
85      1050  CONTINUE
86      WRITE(6,9993)
87      9993  FORMAT(" FACIOM ")
88      MP=M
89      CALL FACIOM(A,AF,NP,MP,KJ
90      *      WRITE(6,*) K,MP
91      *      IF (K.EE.NP) GO TO 334
92      WRITE(6,9994)
93      9994  FORMAT(" SOC ")
94      CALL SOC(AF,B,UNEN,NP,MP)
95      C*==========
96      IFLAG=0
97      WRITE(6,9345)
98      9345  FORMAT(" CHECK JFLAG ")
99      CALL CHECK(U,UNEN,EPSI,EP,SK,JFLAG,NP,NA,NY,LIMIT,ITER)
100      ITER=ITER+1
101      WRITE(6,124) JFLAG,ITER
102      124  FORMAT(" JFLAG/ITER",I2,I2)
103      IF (JFLAG.EE.1) GO TO 400
104      CALL STIFF(U,PA,PY,TIME,TO,LEE,A,B,
105      *      HI,HJ,N1,N2,ICO,ICL,NI,NA,NY,IPR,M,I,IT,JT,ITN,JTN)
106      WRITE(6,9993)
107      9993  FORMAT(" STIFF DOS ")
108      GO TO 330
109      800  GO TO 80
110      354  WRITE(6,1111)
111      1111  FORMAT(" HURMOM ")
112      WRITE(7,1111)
113      LOCK 6
114      LOCK 7
115      CALL EXIT
116      100  CONTINUE

```



```
117 C*  
118 RETURN  
119 END  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130
```

```

1  C* STIFF=====
2  C*
3  C*
4  C* ESIA RUTINA SE ENCARGA DE CALCULAR A Y B, SIENDO
5  C* ENTONCES LA PARTE EN QUE SE UTILIZA FFM.
6  C*
7  C*
8  C*
9  C*
10 C*
11 C*
12 C* SUBROUTINE STIFF(U, PX, PY, IT, F, TO, I, E, A, B,
13 C* HT, HJ, NI, NJ, ICO, ICC, IC, NX, NY, IPR, M, IT, JT, IIN, JTN)
14 C*
15 C* REAL HP
16 C* DIMENSION U(NA, NY), PX(3, NI), PY(3, NJ), A(NP, M), B(NP), ICO(4), ICC(4),
17 C* DIMENSION US(8), UP(8), IND(3), JND(3), X(6), Y(6), XN(6), YN(6)
18 C* DIMENSION IT(6), JT(6), IIN(6), JTN(6)
19 C* EXTERNAL P, U,
20 C*
21 C* DO 20 N=1, NP
22 C*   B(N)=0.
23 C*   DO 10 MN=1, M
24 C*     A(N, MN)=0.
25 C*   CONTINUE
26 C* CONTINUE
27 C*
28 C* DO 101 N=1, NI
29 C*   WRITE(6, *) N
30 C*   DO 101 NN=1, NJ
31 C*     RMP=( PY(1, NN) - PY(5, NN) ) / ( PX(3, N) - PX(1, N) )
32 C*     BB=- ( RMP*PA(3, N) - PY(1, NN) )
33 C*     DO 3 IXZ=176
34 C*       IT1=I1(1XZ)
35 C*       IT2=JT(1XZ)
36 C*       IT3=IIN(1XZ)
37 C*       IT4=JTN(1XZ)
38 C*       X(1XZ)=PX(IT1, N)
39 C*       Y(1XZ)=PY(IT2, NN)
40 C*       XN(1XZ)=PA(IT3, N)
41 C*       YN(1XZ)=PT(IT4, NN)
42 C*
43 C*   3 CONTINUE
44 C*   XT1=X(1)+0.45*PI
45 C*   XT2=X(2)-0.45*PI
46 C*   IF ((X(4).GT.X(1)).AND.(X(4).LT.X(2))) GO TO 1
47 C*   Y(4)=RMP*X(4) + BB
48 C*   GO TO 2
49 C*
50 C*   1 X(4)=(Y(4)-BB)/RMP
51 C*   XN(4)=X(4)
52 C*   YN(4)=Y(4)
53 C*   IN1=(N-1)*2+ICC(1)
54 C*   IN2=IN1+1
55 C*   IN3=IN2+1
56 C*   JN1=(NN-1)*2+ICC(3)
57 C*   JN2=JN1+1
58 C*   JN3=JN2+1
59 C*   IND(1)=IN1
60 C*   IND(2)=IN2
61 C*   IND(3)=IN3
62 C*   JND(1)=JN1
63 C*   JND(2)=JN2

```

```

59      JND(3)=JN3
60      UR(1)=0.
61      UR(2)=0.
62      UR(3)=0.
63      UR(5)=0.
64      UR(6)=0.
65      US(1)=0.
66      US(2)=0.
67      US(3)=0.
68      US(5)=0.
69      US(6)=0.
70      IF (JN1.LE.0) GO TO 40
71      IF (IN1.G1.0) UP(I)=U(IN1,JN1)
72      UR(6)=U(IN1,JN1)
73      IF (IN3.G1.NX) GO TO 40
74      UR(2)=U(IN3,JN1)
75      US(2)=UB(2)
76      40      IF (IM1.LE.0) GO TO 41
77      UR(5)=U(IM1,JN2)
78      IF (JN3.G1.NY) GO TO 41
79      UR(3)=U(IM1,JN3)
80      US(3)=UB(3)
81      41      UR(4)=U(IM1,JN2)
82      US(4)=UB(4)
83      IF (IN3.LE.NX) US(6)=U(IN3,JN2)
84      IF (IN3.G1.NY) GO TO 42
85      US(5)=U(IM1,JN3)
86      IF (IN3.LE.NX) US(1)=U(IN3,JN3)
87      143  FORMAT(' RR1,IJ,KL',G15.8,' ',2(' ',I2))
88      243  FORMAT(' RR2,IJ,KL',G15.8,' ',2(' ',I2))
89      141  FORMAT(3(' X',G15.8))
90      142  FORMAT(3(' Y',G15.8))
91      241  FORMAT(3(' XH',G15.8))
92      242  FORMAT(3(' YH',G15.8))
93      42  DO 30 IJ=1,6
94          I1=IT(IJ)
95          IM1=IND(I1)
96          IF ((IM1.LI.1).OR.(IM1.G1.NX)) GO TO 30
97          J1=JT(IJ)
98          JN1=JND(J1)
99          IF ((JN1.LI.1).OR.(JN1.G1.NY)) GO TO 30
100         II1=(I1 + JN1-1)*NX
101         DO 31 KL=1,IJ
102             K1=IT(KL)
103             KM1=IND(K1)
104             IF ((KM1.LI.1).OR.(KM1.GT.NX)) GO TO 31
105             L1=JT(KL)
106             LM1=JND(L1)
107             IF ((LM1.LI.1).OR.(LM1.GT.NY)) GO TO 31
108             JJ1=KM1 + (LM1-1)*NX
109             II=MAX(II1,JJ1)
110             JJT=IABD(II1-JJ1)
111             JJ=M-JJT
112             RR1=FINT(IME,TO,US,X,Y,P,IJ,KL,1,IPR)
113             SS1=FINT(IME,TO,US,X,Y,Q,IJ,KL,0,IPR)
114             A(II,JJ)=A(II1,JJ) + RR1 + SS1
115
116      31  CONTINUE
          TT1=FINT(IME,IO,US,X,Y,P,IJ,0,0,IPR)

```

```

59      JND(3)=JN3
60      UB(1)=0.
61      UB(2)=0.
62      UB(3)=0.
63      UB(5)=0.
64      UB(6)=0.
65      US(1)=0.
66      US(2)=0.
67      US(3)=0.
68      US(5)=0.
69      US(6)=0.
70      IF (JN1.LE.U) GO TO 40
71      IF (IN1.G!.U) U(1)=U(IN1,JN1)
72      UB(6)=U(IN1,JN1)
73      IF (IN3.G!.NX) GO TO 40
74      UB(2)=U(IN1,JN1)
75      US(2)=UB(2)
76      40  IF (IN1.LE.U) GO TO 41
77      UB(5)=U(IN1,JN1)
78      IF (JN3.G!.NY) GO TO 41
79      UB(3)=U(IN1,JN2)
80      US(3)=UB(3)
81      41  UB(4)=U(IN1,JN1)
82      US(4)=UB(4)
83      IF (IN3.LE.NX) US(6)=U(IN3,JN2)
84      IF (JN3.G!.NY) GO TO 42
85      US(5)=U(IN1,JN3)
86      IF (IN3.LE.NX) US(1)=U(IN3,JN3)
143  243  141  142  241  242  42  DO 30 IJ=1,6
94      II=IT(IJ)
95      IM1=IND(IJ)
96      IF ((IM1.LI-1).OR.(IM1.G!.NX)) GO TO 30
97      JI=JT(IJ)
98      JM1=JND(JI)
99      IF ((JM1.LI-1).OR.(JM1.G!.NY)) GO TO 30
100     II=II1 + (JM1-1)*NX
101     DO 31 KL=1,IJ
102     KI=IT(KL)
103     KM1=IND(KI)
104     IF ((KM1.LI-1).OR.(KM1.GT.NX)) GO TO 31
105     LI=JT(KL)
106     LM1=JND(LI)
107     IF ((LM1.LI-1).OR.(LM1.GT.NY)) GO TO 31
108     JJ1=KM1 + (LM1-1)*NX
109     II=MAX(II1,JJ1)
110     JJT=TAB3(II1-JJ1)
111     JJ=MAX(JJT)
112     RR1=FINT(IME,TO,US,X,Y,P,IJ,KL,1,IPR)
113     SS1=FINT(IME,TO,US,X,Y,Q,IJ,KL,0,IPR)
114     A(II,JJ)=A(11,JJ) + RM1 + SS1
115     31  CONTINUE
116     TT1=FINT(IME,0,US,X,Y,I,IJ,0,0,IPR)

```

```

117      B(II)=B(II)+IT1
118      30 CONTINUE
119      DO 52 I1=1,6
120          IP=I1N(I1)
121          IM2=I1M(I2)
122          IF ((IM2.LI.1).OR.(IM2.GI.NX)) GO TO 34
123          J2=J1N(I2)
124          JM2=J1M(J2)
125          IF ((JM2.LI.1).OR.(JM2.GI.NY)) GO TO 34
126          IP2=IP2 + (JM2-1)*IX
127          DO 33 KL=1,IJ
128              KP=I1N(KL)
129              KM2=I1M(KL2)
130              IF ((KM2.LI.1).OR.(KM2.GI.NX)) GO TO 33
131              LP=J1N(KL)
132              LM2=J1M(KL2)
133              IF ((LM2.LI.1).OR.(LM2.GI.NY)) GO TO 33
134              JJ2=KM2 + (LM2-1)*IX
135              II=MAX(II,II2,JJ2)
136              JJT=I1M(II2-JJ2)
137              JJ=M-JJT
138              RR2=FINI(TIME,TO,UB,XN,YN,P,IJ,KL,1,IPR)
139              SS2=FINI(TIME,TO,UB,XC,YN,P,IJ,KL,0,IPR)
140              A(II,JJ)=A(II,JJ) + RR2**SS2
141          33 CONTINUE
142          IT2=FINI(TIME,TO,UB,XN,YN,P,IJ,0,0,IPR)
143          B(II)=B(II)+IT2
144      32 CONTINUE
145      100 CONTINUE
146      C* =====
147      C*
148      101 CONTINUE
149      WRITE(6,105)
150      105 FORMAT (" A/B EN STIFF")
151      RETURN
152      END

```

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
31
32
33
34
35
36
37
38
39
40
41
42
43
44

```

C* FUNCIONES=====
C*
C* SIGUEN LAS FUNCIONES P, D, F Y FEX:
C* P, D Y F CORRESPONDEN A TALEN EN LA ECUACION
C* DU/DT= (P(U))' = U(U) + F
C*
C* FEX CALCULA LA SOLUCION EXACTA POR MOTIVOS DE
C* COMPARACION.
C*
C* =====
C* FUNCTION P(XI, YI, UI, T1, T2, TIME, TU, T, R)
C* DIMENSION X1(6), Y1(6), U1(6)
C* ISP=1
C* CALL CALC(XI, YI, UI, T1, T2, TIME, T0, XA, YA, UA, ISP)
C* GO TO(100, 200, 300, 400, 500, 600, 700, 800, 900), IPR
C*
100 P=1.
   RETURN
C*
200 P=1.
   RETURN
C*
300 P=UA
   RETURN
C*
400 P=UA
   RETURN
500 CONTINUE
   P=1.
   RETURN
600 CONTINUE
   P=UA
   RETURN
700 CONTINUE
   P=UA
   RETURN
800 CONTINUE
   P=1.
   RETURN
900 CONTINUE
   P=UA
   RETURN
   END

```

```
1      FUNCTION Q(X1,Y1,U1,T1,T2,TIME,T0,IPR)
2      DIMENSION XI(6),YI(6),UI(6)
3      ISP=1
4      CALL CALC(XI,YI,U1,T1,T2,TIME,T0,XA,YA,UA,ISP)
5      DATA PI/3.1415926535898/
6      GO TO(100,200,300,400,500,600,700,800,900),IPR
7
8      C*
9      100  Q=0.
10     RETURN
11
12     C*
13     200  Q=0.
14     RETURN
15
16     C*
17     300  Q=1.
18     RETURN
19
20     C*
21     400  Q=1./(TIME-10)
22     RETURN
23     500  Q=0.
24     RETURN
25     600  Q=0.
26     RETURN
27     700  Q=0.
28     RETURN
29     800  Q=-2.*(PI**2)
30     RETURN
31     900  Q=1.
32     RETURN
33     END
```

```

1      FUNCTION F(A1,Y1,U1,T1,I2,TIME,T0,IPR)
2      DIMENSION X1(6),Y1(6),U1(6)
3      ISP=1
4      CALL CALC(X1,I1,U1,T1,T0,TIME,T0,XA,YA,UA,ISP)
5      DATA PI/3.1415926535898/
6      GO TO(100,200,300,400,500,600,700,800,900),IPR
7
8      C* 100 F=(PI**2)*SIN(PI**A)
9      RETURN
10
11     C* 200 CONTINUE
12     F=(PI**2)*SIN(PI**A)
13     RETURN
14
15     C* 300 COSE=COS(PI*XA)
16     EXPZ=EXP(-2.*TIME)
17     F=(PI**2/8.)*EXPZ*SIN(PI*YA)*(2. + 8.*COSE + 5.*(COSE**2))
18     RETURN
19
20     C* 400 COSE=COS(PI*XA)
21     COSS = 1. + COSE/C.
22     SENE = SIN(PI*YA)
23     IIX = SENE*COSS
24     DUUX2 = ((PI*COSS(PI*YA)*COSS)**2)
25     DUY2 = ((PI*SENE**2)*SIN(PI*XA)/2. )**2)
26     DZUX = -(PI**2)*IIX
27     DZUY = -(PI**2)*SENE*COSE/2.
28     TEME=-1./((TIME-T0)**2)
29     F=-TEME*(DUUX2 + DUY2 + UX*(DZUX + DZUY))
30     RETURN
31
32     500 F=0.
33     RETURN
34
35     600 COSE=COS(PI*XA)
36     F=(PI**2/8.)*SIN(PI*YA)*(2. + 8.*COSE + 5.*(COSE**2))
37     RETURN
38
39     C* 700 COSE=COS(PI*XA)
40     COSS = 1. + COSE/C.
41     SENE = SIN(PI*YA)
42     IIX = SENE*COSS
43     DUUX2 = ((PI*COSS(PI*YA)*COSS)**2)
44     DUY2 = ((PI*SENE**2)*SIN(PI*XA)/2. )**2)
45     DZUX = -(PI**2)*IIX
46     DZUY = -(PI**2)*SENE*COSE/2.
47     F=- (DUUX2 + DUY2 + UX*(DZUX + DZUY))
48     RETURN
49
50     800 F=0.
51     RETURN
52
53     900 CONTINUE
54     F=9*((PI*UA)**2)/C. - 2*(PI**2)*EXP(-2.*TIME)*SIN(PI*XA)
55     RETURN
56     END

```



```

1      FUNCTION FEX(XA,YA,TIME,T0,IPK)
2
3      C*
4      DATA PI/3.1415926535898/
5
6      C*
7      GO TO (100,200,300,400,500,600,700,800,900),IPK
8
9      C*
10     100 FEX=SIN(PI*XA)
11         RETURN
12     200 FEX=SIN(PI*YA)
13         RETURN
14     300 COSC=1. + COS(PI*XA)/2.
15         EXPZ=EXP(-TIME)
16         SENE=SIN(PI*YA)
17         IF (SENE .LT. 0.) GO TO 301
18         SENE=SQRT(SENE)
19         FEX=EXPZ*SENE*COSC
20         GO TO 302
21     301 WRITE(6,341) XA,YA,SENE
22     341 FORMAT(" FAIL ON",3G15.8)
23         LOCK 6
24         CALL EXIT
25     302 CONTINUE
26         RETURN
27
28     C*
29     400 SEN=SIN(PI*YA)
30     COSC=1. + COS(PI*XA)/2.
31     FEX=SEN*COSC/(TIME-T0)
32     RETURN
33     500 FEX=SIN(PI*YA)*EXP(-2*(PI**2)*TIME)*COS(PI*XA)
34     RETURN
35     600 COSC=1. + COS(PI*XA)/2.
36     SENE=SIN(PI*YA)
37     IF (SENE .LT. 0.) GO TO 601
38     SENE=SQRT(SENE)
39     FEX=SENE*COSC
40     GO TO 602
41     601 WRITE(6,641) XA,YA,SENE
42     641 FORMAT(" FAIL ON",3G15.8)
43         LOCK 6
44         CALL EXIT
45     602 CONTINUE
46         RETURN
47
48     C*
49     700 SEN=SIN(PI*YA)
50     COSC=1. + COS(PI*XA)/2.
51     FEX=SEN*COSC
52     RETURN
53     800 FEX=SIN(PI*YA)*COS(PI*XA)
54     RETURN
55     900 CONTINUE
56     SEN=SIN(PI*XA)
57     IF (SEN .LT. 0.) CALL EXIT
58     FEX=EXP(-TIME)*SQRT(SEN)*COS(PI*YA)
59     RETURN
60     END

```

```

1 C* ERROR=====
2 C*
3 C*
4 C*
5 C* AQUI VEO EL VALOR DE LA SOLUCION EN LOS PUNTOS DADOS
6 C* PARA LA APROXIMACION POR FEM7 LOS COMPARO CON LOS
7 C* CALCULADOS DE MANERA EXACTA, OBTENIENDO EL MAXIMO
8 C* ERROR Y TAMBIEN EL ERROR L2.
9 C*
10 C* =====
11 C*
12 C* SUBROUTINE ERROK(TIME,TU,U,NI,NJ,ICU,ICL,AI,XF,YI,YF,PX,PY,IPK,
13 C* NIMP,MIMP,HI,HJ,MP,XX,YY,LEE,EPS,IT,JI,IIN,JIN)
14 C*
15 REAL MP
16 DIMENSION U(MA,MY),X(6),Y(6),PX(3,NI),PY(3,NJ),ICO(4),ICC(4)
17 DIMENSION UA(6),IND(3),JND(3),IT(6),JI(6),IIN(6),JIN(6)
18 C*
19 ISP=1
20 E=0
21 E2=0
22 NMAX=1
23 WRITE(6,800) AI,YI,XF,YF
24 800 FORMAT(' AI,YI, XF,YF',4G15.8)
25 HHI=(XF-XI)/FLOAT(MIMP-1)
26 HHJ=(YF-YI)/FLOAT(MIMP-1)
27 WRITE(6,842) MIMP,MIMP
28 842 FORMAT(' MIMP,MIMP ',2(14,' '))
29 WRITE(6,801)
30 801 FORMAT(' MNH NR XA YA UE UA ER ')
31 DO 100 NN=1,MIMP
32 WRITE(6,845)
33 845 FORMAT(' ')
34 DO 100 NN=1,MIMP
35 XIMP=XI+HHI*(NN-1)
36 YIMP=YI+HHJ*(NN-1)
37 DO 11 MN=1,NI
38 XX=PX(3,MN)
39 DO 110 MN=1,NJ
40 YY=PY(3,MN)
41 IF (YIMP-GT.YY) GO TO 110
42 MN=MM
43 GO TO 110
44 110 CONTINUE
45 MNH=NJ
46 112 CONTINUE
47 IF (XIMP-GT.XX) GO TO 210
48 NR=MM
49 GO TO 220
50 210 CONTINUE
51 11 CONTINUE
52 NR=NI
53 220 CONTINUE
54 RMP=(PY(1,MNH) - PY(3,MNH)) / (PX(3,NN) - PX(1,NR))
55 BB=- (RMP*PX(3,NR) - PY(1,MNH))
56 YYY=RMP*XIMP + BB
57 LI=1
58 IF (YIMP-LI,YYY) LI=0
59 DO 3 IXZ=1,6

```

```

59         IF (L1.EQ.1) GO TO 21
60         IT1=Y1(IXZ)
61         IT2=JY(IXZ)
62         GO TO 22
63         IT1=I*H(LIAZ)
64         IT2=J*H(LIAZ)
65         21 X(IXZ)=FX(LI1,NK)
66         Y(IXZ)=FY(LI2,NNN)
67         3 CONTINUE
68         XT1=X(1)+0.4D*PI
69         YTP=Y(2)+0.4D*PI
70         IF ((Y(4) GT X(1)) .AND. (X(4) .LT. XT2)) GO TO 1
71         Y(4)=RMP*X(4) + BB
72         GO TO 2
73         1 X(4)=(Y(4)-BB)/RMP
74         2 CONTINUE
75         IN1=(NR-1)*2+ICC(1)
76         JN2=IN1+1
77         IN3=IN2+1
78         JN1=(NNN-1)*2+ICC(3)
79         JN2=JN1+1
80         JN3=JN2+1
81         UX(1)=0.
82         UX(2)=0.
83         UX(3)=0.
84         UX(5)=0.
85         UX(6)=0.
86         IF ((IN1.GE.1) .AND. (JN3.LE.NY)) UX(3)=U(IN1,JN3)
87         IF ((IN3.LE.NA) .AND. (JN2.GE.1)) UX(5)=U(IN3,JN1)
88         UX(4)=U(IN1,JN2)
89         IF (L1.EQ.1) GO TO 30
90         IF ((IN1.GE.1) .AND. (JN1.GE.1)) UX(1)=U(IN1,JN1)
91         IF (IN1.GE.1) UX(5)=U(IN1,JN2)
92         IF (JN1.GE.1) UX(6)=U(IN2,JN1)
93         GO TO 40
94         30 IF ((IN3.LE.NA) .AND. (JN3.LE.NY)) UX(1)=U(IN3,JN3)
95         IF (JN3.LE.NY) UX(5)=U(IN3,JN3)
96         IF (IN3.LE.NX) UX(6)=U(IN3,JN2)
97         40 CONTINUE
98         CALL FVAL(A,Y,UX,TIME,TO,XIMP,YIMP,ISP,EPS,T1,T2)
99         CALL CALU(A,Y,UX,T1,T2,TIME,TO,XA,YA,UA,ISP)
100        UE=FEY(XA,YA,TIME,TO,IPR)
101        ER=UE-UA
102        ABSER=ABS(ER)
103        IF (ABSER.LE.E) GO TO 99
104        XMAX=XA
105        YMAX=YA
106        E=ABSER
107        99 F2=E2+(ER**2)
108        WRITE (6,42) NNN,NR,XA,YA,UE,UA,ER
109        42 FORMAT(2(I2," "),5G15.8)
110        100 CONTINUE
111        E2=SQRT(E2/FLUAI*(NIMP*NIMP))
112        C* WRITE(6,10) E,XMAX,YMAX,E2
113        C* WRITE(7,10) E,XMAX,YMAX,E2
114        C* WRITE(6,10) E,XMAX,YMAX,E2
115        10 FORMAT(" ER2 =74G15.8)
116        C*

```

(10/21/85) (ISAX)PROGRAMA/ED ON IIMAS.

PAGINA: 7-3

117
118
119
120
121
122
123

RETURN
END

```

1  C* SUBROUTINE EVAL=====
2  C*
3  C*   ESTA RUTINA OBTIENE T1 Y T2 A PARTIR DE XIMP,YIMP
4  C*   EN EL TRIANGULO DETERMINADO POR X1,Y1,U1=
5  C*   UTILIZO EL METODO DE NEWTON PARA ELLO.
6  C*
7  C* =====
8  SUBROUTINE EVAL(X1,Y1,U1,T1,T2,XIMP,YIMP,ISP,EPS,T1,T2)
9  DIMENSION XI(6),Y1(6),U1(6)
10  ISP=1
11  T1=1./3.
12  T2=1./3.
13  ZPKINERA ESTIMACION
14  IFIN=4
15  DO 2 J=1,IFIN
16  DX1=0.
17  DXT2=0.
18  DYT1=0.
19  DYT2=0.
20  DO 1 I=1,6
21  DXT1=DX1 + UIJ(I,T1,T2)*XI(I)
22  DXT2=DXT2 + UIJ(I,T1,T2)*XI(I)
23  DYT1=DYT1 + UIJ(I,T1,T2)*YI(I)
24  DYT2=DYT2 + UIJ(I,T1,T2)*YI(I)
25  1 CONTINUE
26  RJAC=DXT1*DYI2 - DXT2*DYI1
27  CALL CALC(X1,Y1,U1,T1,T2,TIME,T0,XX,YY,UX,ISP)
28  RES1=XIMP-XX
29  RES2=YIMP-YY
30  FAC1=(DYI2*RES1 - DXT2*RES2)/RJAC
31  FAC2=(-DYI1*RES1 + DXT1*RES2)/RJAC
32  T1=T1 + FAC1
33  T2=T2 + FAC2
34  IF(T1.GE.0.) GO TO 101
35  T2=T2/(1.-T1)
36  T3=T3/(1.-T1)
37  T1=0.
38  101 IF(T2.GE.0.) GO TO 102
39  T1=T1/(1.-T2)
40  T3=T3/(1.-T2)
41  T2=0.
42  102 IF(T3.GE.0.) GO TO 103
43  T2=T2/(1.-T3)
44  T1=T1/(1.-T3)
45  T3=0.
46  103 FAC1=ABS(FAC1)
47  FAC2=ABS(FAC2)
48  IF ((FAC1.LT. EPS).AND.(FAC2.LT. EPS)) GO TO 4
49  2 CONTINUE
50  4 CONTINUE
51  RETURN
52  END

```

ZCORRECCION PARA QUE
Z-T1 T2 Y T3 ESTEN DENTRO
ZDEC TRIANGULO

```

1 C* FINT=====
2 C*
3 C*
4 C* ESTA RUTINA INTEGRA MEDIANTE CUADRATURA GAUSSIANA DE
5 C* SIETE PUNTOS:
6 C*
7 C*
8 C*
9 C* =====
10 C* FUNCTION FINT(TIME, IO, (I1, XI, Y1, F, I, J, ID, IPR))
11 C*
12 C* DIMENSION UI(6), X1(6), Y1(6), DX(2,3)
13 C* DIMENSION X(7), I(7), U(6), W(7)
14 C* DATA X/0.3333333, 0.79742699,
15 C* 0.10128651, 0.10128651,
16 C* 0.05471587, 0.47014200,
17 C* 0.47014206, -0.47014200,
18 C* DATA Y/0.3333333, 0.10128651,
19 C* 0.79742699, 0.10128651,
20 C* 0.47014206, 0.05471587,
21 C* 0.47014206,
22 C* DATA W/0.22593918, 0.12593918,
23 C* 0.12593918, 0.12593918,
24 C* 0.13237415, 0.13239415,
25 C* 0.13237415,
26 C*
27 C* FINT=0.
28 C* DO 13 J=1, 7
29 C* T1=X(I)
30 C* T2=Y(I)
31 C* DX1=0.
32 C* DX2=0.
33 C* DYT1=0.
34 C* DYT2=0.
35 C* DO 1 I1=1, 6
36 C* DX1=DX(I1, 1) + UI(J, I1, T1, T2, 2)*X1(I1)
37 C* DX2=DX(I1, 2) + UI(J, I1, T1, T2, 3)*X1(I1)
38 C* DYT1=DY(I1, 1) + UI(J, I1, T1, T2, 2)*Y1(I1)
39 C* DYT2=DY(I1, 2) + UI(J, I1, T1, T2, 3)*Y1(I1)
40 C* 1 CONTINUE
41 C* DX(1, 1)=1.
42 C* DX(2, 1)=1.
43 C* DX(1, 2)=1.
44 C* DX(2, 2)=1.
45 C* DX(1, 3)=1.
46 C* DX(2, 3)=1.
47 C* RJAC=ABS( DX1*DYT2 - DX2*DYT1 )
48 C* DX(1, 2)=DYT2
49 C* DX(2, 2)=-DXT2
50 C* DX(1, 3)=-DXT1
51 C* DX(2, 3)=DXT1
52 C* ID1=ID+1
53 C* ID2=7-ID+1
54 C* FJ=I(N)*F(XI, Y1, UI, T1, T2, TIME, IO, IPR)*RJAC**((1-2*ID))
55 C* DO 2 ISP=ID1, ID2
56 C* DO 2 JSP=IUI, IUD
57 C* DO 2 IVAR=1, ID1
58 C* FINT=FINT*(UI(J, I1, T1, T2, ISP)*UI(J, T1, T2, JSP)
C* FINT=FINT + (FINT*(DX(IVAR, ISP)*DX(IVAR, JSP)))

```

```
59  
60      2   CONTINUE  
61      14  FINT=FINI  
62      13  CONTINUE  
63  C*  
64      RETURN  
65      END  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80
```

```

1 C* BASE=====NEW=====
2 C*
3 C*
4 C* ADUI TENGO LAS FUNCIONES BASE EN 2D
5 C*
6 C*
7 C* =====
8 FUNCTION HIJ(J, I, T2, ISP)
9 C*
10 T3=1.-T2-T1
11 JP=J+1
12 GO TO (10,110,210), ISP
13 C* 10 GO TO (20,30,32,34,36,38,40), JP
14 C*
15 20 HIJ=1.
16 RETURN
17 30 HIJ=T1*(2.*I1-1.)
18 RETURN
19 32 HIJ=T2*(2.*I2-1.)
20 RETURN
21 34 HIJ=T3*(2.*I3-1.)
22 RETURN
23 36 HIJ=4.*T2*T3
24 RETURN
25 38 HIJ=4.*T3*T1
26 RETURN
27 40 HIJ=4.*T1*T2
28 RETURN
29 110 GO TO (120,130,132,134,136,138,140), JP
30 C*
31 120 HIJ=0.
32 RETURN
33 130 HIJ=4.*T1-1.
34 RETURN
35 132 HIJ=0.
36 RETURN
37 134 HIJ=-4.*T3+1.
38 RETURN
39 136 HIJ=-4.*T2
40 RETURN
41 138 HIJ=4.*(T3-1.)
42 RETURN
43 140 HIJ=4.*T2
44 RETURN
45 210 GO TO (220,230,232,234,236,238,240), JP
46 C*
47 220 HIJ=0.
48 RETURN
49 230 HIJ=0.
50 RETURN
51 232 HIJ=4.*T2-1.
52 RETURN
53 234 HIJ=-4.*T3+1.
54 RETURN
55 236 HIJ=4.*(T3-1.)
56 RETURN
57 238 HIJ=-4.*T1
58 RETURN

```



```
59 240 UIJ=4.*T1
60      RETURN
61      END
62
63
```

```

1  C* CALC=====
2  C*
3  C*
4  C*   AQUI, DADAS LAS FUNCIONES BASE Y LOS SEIS PUNTOS EN EL
5  C*   TRIANGULO PERTINENTE, CALCULO XA Y YA.
6  C*
7  C*
8  C*   =====
9  C*   SUBROUTINE CALC(XI, YI, UI, I1, T2, TIME, TU, XA, YA, UA, ISP)
10  C*   DIMENSION XI(6), YI(6), UI(6)
11  C*   XA=0.
12  C*   YA=0.
13  C*   UA=0.
14  C*   DO 11 IYX=1,6
15  C*     XA=XA + UI(IYX, I1, T2, ISP)*XI(IYX)
16  C*     YA=YA + UI(IYX, I1, T2, ISP)*YI(IYX)
17  C*     UA=UA + UI(IYX, I1, T2, ISP)*UI(IYX)
18  C* 11 CONTINUE
19  C*   RETURN
20  C*   END

```

```

1 C* CHECK=====
2 C**
3 C**
4 C** COMPARA U CON UNEW ... ES DECIR LA ESTIMACION ANTERIOR
5 C** CON LA NUEVA, Y SI LA MAYOR DIFERENCIA ES MENOR QUE
6 C** EPSI, EPSR O SI YA DICE EL NUMERO MAXIMO DE ITERACIONES
7 C** DESEADO, DETIENE EL PROCESO;
8 C**
9 C**
10 C* ==
11 SUBROUTINE CHECK(U,UNEW,EPSI,EPSR,I,FLAG,NP,NX,NI,LIMIT,ITER)
12 DIMENSION U(NP),UNEW(NP)
13 IFLAG=0
14 SMAX=0.
15 RMAX=0.
16 DO 10 N=1,NP
17 TEMP=ABS(U(N) - UNEW(N))
18 IF (TEMP .GT. RMAX) RMAX=TEMP
19 TSMP=ABS(UNEW(N))
20 IF (TSMP .GT. SMAX) SMAX = TSMP
21 10 CONTINUE
22 EPS=EPSI+EPSR*SMAX
23 IF (RMAX .LT. EPS) GO10 20
24 IF (ITER .GE. LIMIT) GOTO 29
25 WRITE(6,112) RMAX,SMAX,LIMIT
26 112 FORMAT(" RMAX,SMAX,LIMIT",2015.8,I4)
27 WRITE(6,113) EPSI,EPSR,EPS,ITER
28 113 FORMAT(" EPSI,EPSR,EPS,ITER",5G15.8,I4)
29 GOTO 30
30 CONTINUE
31 IFLAG=1
32 CONTINUE
33 DO 15 J=1,NP
34 U(J)=UNEW(J)
35 15 CONTINUE
36 RETURN
37 END
38 C* ==
39 SUBROUTINE CHECK(U,UNEW,EPSI,EPSR,I,FLAG,NP,NX,NI,LIMIT,ITER)
40 DIMENSION U(NP),UNEW(NP)
41 IFLAG=0
42 SMAX=0.
43 RMAX=0.
44 DO 10 N=1,NP
45 TEMP=ABS(U(N) - UNEW(N))
46 IF (TEMP .GT. RMAX) RMAX=TEMP
47 TSMP=ABS(UNEW(N))
48 IF (TSMP .GT. SMAX) SMAX = TSMP
49 WRITE(6,521) N,U(N),UNEW(N),TEMP
50 10 CONTINUE
51 321 FORMAT(" N,U,UNEW,TEMP",I4,5(" ",5G15.8))
52 EPS=EPSI+EPSR*SMAX
53 WRITE(6,112) RMAX,SMAX,LIMIT
54 112 FORMAT(" RMAX,SMAX,LIMIT",2015.8,I4)
55 WRITE(6,113) EPSI,EPSR,EPS,ITER
56 113 FORMAT(" EPSI,EPSR,EPS,ITER",5G15.8,I4)
57 IF (RMAX .LT. EPS) GO10 20
58 IF (ITER .GE. LIMIT) GOTO 29

```

```
59      GOTO 30
60      CONTINUE
61      29 IFLAG=1
62      20 CONTINUE
63      30 DO 15 J=1,NP
64          U(J)=UNEM(J)
65      15 CONTINUE
66      RETURN
67      END
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
```

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
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53

```

C* FACTUR=====NEW=====
C*
C*
C*   ESTA RUTINA FACTORIZA MATRICES SINGULARES BANDAAS.
C*
C*
C* =====
C* SUBROUTINE FACTUR(LAP,AF,NP,M,K)
C* DIMENSION AF(NP,M),AF(NP,M)
C*   IAP=2
C*   DO 4 K=1, NP
C*     DO 2 J=1,M
C*       AF(K,J) = AF(K,J)
C*     CONTINUE
C*   4 CONTINUE
C*   DO 90 K=1, NP
C*     AKM=AF(K,M)
C*     IF (AKM .EQ. 0.0) GO TO 49
C*     IF (NP .LE. 1) GO TO 90
C*     L = K + M - 1
C*     IF (L .GE. NP) L = NP - 1
C*     DO 9 J = K,L
C*       DO 8 I = K,J
C*         IF (J .LE. I) GO TO 8
C*         AF(J+1,I-J+M) = AF(J+1,I-J+M) - AF(I+1,K+M-I-1)*
C*           AF(J+1,K+M-J-1)
C*       CONTINUE
C*       AA = AF(J+1,K+M-J-1)/AKM
C*       AF(J+1,M) = AF(J+1,M) - AA*AF(J+1,K+M-J-1)
C*       AF(J+1,K+M-J-1) = AA
C*   9 CONTINUE
C* 90 CONTINUE
C*   K = NP+1
C* 49 RETURN
C* END

```

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

```

1 C* SOL=====
2 C**
3 C**
4 C**
5 C**
6 C**
7 C**
8 C**
9 C**
10 SUBROUTINE SOL(AF,B,U,N,M)
11 DIMENSION AF(N,M),B(N),U(N),V(5000)
12 IW2=2
13 DO 11 K = 1,N
14 V(K) = B(K)
15 DO 3 K = 1,N
16 IF (K .LE. 1) GO TO 3
17 Z = V(K)
18 L = IW2
19 IF (K .GT. M) L = K - M + IW2
20 DO 1 J = L,K
21 Z = Z - AF(N,J-1-K+M)*V(J-1)
22 V(K) = Z
23 CONTINUE
24 DO 4 K = 1,N
25 U(K) = V(K)/AF(N,M)
26 DO 7 K = 1,N
27 J = N + 1 - K
28 IF (J .LE. 1) GO TO 6
29 Z = U(J)
30 L = IW2
31 IF (J .GT. M) L = J - M + IW2
32 DO 5 I = L,J
33 U(I-1) = U(I-1) - AF(J,I0)*Z
34 CONTINUE
35 CONTINUE
36 CONTINUE
37 RETURN
38 END
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```

```

1 C* INICIA=====
2 C*
3 C*
4 C*
5 C*   AQUI CALCULO EL VALOR INICIAL DE MI SOLUCION BASANDOME
6 C*   EN LA SOLUCION EXACTA.
7 C*
8 C*
9 C*   =====
10 C*   SUBROUTINE INICIA(U, PX, PY, TIME, TO, LEE, MA, NI, NJ, ILO, ICC,
11 C*     * NP, NX, NY, IPK, M, IT, JT, ITN, JTN)
12 C*
13 C*   EXTERNAL UNU
14 C*   DIMENSION U(LA, NY), X(6), Y(6), XN(6), YN(6), PX(3, NI), PY(3, NJ)
15 C*   REAL MA(NP, M), MP
16 C*   DIMENSION INO(3), JNO(3)
17 C*   DIMENSION ILO(4), OS(6), UB(6), IT(6), JT(6), ITN(6), JTN(6), ICC(4)
18 C*
19 C*   WRITE(6, 231) NX, NI, NP
20 C*   231 FORMAT("  NX:NI, NP", 3(I4, " "))
21 C*   DO 100 N=1, NI
22 C*     WRITE(6, *) N
23 C*     DO 100 NN=1, NJ
24 C*       RMP=( PY(1, NN) - PY(3, NN) ) / ( PX(3, N) - PX(1, N) )
25 C*       BB=( RMP*PA(3, N) - PY(1, NN) )
26 C*       DO 3 IXZ=1, 6
27 C*         IT1=1(IXZ)
28 C*         IT2=JT(IXZ)
29 C*         IT3=ITN(IXZ)
30 C*         IT4=JTN(IXZ)
31 C*         X(IXZ)=PX(1, N)
32 C*         Y(IXZ)=PY(1, NN)
33 C*         XN(IXZ)=PA(IT3, N)
34 C*         YN(IXZ)=PT(IT4, NN)
35 C*
36 C*   3 CONTINUE
37 C*   XT1=X(1)+0.45*PI
38 C*   XT2=X(2)-0.45*PI
39 C*   IF ((X(4).GT.X11).AND.(X(4).LT.XT2)) GO TO 1
40 C*   Y(4)=RMP*X(4) + BB
41 C*   GO TO 4
42 C*   1 CONTINUE
43 C*   X(4)=(Y(4)-BB)/RMP
44 C*   2 XN(4)=X(4)
45 C*   YN(4)=Y(4)
46 C*   IN1=(N-1)*2+ILO(1)
47 C*   IN2=IN1+1
48 C*   IN3=IN2+1
49 C*   JN1=(NN-1)*2+ILO(3)
50 C*   JN2=JN1+1
51 C*   JN3=JN2+1
52 C*   IND(1)=IN1
53 C*   IND(2)=IN2
54 C*   IND(3)=IN3
55 C*   JND(1)=JN1
56 C*   JND(2)=JN2
57 C*   JND(3)=JN3
58 C*   IF (NH.GE.1) GO TO 1U
59 C*   IF (NH.LE.3) GO TO 1O
60 C*   U(IN1, JN1)=FEX(X(1), Y(1), TIME, TO, IPR)
61 C*   U(IN2, JN1)=FEX(X(6), Y(6), TIME, TO, IPR)

```

```

1  C* INICIA=====
2  C**
3  C**
4  C**
5  C*   AQUI CALCULO EL VALOR INICIAL DE MI SOLUCION BASANDOME
6  C**   EN LA SOLUCION EXACTA.
7  C**
8  C*   =====
9  C*   SUBROUTINE INICIA(U,PX,PY,TIME,TO,LEE,MA,NI,NJ,ICO,ICC,
10  C*   *   NP,NX,NY,IPK,M,IT,JT,ITN,JTN)
11  C*   EXTERNAL UNO
12  C*   DIMENSION ULMA,NY),X(6),Y(6),XN(6),YN(6),PX(3,NI),PY(3,NJ)
13  C*   REAL MA(NP,M),RMP
14  C*   DIMENSION IND(3),JND(3)
15  C*   DIMENSION ILO(4),OS(6),UB(6),IT(6),JT(6),ITN(6),JTN(6),ICC(4)
16  C*
17  C*   WRITE(6,231) NX,NI,NP
18  231  FORMAT(" NX,NI,NP",3(I4," "),)
19  C*   DO 100 NN=1,NI
20  C*   WRITE(6,*) NN
21  C*   DO 100 NN=1,NJ
22  C*   RMP=( PY(1,NN) - PY(3,NN) )/( PX(3,NI) - PX(1,NI) )
23  C*   BB=(RMP*PX(5,NI) - PY(1,NN))
24  C*   DO 3 IXZ=1,6
25  C*   IT1=1(IXZ)
26  C*   IT2=JT(IXZ)
27  C*   IT3=ITN(IXZ)
28  C*   IT4=JTN(IXZ)
29  C*   X(IXZ)=PX(IT1,NI)
30  C*   Y(IXZ)=PY(IT2,NI)
31  C*   XN(IXZ)=PX(IT3,NI)
32  C*   YN(IXZ)=PY(IT4,NI)
33  C*   3  CONTINUE
34  C*   XT1=X(1)+0.45*NI
35  C*   XT2=X(2)+0.45*NI
36  C*   IF ((X(4) GT X11).AND.(X(4).LT.XT2)) GO TO 1
37  C*   Y(4)=RMP*X(4) + BB
38  C*   GO TO 4
39  C*   1  CONTINUE
40  C*   X(4)=(Y(4)-BB)/RMP
41  C*   2  XN(4)=X(4)
42  C*   YN(4)=Y(4)
43  C*   IN1=(N-1)*2+ICO(1)
44  C*   IN2=IN1+1
45  C*   IN3=IN2+1
46  C*   JN1=(IN1-1)*2+ICO(3)
47  C*   JN2=JN1+1
48  C*   JN3=JN2+1
49  C*   IND(1)=IN1
50  C*   IND(2)=IN2
51  C*   IND(3)=IN3
52  C*   JND(1)=JN1
53  C*   JND(2)=JN2
54  C*   JND(3)=JN3
55  C*   IF (NN.GT.1) GO TO 10
56  C*   IF (JN1.LE.0) GO TO 10
57  C*   IF ((N.LE.1).AND.(IN1.GT.0)) U(IN1,JN1)=FEX(X(1),Y(1),TIME,TO,IPR)
58  C*   U(IN2,JN1)=FEX(X(6),Y(6),TIME,TO,IPR)

```



```

59      IF (IN3.LE.NX) U(IN3,JN1)=FEX(X(2),Y(2),TIME,TO,IPR)
60      IF (N.GT.1) GO TO 20
61      IF (IM1.GE.0) GO TO 20
62      U(IM1,JM1)=FEX(X(5),Y(5),TIME,TO,IPR)
63      IF (JN3.LE.NY) U(IM1,JN3)=FEX(X(3),Y(3),TIME,TO,IPR)
64      U(IM2,JM2)=FEX(X(4),Y(4),TIME,TO,IPR)
65      IF (IN3.LE.NX) U(IN3,JN2)=FEX(XN(6),YN(6),TIME,TO,IPP)
66      IF (JN3.GE.NY) GO TO 40
67      U(IM2,JN3)=FEX(XN(5),YN(5),TIME,TO,IPR)
68      IF (IN3.LE.NX) U(IN3,JN3)=FEX(XN(1),YN(1),TIME,TO,IPP)
69      141 FORMAT(3(" X"IG15-8))
70      142 FORMAT(3(" Y"IG15-8))
71      40 DO 30 IJ=1,6
72          II=IT(IJ)
73          IM1=IND(IJ)
74          IF ((IM1.LI.1).OR.(IM1.GI.NX)) GO TO 30
75          JI=JT(IJ)
76          JM1=JND(JI)
77          IF ((JM1.LI.1).OR.(JM1.GI.NY)) GO TO 30
78          I11=IM1 + (JM1-1)*NX
79          DO 31 KL=1,IJ
80              K1=IT(KL)
81              KM1=IND(K1)
82              IF ((KM1.LI.1).OR.(KM1.GI.NX)) GO TO 31
83              L1=JT(KL)
84              LM1=JND(L1)
85              IF ((LM1.LI.1).OR.(LM1.GI.NY)) GO TO 31
86              JJ1=KM1 + (LM1-1)*NX
87              I1=MAX0(I11,JJ1)
88              JJT=ABS(I11-JJ1)
89              JJ=M-JJT
90              RR1=FN1(TIME,TO,US,X,Y,UND,IJ,KL,0,IPR)
91              RA(I1,JJ)=RA(I1,JJ) + RR1
92      31 CONTINUE
93      30 CONTINUE
94      DO 32 IJ=1,6
95          I2=ITH(IJ)
96          IM2=IND(I2)
97          IF ((IM2.LI.1).OR.(IM2.GI.NX)) GO TO 32
98          J2=JTH(IJ)
99          JM2=JND(J2)
100         IF ((JM2.LI.1).OR.(JM2.GI.NY)) GO TO 32
101         I12=IM2 + (JM2-1)*NX
102         DO 33 KL=1,IJ
103             K2=ITH(KL)
104             KM2=IND(K2)
105             IF ((KM2.LI.1).OR.(KM2.GI.NX)) GO TO 33
106             L2=JTH(KL)
107             LM2=JND(L2)
108             IF ((LM2.LI.1).OR.(LM2.GI.NY)) GO TO 33
109             JJ2=KM2 + (LM2-1)*NX
110             I1=MAX0(I12,JJ2)
111             JJT=ABS(I12-JJ2)
112             JJ=M-JJT
113             RR2=FN1(TIME,TO,UB,XN,YN,UND,IJ,KL,0,IPR)
114             RA(I1,JJ)=RA(I1,JJ) + RR2
115      33 CONTINUE
116      32 CONTINUE

```

```

117 100 CONTINUE
118 C* =====
119 C*
120 RETURN
121 END
122 C* RE1=====
123 C*
124 C*
125 C* AQUI CALCULO UN VALOR DE
126 C* LA SOLUCION EXACTA.
127 C*
128 C*
129 C* =====
130 SUBROUTINE RE1(U, PX, PT, TIME, TO, LEE, MA, NI, NJ, ICG, ICC,
131 * NP, NX, NY, YPR, M, IT, JT, ITR, JTN)
132
133 EXTERNAL UNU
134 DIMENSION U(NX, NY), X(6), Y(6), XN(6), YN(6), PX(3, NI), PY(3, NJ)
135 REAL MA(NP, M); MP
136 DIMENSION IND(3), JND(3)
137 DIMENSION ICG(4), US(6), UB(6), IT(6), JT(6), ITN(6), JTN(6), ICC(4)
138
139 C*
140 WRITE(6, 231) NX, NI, NP
141 231 FORMAT(" NX:NI, NP", 5(I4, " "))
142 DO 100 N=1, NI
143 C*
144 WRITE(6, *) N
145 DO 100 NN=1, NJ
146 RHP=( PY(1, NN) - PY(5, NN) ) / ( PX(3, NI) - PX(1, NI) )
147 BB=- (RHP*PA(5, N) - PY(1, NN))
148 DO 3 IXZ=1, 6
149 IT1=IT(IXZ)
150 IT2=JT(IXZ)
151 IT3=YN(IXZ)
152 IT4=JTN(IXZ)
153 X(IXZ)=PX(IT1, NI)
154 Y(IXZ)=PY(IT2, NN)
155 XN(IXZ)=PX(IT3, NI)
156 YN(IXZ)=PY(IT4, NN)
157
158 3 CONTINUE
159 XT1=X(1)+0.35*MI
160 XT2=X(2)+0.35*MI
161 IF ((X(4).GT.XT1).AND.(X(4).LT.XT2)) GO TO 1
162 Y(4)=RHP*X(4) + BB
163 GO TO
164 1 X(4)=(Y(4)-BB)/RHP
165 XN(4)=X(4)
166 YN(4)=Y(4)
167 IN1=(N-1)+ICG(1)
168 IN2=IN1+1
169 IN3=IN2+1
170 JN1=(NN-1)*2+ICG(3)
171 JN2=JN1+1
172 JN3=JN2+1
173 IND(1)=IN1
174 IND(2)=IN2
175 IND(3)=IN3
176 JND(1)=JN1
177 JND(2)=JN2
178 JND(3)=JN3

```

```

175     IF (NM.GT.1) GO TO 10
176     IF (JM1.LE.0) GO TO 10
177     IF ((N.LE.1).AND.(IM1.GT.0)) U(IM1,JM1)=FEX(X(1),Y(1),TIME,TO,IPR)
178     U(IM2,JM1)=FEX(X(6),Y(6),TIME,TO,IPR)
179     IF (N3.LE.NX) U(IM3,JM1)=FEX(X(2),Y(2),TIME,TO,IPR)
180     10  IF (N.GT.1) GO TO 20
181     IF (IM1.LE.0) GO TO 20
182     U(IM1,IM2)=FEX(X(5),Y(5),TIME,TO,IPR)
183     IF (JM3.LE.NY) U(IM1,JM3)=FEX(X(3),Y(3),TIME,TO,IPR)
184     20  U(IM2,IM2)=FEX(X(4),Y(4),TIME,TO,IPR)
185     IF (N3.LE.NX) U(IM3,JM2)=FEX(X(6),Y(6),TIME,TO,IPR)
186     IF (JM3.GT.NY) GO TO 40
187     U(IM2,JM3)=FEX(X(5),Y(5),TIME,TO,IPR)
188     IF (N3.LE.NX) U(IM3,JM3)=FEX(X(1),Y(1),TIME,TO,IPR)
189     141 FORMAT(3(" X",G15.8))
190     142 FORMAT(3(" Y",G15.8))
191     40  DO 30 IJ=1,6
192         I1=IT(IJ)
193         IM1=IMD(I1)
194         IF ((IM1.LT.1).OR.(IM1.GT.NX)) GO TO 30
195         J1=JT(IJ)
196         JM1=JMD(J1)
197         IF ((JM1.LT.1).OR.(JM1.GT.NY)) GO TO 30
198         II1=I1 + (JM1-1)*NX
199         DO 31 KL=1,IJ
200             K1=IT(KL)
201             KM1=INDX(K1)
202             IF ((KM1.LT.1).OR.(KM1.GT.NX)) GO TO 31
203             L1=JT(KL)
204             LM1=JMD(L1)
205             IF ((LM1.LT.1).OR.(LM1.GT.NY)) GO TO 31
206             JJ1=KM1 + (LM1-1)*NX
207             II=MAX0(II1,JJ1)
208             JJT=IABD(II1-JJ1)
209             JJ=M-JJT
210             RR1=FINF(TIME,TO,US,X,Y,UNO,IJ,KL,0,IPR)
211             MA(II,JJ)=MA(II,JJ) + RR1
212     31  CONTINUE
213     30  CONTINUE
214         DO 32 IJ=1,6
215             I2=ITN(IJ)
216             IM2=IMD(I2)
217             IF ((IM2.LT.1).OR.(IM2.GT.NX)) GO TO 32
218             J2=JTN(IJ)
219             JM2=JMD(J2)
220             IF ((JM2.LT.1).OR.(JM2.GT.NY)) GO TO 32
221             II2=I2 + (JM2-1)*NX
222             DO 33 KL=1,IJ
223                 K2=ITN(KL)
224                 KM2=INDX(K2)
225                 IF ((KM2.LT.1).OR.(KM2.GT.NX)) GO TO 33
226                 L2=JTN(KL)
227                 LM2=JMD(L2)
228                 IF ((LM2.LT.1).OR.(LM2.GT.NY)) GO TO 33
229                 JJ2=KM2 + (LM2-1)*NX
230                 II=MAX0(II2,JJ2)
231                 JJT=IABD(II2-JJ2)
232                 JJ=M-JJT

```

(10/21/85) (ISAX)PROGRAMA/ED ON IMAS.

PAGINA: 1/5

```
233          RR2=FINI(IME,TO,UB,XN,YN,UNO,IJ,KL,O,IPR)
234          VA(II,JJ)=MA(II,JJ)+RR2
235          33 CONTINUE
236          32 CONTINUE
237          100 CONTINUE
238 C* -----
239 C*
240          RETURN
241          END
```

(10/21/85) (ISAX)PROGRAMA/CD ON IIMAS.

PAGINA: 1/5

```
233          RR2=FINI(IME,TO,UB,XI,YN,UND,IJ,KL,O,IPR)
234          VA(II,JJ)=RA(II,JJ)+RR2
235          33 CONTINUE
236          32 CONTINUE
237          100 CONTINUE
238 C* =====
239 C*
240          RETURN
241          END
```

1
2
3
4
5
6
7

```
FUNCTION IXX(NX,LM)
  NXPI=NX+1
  IF (LM.GT.NXPI) IAX=NXPI+NX-LM
  IF ((LM.GE.3).AND.(LM.LE.NXPI)) IXX=5+NX-LM
  IF (LM.LT.3) :XX=10-LM
  RETURN
END
```

010/21/85) (ISAX)PROGRAMA/CD ON IIMAS.

PAGINA: 17-1

```
1 FUNCTION IYI(NX,LM)
2 NXPI=NX+1
3 IF (LM.GT.7) IYY=13+LM
4 IF ((LM.GT.5).AND.(LM.LE.8)) IYY=6+LM
5 IF (LM.LE.3) IYY=5M
6 RETURN
7 END
8
9
10
11
12
13
14
15
16
17
18
```

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

C**
C**

```

FUNCTION UNU=====
=====
FUNCTION UNU(XI,YI,UI,TI,T2,TIME,TO,ICR)
DIMENSION XI(6),YI(6),UI(6)
UNU=1
RETURN
END

```