

**METODO MULTIGRILLA Y REFINAMIENTO ADAPTIVO
EN PROBLEMAS ELIPTICOS DISCRETIZADOS POR EL
METODO DE LOS ELEMENTOS FINITOS**

Mario Storti*

Norberto Nigro**

Instituto de Desarrollo Tecnológico para
la Industria Química (INTEC)
UNL-CONICET
Santa Fe-Argentina

Resumen

Las técnicas de multigrilla y refinamiento adaptivo han demostrado ser una herramienta muy poderosa en la resolución de problemas de mecánica de fluidos. En mallas estructuradas la resolución por multigrilla del sistema de ecuaciones planteado insume un número de operaciones que es del orden del número de grados de libertad, independientemente de la dimensionalidad del espacio. En este trabajo se desarrollan técnicas originales para el almacenamiento de los datos de la malla en estructuras tipo árbol facilitando el acceso a la información tanto para la resolución por el método de multigrilla como para el refinamiento adaptivo. Se discute el uso de indicadores de error en la etapa de refinamiento adaptivo. Finalmente se presentan ejemplos numéricos.

Abstract

Multigrid and adaptive refinement techniques are powerful tools in the resolution of problems arising from fluid mechanics. In structured grids the number of numerical operations needed in the resolution via the multigrid method is of the order of the number of degrees of freedom, regardless the dimensionality of the problem. In this work, original techniques for data storage in tree-like structures are developed, simplifying access to the information. The use of error indicators in the adaptive refinement stage is discussed. Finally, numerical examples are presented.

* Becario de Formación Superior del CONICET

** Becario de Iniciación del CONICET

0. INTRODUCCION

La discretización por el MEF conduce a la resolución de un sistema algebraico de N ecuaciones. Dado que la precisión en la solución está fuertemente ligada a N , se dedica mucho esfuerzo al desarrollo de métodos computacionales más eficientes para resolver este sistema. Inicialmente se usaron métodos directos los cuales arriban a la solución en un número finito de operaciones. Como ejemplo, podemos citar la factorización por el método de Gauss o de Cholesky. Por otro lado están los métodos iterativos. Estos se caracterizan por generar una secuencia de soluciones aproximadas que convergen a la exacta. Podemos citar entre otros el método de gradientes conjugados con o sin *precondicionamiento*. La eficiencia de todos estos métodos puede compararse a partir del número de operaciones que requieren para su solución. Este valor es en todos los casos proporcional a N^α donde α depende de la dimensión del espacio y del método en sí (ver Tabla 1). El método de multigrilla (MMG) es óptimo en este sentido ya que requiere estrictamente $O(N)$ operaciones. Es especialmente apto cuando la malla es estructurada y el operador diferencial homogéneo, y en principio puede aplicarse a operadores no necesariamente autoadjuntos y definidos positivos. Debido a esto su uso es corriente en problemas de mecánica de fluidos, principalmente para la ecuación de Euler compresible [1] y flujo potencial transónico[3].

Tabla 1. Ordenes de convergencia α de los métodos iterativos

Método	2D	3D
Cholesky banda:		
factorización	2	2.33
resolución	1.5	1.67
Dissección anidada:		
factorización	1.5	2.
resolución	1.	1.33
Gradientes conjugados	1.5	1.33
Gradientes conjugados precondicionados	1.25	1.17
Multigrilla	1.	1.

En un contexto de elementos finitos, la programación del MMG se torna mucho más complicada debido a la mayor generalidad de las mallas utilizadas. Si bien el costo total de resolución sigue siendo óptimo ($O(N)$), la ventaja frente a los otros métodos se hace sensible a partir de un N relativamente elevado ($\approx 20\ 000$). Por otra parte las técnicas de refinamiento adaptivo (RA) han probado ser muy eficientes en un contexto de MEF, ya que éste está orientado a la modelización de geometrías complicadas.

1. NOTACION GENERAL

Seguiremos en lo posible la notación utilizada por Bank[4]. Por simplicidad nos restringiremos a 2D. Sea G^0 una malla de elementos finitos convencional, es decir

$$\begin{aligned}
 G^0 &= \cup_{i=1}^{N_e} \Omega_e; & \Omega_e &= \text{triángulos o cuadrángulos} \\
 & & v_j^e &= \text{vértices de } \Omega_e \quad j = 1, \dots, N_e \\
 & & \Omega_e \cap \Omega_{e'} &= \emptyset \text{ o un lado o un vértice}
 \end{aligned}
 \tag{1}$$

con N_e el número de nodos conectados al elemento e . G^0 es la malla ingresada por el usuario, definimos entonces como mallas admisibles aquellas que se obtienen de la original o de otra admisible subdividiendo un elemento triangular o cuadrangular en cuatro, uniendo el punto medio de los lados del elemento. Durante el proceso de refinamiento se genera una secuencia de mallas admisibles $G^0, G^1, \dots, G^R, \dots$. El criterio de refinamiento será descrito posteriormente, por el momento nos limitamos a suponer la existencia de tal secuencia. Una forma natural de describir este tipo de mallas es mediante una estructura arborescente. Esta tiene como nodos raíces los elementos de la malla original G^0 . El refinar uno de estos elementos se describe en el árbol como la aparición de cuatro nodos hijos (asociados con otros tantos elementos creados) hijos del nodo asociado con el elemento refinado. Para no confundir los nodos del árbol con los nodos de la malla del MEF llamaremos a los primeros "puntos". Hay una correspondencia unívoca entre la malla admisible y el árbol correspondiente. A cada punto del árbol le asociamos un nivel ℓ de la forma natural: los puntos raíces son de nivel $\ell = 0$ y los hijos de un punto de nivel ℓ son de nivel $\ell + 1$. Sea $\gamma = \gamma(G^R)$ el nivel del elemento de máximo nivel de G^R ; definimos una secuencia de mallas admisibles $G_0^R, G_1^R, \dots, G_\gamma^R$, de la siguiente forma:

$$i) G_\gamma^R = G^R \quad (2)$$

$$ii) \text{árbol de } G_{\ell-1}^R = \text{árbol de } G_\ell^R \text{ eliminando los puntos del nivel } \ell \quad (3)$$

Obviamente tenemos $G_0^R = G^0 \forall R$.

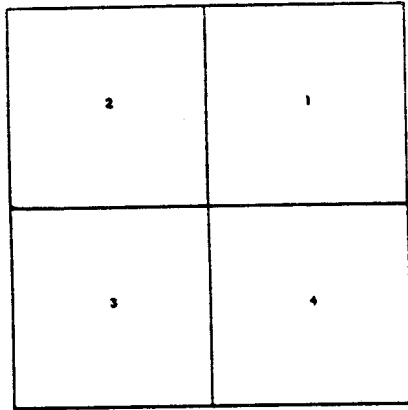
En la figura 1 podemos ver un ejemplo con las mallas G_ℓ^R para clarificar lo antedicho. En la figura 1.a vemos la malla original G^0 . La malla G^1 se obtiene refinando los elementos 3 y 4 (fig 1.b) y contiene 2 elementos del nivel 0 (elementos 1 y 2) y 8 elementos de nivel 1 (elementos 5 a 12). La malla G^2 se obtiene refinando los elementos 1, 6 y 12 de G^1 (fig 1.c) y contiene 1 elemento de nivel 0, 10 de nivel 1 y 8 de nivel 2. Por definición, tenemos $G_{\ell=2}^2 \equiv G^2$. La malla G_1^2 se obtiene eliminando los elementos de nivel 2 (fig 1.d), vemos que $G_1^2 \neq G^1$, esto ocurrirá siempre que al refinar una cierta malla G^R se refine algún elemento de nivel $\ell \leq \gamma(G^R)$. En el ejemplo se ha refinado el elemento 1 de G^1 que es de nivel 0 y no 1 ($= \gamma(G^1)$). En el ejemplo $\gamma(G^R) = R$. Esto puede no ser así si en algún paso de refinamiento no se refina ningún elemento de nivel máximo γ . En la figura 2 se observa el árbol correspondiente a $G_{\ell=2}^2$. Los árboles correspondientes a las diferentes G_ℓ^R pueden obtenerse como subgrafos de G_γ^R .

El método MMG-RA se puede resumir de la siguiente forma: dada una malla G^{R-1} el proceso de RA da como resultado una malla G^R , interpolando la solución obtenida en G^{R-1} obtenemos una inicialización para el MMG quien, después de un proceso iterativo, nos da finalmente la solución aproximada para este nivel de refinamiento. A continuación nos concentraremos en el MMG dejando el RA para más tarde. Nos ubicaremos entonces en una malla admisible G^R y a partir de ahora por simplicidad obviamos el supraíndice R . Para cada una de las G_ℓ el MEF da una ecuación:

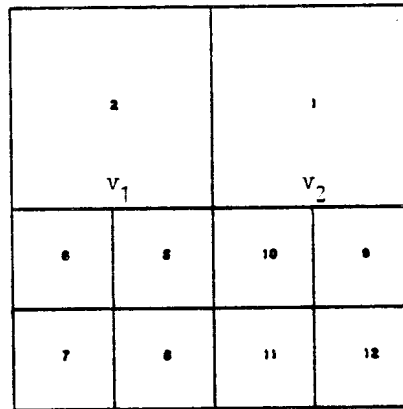
$$A_\ell \mathbf{u}_\ell = \mathbf{b}_\ell \quad (4)$$

Sea \mathcal{N}_ℓ el conjunto de los nodos de la malla G_ℓ . Estos pueden ser clasificados de la siguiente forma:

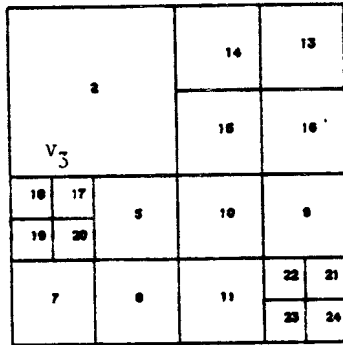
$$\mathcal{N}_\ell = \mathcal{N}_{F,\ell} \cup \mathcal{N}_{B,\ell} \cup \mathcal{N}_{R,\ell} \quad (5)$$



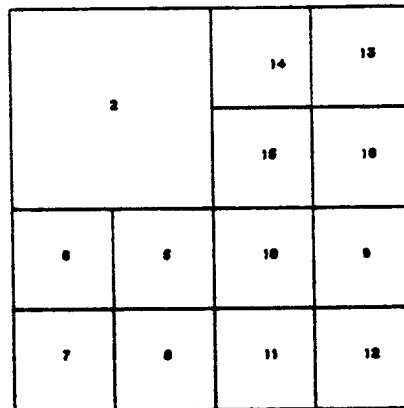
a) R=0 l=0
 R=1 l=0
 R=2 l=0



b) R=1 l=1



c) R=2 l=2

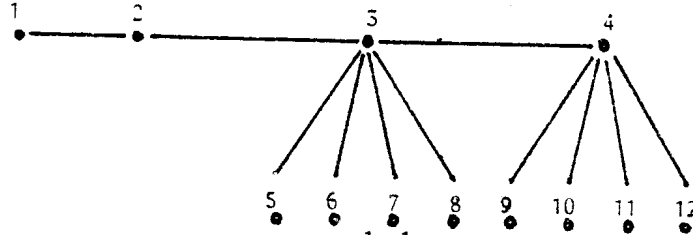


d) R=2 l=1

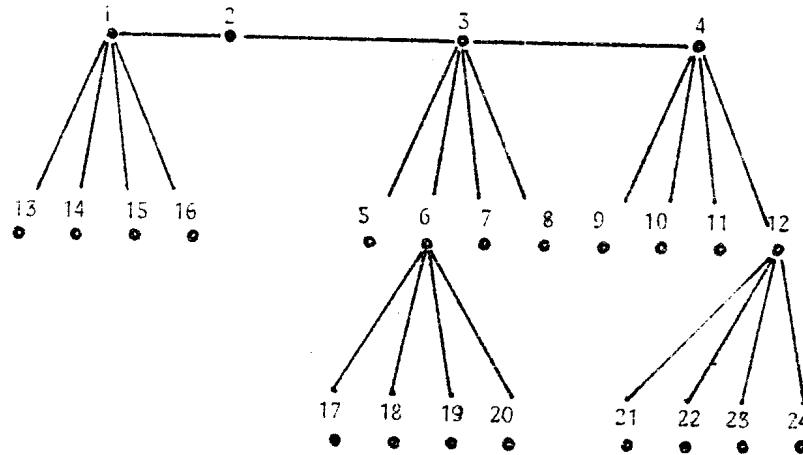
Figura 1. Secuencia de mallas G_1^R



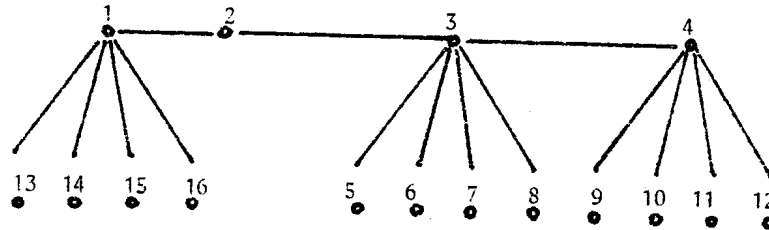
a) árbol de $G^0 = G_0^0 = G_0^1 = G_0^2$



b) árbol de $G^1 = G_1^1$



c) árbol de $G^2 = G_2^2$



d) árbol de G_1^2

Figura 2. Arbol de la secuencia de mallas

$$\mathcal{N}_{F,\ell}, \mathcal{N}_{B,\ell}, \mathcal{N}_{R,\ell} \text{ disjuntos} \quad (6)$$

Sea un nodo u_i :

$$\begin{aligned} u_i \in \mathcal{N}_{F,\ell} &: && \text{si está conectado sólo a elementos de nivel } \ell \\ u_i \in \mathcal{N}_{B,\ell} &: && \text{si está conectado a algún elemento de nivel } \ell \\ &&& \text{y a otro de nivel } \ell' < \ell \\ u_i \in \mathcal{N}_{R,\ell} &: && \text{si no está conectado a elementos de nivel } \ell \end{aligned}$$

Se verifica que:

$$\mathcal{N}_0 \subset \mathcal{N}_1 \subset \dots \subset \mathcal{N}_\gamma = \mathcal{N} \quad (7)$$

Además definimos:

$$\mathcal{N}_{T,\ell} = \mathcal{N} - \mathcal{N}_\ell \quad (8)$$

Dada una aproximación u_ℓ^j a u_ℓ podemos encontrar una mejor aproximación de la siguiente forma:

$$u_\ell^{j+1} = u_\ell^j - \omega r_\ell^j \quad (9)$$

donde r_ℓ^j es el residuo de la ecuación en G_ℓ :

$$r_\ell^j = A_\ell u_\ell^j - b_\ell \quad (10)$$

Este proceso puede ser descrito en forma compacta como:

$$u_\ell^{j+1} = \text{Relax}_\ell(u_\ell^j, b_\ell) \quad (11)$$

Puede demostrarse que la secuencia converge a u_ℓ para valores de ω que satisfagan:

$$0 < \omega < \frac{2}{(\lambda_{\max})_\ell} \quad (12)$$

Sin embargo, un algoritmo basado solamente en este tipo de método no sería eficiente. Esto se debe a que las componentes del error en el autovector s_j de A_ℓ es reducida en un factor $(1 - \omega \lambda_j)$ en cada paso. Ahora bien, los operadores usados corrientemente se caracterizan por un número de condición $\kappa = \lambda_{\max}/\lambda_{\min}$ elevados. Podemos ver entonces que para $\lambda_j = \lambda_{\min}$:

$$1 - \omega \lambda_{\min} \cong 1 - \frac{2}{\kappa} \quad (13)$$

es decir, la convergencia es muy pobre para las componentes correspondientes a los autovalores menores. Haciendo un análisis de Fourier se puede demostrar que los autovalores bajos están asociados generalmente a las componentes suaves del residuo. La idea central del MMG consiste en aplicar este tipo de método para reducir las componentes oscilatorias del residuo y una vez que éste es suficientemente regular obtener una corrección en la malla mas gruesa $G_{\ell-1}$. Para ello el residuo se transfiere a esta malla usando algún operador interpolante:

$$b_{\ell-1} = J_\ell^{\ell-1} r_\ell \quad (14)$$

En G_{l-1} planteamos una ecuación para la corrección:

$$A_{l-1} w_{l-1} = b_{l-1} \quad (15)$$

El algoritmo de dos grillas consiste en resolver exactamente esta ecuación y agregar la corrección interpolada:

$$\bar{u}_l^{j+1} = u_l^{j+1} + I_{l-1}^l w_{l-1} \quad (16)$$

Este proceso elimina efectivamente las componentes suaves obteniéndose un residuo fuertemente oscilatorio, apto para reiniciar el proceso de relajación. Este proceso es detallado a continuación:

Procedimiento 2-grillas:

entrada: $u_l = u_l^0, \epsilon$

2G1) $u_l = \text{Relax}_l(u_l, b_l)$

2G2) Si $\|r_l^{j+1}\| < \epsilon$ entonces PARE

2G3) Si la convergencia es lenta entonces:

comienzo

2G3.1) $b_{l-1} \leftarrow I_{l-1}^{l-1} r_l$

2G3.2) $w_{l-1} = A_{l-1}^{-1} b_{l-1}$ en G_{l-1}

2G3.3) $u \leftarrow I_{l-1}^l w_{l-1} + u_l$

fin

2G4) vaya a 2G1)

Este método no es muy utilizado ya que una extensión obvia consiste en resolver el paso de corrección 2G3.2 por el método de 2-grillas en G_{l-1}, G_{l-2} . Además la precisión en la resolución no tiene porqué ser muy elevada dado que se desea obtener una corrección y no una solución en sí. Un criterio seguido por Brandt y que ha dado buenos resultados es:

$$\|w_{l-1} - A_{l-1}^{-1} b_{l-1}\| < \epsilon_F \|b_{l-1}\| \quad (17)$$

De esta forma se obtiene el MMG en forma recursiva.

Haciendo un análisis de Fourier puede demostrarse que el ciclo 2G1-2G4 reduce la norma del residuo en un factor fijo ρ independientemente de h . El costo W_{l+1} de un tal ciclo se puede calcular recursivamente:

$$W_{l+1} = w_R \nu N_{l+1} + W_l \frac{\log \epsilon_F}{\log \rho} \quad (18)$$

donde:

$N_l = \# \mathcal{N}_l$

$w_R =$ número de operaciones por grado de libertad
y por relajación

$\log \epsilon_F / \log \rho =$ número de ciclos del algoritmo de 2-grillas
en $G_{l-1} - G_{l-2}$ necesarios para reducir el
residuo en un factor ϵ_F

Suponiendo una malla refinada en forma homogénea: $N_{\ell+1} = nN_{\ell}$ con n constante ($n = 4$ en nuestro caso de triángulos y cuadrados) y el valor asintótico para $W_{\ell+1}$ es:

$$W_{\ell+1} = w_R \nu N_{\ell+1} / \left(1 - \frac{1 \log \epsilon_F}{n \log \rho} \right) \quad (19)$$

A continuación mencionaremos ciertos detalles de la implementación del MMG en el contexto FEM:

- i) En el MDF la regularidad de la malla original y de cada una de las G_{ℓ} facilita la tarea del almacenamiento de las diferentes variables de trabajo: incógnitas, residuos, etc. . . En especial, durante el proceso de relajación debemos acceder a los valores de los nodos conectados por algún elemento al nodo en cuestión. Además esto debe hacerse en todos los niveles $0 \leq \ell \leq \gamma$.
- ii) El almacenamiento de los datos correspondientes a los elementos se hace a través de un árbol. Esto es óptimo ya que permite guardar la información de todas las mallas G_0, \dots, G_{γ} sin duplicar la información.
- iii) Al relajar a un cierto nivel G_{ℓ} sólo lo hacemos sobre los nodos de $\mathcal{N}_{F,\ell}$ (la parte "fina" de G_{ℓ}). Sólo los elementos de nivel ℓ dan contribución a estos nodos y, por lo tanto, sólo hace falta conocer los valores nodales en $\mathcal{N}_{F,\ell} \cup \mathcal{N}_{B,\ell}$.
- iv) Puede demostrarse la siguiente serie de inclusiones:

$$\begin{aligned} \mathcal{N}_{R,\ell} &\subset \mathcal{N}_{R,\ell+1} \\ \mathcal{N}_{B,\ell} &\subset (\mathcal{N}_{R,\ell+1} \cup \mathcal{N}_{B,\ell+1}) \\ \mathcal{N}_{F,\ell} &\subset (\mathcal{N}_{R,\ell+1} \cup \mathcal{N}_{B,\ell+1} \cup \mathcal{N}_{F,\ell+1}) \end{aligned} \quad (20)$$

Por lo tanto debemos tener para un dado v :

$$\begin{array}{llll} v \in \mathcal{N}_{T,\ell}; & 0 \leq \ell & \text{y} & \ell \leq k_F \\ v \in \mathcal{N}_{F,\ell}; & k_F \leq \ell & \text{y} & \ell \leq k_B \\ v \in \mathcal{N}_{B,\ell}; & k_B \leq \ell & \text{y} & \ell \leq k_R \\ v \in \mathcal{N}_{R,\ell}; & k_R \leq \ell & \text{y} & \ell \leq \gamma \end{array}$$

La detección de los nodos irregulares es muy importante ya que estos nodos no constituyen *grados de libertad* en la ecuación sino que el valor de la función en estos nodos está impuesto como el promedio de los valores de los nodos adyacentes. Por ejemplo los nodos v_1 y v_2 en la malla de la figura 1.b son irregulares.

- v) **Nodos irregulares:** algunos autores consideran que no es deseable la existencia de nodos irregulares en la malla alegando que la existencia de estos nodos aumenta el trabajo computacional y la cantidad de funciones de forma que tienen soporte en cada elemento. Nuestra opinión es que estos nodos no traen problemas si se mantiene el criterio de 1-irregularidad [4], es decir que un nodo no debe estar conectado a elementos que difieran en más de un nivel. Por ejemplo, la malla G^1 en la figura 1.b satisface el criterio de 1-irregularidad mientras que la malla G^2 de la figura 1.d no, debido al vértice v_2 que está conectado a los elementos 18 ($\ell = 2$) y 2 ($\ell = 0$).
- vi) La imposición del valor de los nodos irregulares al promedio de los nodos adyacentes es tenida en cuenta automáticamente por el MMG.

- vii) Al pasar a una malla gruesa el residuo debe ser calculado en toda la malla mas fina G_i para luego transferirlo a la G_{i-1} . En este caso el residuo de los nodos irregulares debe ser transferido por partes iguales a los nodos adyacentes (éstos no pueden ser irregulares si se satisface el criterio de 1-irregularidad)

3. TECNICAS DE REFINAMIENTO ADAPTIVO

Las técnicas de RA constan de dos puntos importantes a saber:

- RA-i) evaluación del error de la aproximación en la malla G^R
 RA-ii) elección de un criterio para decidir cuales elementos de G^R serán refinados al pasar a G^{R+1}

Un análisis del error por interpolación nos da la siguiente estimación:

$$|u_I(\mathbf{x}) - u(\mathbf{x})| \approx \left| \sum_{i,j=1}^n \frac{\partial^2 u}{\partial x_i \partial x_j} \Delta x_i \Delta x_j \right| \quad (21)$$

Donde u_I es le interpolada por elementos finitos de u . Tomamos entonces como indicador del error (IE):

$$E = \sum_{\epsilon} \Phi_{\epsilon} h_{\epsilon}^2 \Omega_{\epsilon} \approx \int_{\Omega} |u_I - u|^2 d\Omega \quad (22)$$

Donde:

$$\Phi_{\epsilon} = \max_{i,j,k} \left| \frac{\partial^2 u}{\partial x_i \partial x_j} (v_k) \right| \quad (23)$$

para $1 \leq i, j \leq n$; v_k nodo del elemento ϵ

Una vez obtenida la aproximación $u_h^R(\mathbf{x})$ por el MEF en G^R podemos obtener estimaciones para $\partial^2 u / \partial x_i \partial x_j$ a partir de fórmulas típicas como:

$$\frac{\partial^2 u}{\partial x_i \partial x_j} (v_k) = - \int_{\Omega} \frac{\partial N_k}{\partial x_i} \frac{\partial u_h^R}{\partial x_j} d\Omega \quad (24)$$

Esto completa el punto RA-i). Con respecto al punto RA-ii) seguiremos la línea desarrollada por Brandt [5]. Para ello remplazemos la suma sobre los elementos por una integral:

$$E = \int_{\Omega} \Phi(\mathbf{x}) h^2(\mathbf{x}) d\Omega \quad (25)$$

El trabajo computacional W es proporcional al número de grados de libertad, por lo tanto del orden del número de elementos:

$$W \approx w N_E = w \sum_{\epsilon} 1 \approx w \sum_{\epsilon} h^{-d} \Omega_{\epsilon} \approx w \int_{\Omega} h^{-d} d\Omega \quad (26)$$

La malla óptima será aquella que nos de un error menor a trabajo computacional constante. tratamos de hallar la función $h(\mathbf{x})$ tal que:

$$\delta(E[h(\mathbf{x})] + \lambda W[h(\mathbf{x})]) = 0 \quad (27)$$

$$\int_{\Omega} (\Phi(\mathbf{x}) 2h - \lambda dw h^{-d-1}) \delta h \, d\Omega = 0 \quad (28)$$

$$\Rightarrow \Phi(\mathbf{x}) h(\mathbf{x})^{2+d} = \frac{\lambda dw}{2} = \text{cte} \quad (29)$$

O, volviendo al problema discreto:

$$\eta_e = \Phi_e h_e^{d+2} = \text{cte} \quad (30)$$

Estas ideas un tanto "heurísticas" nos conducen a un criterio práctico de refinamiento una vez hallada la solución u_h^R :

- 1) Calcular las "eficiencias" η_e para cada elemento.
- 2) La malla G^{R+1} se obtendrá refinando aquellos elementos e tales que:

$$\eta_e \geq C_\eta \eta_{\max} = C_\eta \max_{1 \leq e \leq N_n} \{\eta_e\} \quad (31)$$

C_η es un parámetro a ingresar por el usuario (por ejemplo $C_\eta = 0.5$).

Mucho trabajo teórico se ha dedicado a la obtención de estimadores de error a posteriori (EEAP). Muchos investigadores, entre los cuales cabe destacar Bank [6] y Babuska [7] han desarrollado una serie de EEAP. En algunos casos se puede demostrar que estos EEAP se acercan (asintóticamente) al error. En este trabajo el RA se ha basado en un indicador de error (IE) más bien que en aquel tipo de EEAP. Este tipo de IE ha sido utilizado por Brandt en un contexto MDF y se basa parte en razonamientos heurísticos y parte en análisis del error de interpolación. El criterio para la elección del IE (o EEAP) fue:

- i) el IE (o EEAP) debe permitir decidir cuáles elementos serán refinados durante el proceso de RA.
- ii) el IE (o EEAP) debe servir como cota de detención del proceso de RA por parte del usuario.
- iii) simplicidad de programación y costo de ejecución.

Con respecto al punto i) podemos decir que, si bien los EEAP cuentan con un aval teórico muy importante como estimadores del error global, poco o nada se conoce de su habilidad para poder predecir cuáles son los elementos a refinar en el próximo paso.

4. EJEMPLOS NUMERICOS

Se trata de resolver la ecuación de Laplace en el cuadrado unitario con condiciones tipo Dirichlet en la frontera:

$$\begin{cases} \Delta u = 0; & \text{en } \Omega = [0, 1] \times [0, 1] \\ u = \bar{u}; & \text{en } \partial\Omega \end{cases}$$

Elegimos:

$$\begin{aligned} \bar{u} &= (x^2 + y^2)^{\alpha/2} \cos\left(\alpha \arctan\left(\frac{y}{x}\right)\right) = \Re\{f(z)\} \\ z &= x + iy \\ f(z) &= z^\alpha \end{aligned}$$

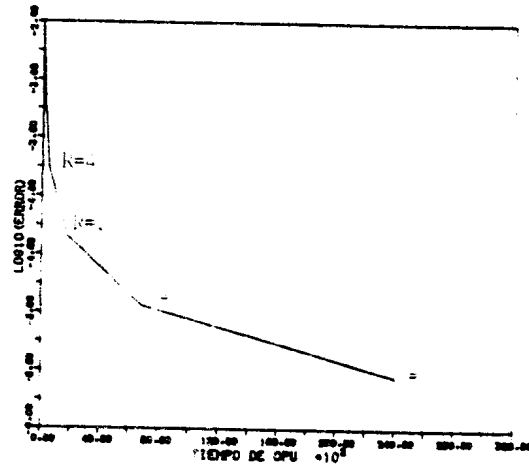
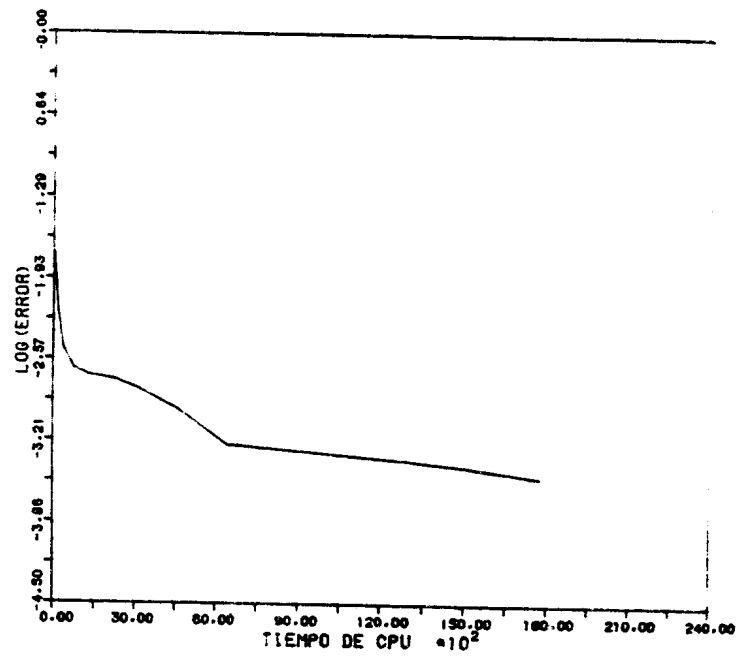
a) $\alpha=1/2$ (1^{er} ejemplo)b) $\alpha=0.5$ (2^{do} ejemplo)

Figura 5. Convergencia del NMG

f tiene (eventualmente, dependiendo de α) una singularidad en $x = y = 0$:

$$\begin{aligned} |\nabla u| &= \left| \frac{df}{dz} \right| = \alpha r^{\alpha-1} \\ |\nabla \nabla u| &= \frac{1}{2} \left(\frac{\partial^2 u}{\partial x^2} \right)^2 + \frac{1}{2} \left(\frac{\partial^2 u}{\partial y^2} \right)^2 + \left(\frac{\partial^2 u}{\partial x \partial y} \right)^2 = \\ &= \left| \frac{d^2 f}{dz^2} \right| = \alpha(\alpha-1) r^{\alpha-2} \end{aligned}$$

El primer ejemplo numérico está relacionado con la convergencia de la implementación del MMG. Se ha tomado $\alpha = 1.2$ y $C_\eta = 0$ de manera que en cada paso de refinamiento *todos los elementos son refinados*. La malla G^0 consiste en un sólo elemento que ocupa todo el dominio de manera que las mallas G^R son mallas uniformes de $2^R \times 2^R$ elementos, $(2^R + 1)^2$ nodos y $(2^R - 1)^2$ grados de libertad (los nodos interiores). Además para estas mallas $\gamma(G^R) = R$. Se aplicó el MMG hasta $R = 7$ es decir, una malla de $2^7 \times 2^7 = 16\,384$ elementos y $(2^7 - 1)^2 = 16\,129$ grados de libertad. En la figura 3.a vemos la curva de evolución del error como función del tiempo de CPU. Cada punto de la curva corresponde a un R . La ordenada corresponde a:

$$\log_{10} \left(\sum_{v^i} |u(v^i) - u_k^R(v^i)|^2 \right)^{1/2}$$

De la gráfica se deduce que:

- i) el tiempo de resolución es proporcional al número de nodos. En particular, el programa consume 240 seg para $R = 7$ y 70 seg para $R = 6$ (4 096 elementos; 3 969 grados de libertad). La factorización de la matriz correspondiente a G^6 (3 969 GDL) por el algoritmo de Cholesky consumió 300 seg de CPU. La memoria requerida por el MMG fue de 110 000 reales (≈ 880 kbytes en doble precisión) para G^6 y 450 000 reales (≈ 3.6 Mbytes) en G^7 . Nótese el comportamiento lineal de la memoria requerida con el número de GDL. Con respecto a la factorización de Cholesky, un pequeño cálculo permite verificar que sólo el almacenamiento en banda se la matriz requiere:

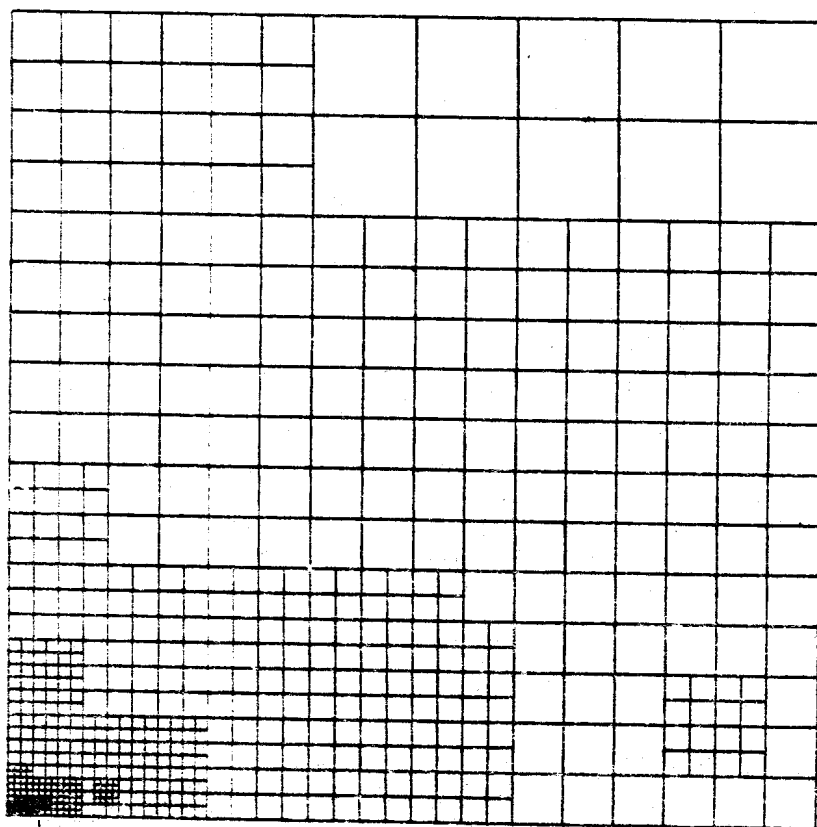
$$(2^6 - 1)^3 = 250\,047 \text{ reales} \approx 2 \text{ Mbytes}$$

en G^6 y 16.4 Mbytes en G^7 . Debido al alto requerimiento en memoria el problema no pudo ser tratado por el método directo en G^7 .

- ii) El error se comportó $O(h^2)$, ($h = 2^{-7}$); Obsérvese que en cada malla el error se reducía en un factor ≈ 4 .

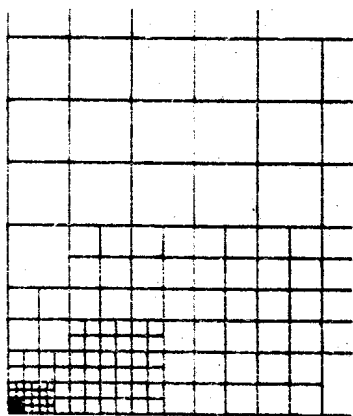
El segundo ejemplo consiste en el mismo problema con $\alpha = 0.5$ esta vez y el objetivo consiste en verificar la malla generada por el proceso de RA ante la presencia de una singularidad. El detalle de la malla después de 12 refinamientos ($R = 12$) puede observarse en la figura 4.a. Las figuras 4.b y 4.c son detalles de la malla cerca de la singularidad:

$$0 \leq x \leq 0.040, \quad 0 \leq y \leq 0.040 \quad \text{Figura 4.b}$$



A

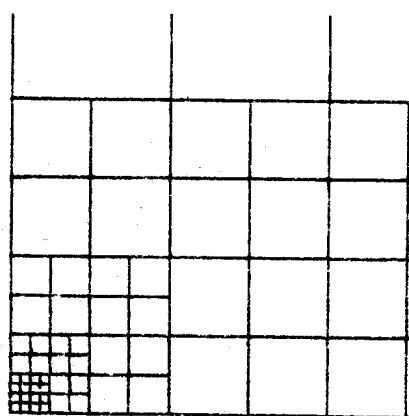
a. malla completa



B

b) detalle X10

A



B

c) detalle X100

Figura 4. Malla generada durante el proceso de refinamiento

$0 \leq x \leq 0.004, 0 \leq y \leq 0.004$ Figura 4.c

En la figura 3.b puede observarse la curva de convergencia (a la solución exacta del problema continuo) como en el ejemplo anterior. Puede observarse que la convergencia es bastante buena a pesar de la singularidad (2 órdenes de magnitud en 180 seg de CPU).

5. NOTACION Y SIMBOLOS

- A = matriz de rigidez obtenida por el MEF
- b = vector miembro derecho en la ecuación del MEF
- d = dimensión del espacio
- e = índice de elemento
- $E \approx$ error de interpolación
- EEAP = Estimador de Error A Posteriori
- G = malla admisible
- GDL = Grado De Libertad
- h = paso de la malla
- IE = Indicador de Error
- I_t^{t-1} = operador de restricción de malla *fina a gruesa*
- I_{t-1}^t = operador de interpolación de malla *gruesa a fina*
- k_F, k_B, k_R = describen el status de un nodo $\forall \ell$
- ℓ = subíndice que indica el nivel dentro del MMG
- MDF = Método de Diferencias Finitas
- MEF = Método de Elementos Finitos
- MMG = Método MultiGrilla
- n = relación entre el número de nodos de G_t y G_{t-1} (asintóticamente)
- N = número de nodos de la malla
- N_e = número de nodos del elemento e
- N_B = número de elementos de la malla
- N_k = función de forma asociada al nodo k
- \mathcal{N} = conjunto de nodos asociado a una malla admisible.
- R = supraíndice que indica el paso de refinamiento
- RA = Refinamiento Adaptivo
- r = vector residuo de la ecuación del MEF
- u = vector de valores nodales del MEF
- u_I = interpolada de u por el MEF
- u_h^R = función resultado del MEF en G^R
- v_e^j = nodo j del elemento e
- w = trabajo computacional por elemento
- w_R = trabajo computacional por nodo por relajación
- W_t = trabajo computacional en la malla G_t
- x_j = autovector de la matriz A
- α = orden de convergencia de los métodos iterativos; también exponente de la función compleja en los ejemplos
- ϵ_F = criterio de detención en la malla gruesa
- Φ = magnitud de las derivadas segundas de u
- γ = máximo nivel de una malla
- κ = número de condición de A

λ = multiplicador de Lagrange en la discretización óptima
 λ_j = autovalor de la matriz A
 ν = número de relajaciones por ciclo del MMG
 ρ = factor de reducción del residuo por ciclo del MMG
 ω = parámetro del método de relajación
 Ω_e = volumen del elemento e
 η_e = "eficiencia" del elemento e

6. BIBLIOGRAFIA

- [1] "Un schéma multigrille en éléments finis décentré pour les équations d'Euler", por Marie-Hélène Lallemand, Fatima Fezoui y Eric Perez, Rapport de Recherche N° 602 del INRIA, (Rocquencourt, Francia), (diciembre 1986)
- [2] "Multigrid Methods", editado por W.Hackbusch & U.Trottenberg, Springer-Verlag, (1986)
- [3] "A Multigrid Finite Element Method for the Transonic Potential Equation", por H.Deconinck & C.Hirsch en [2].
- [4] "Refinement Algorithms and data Structures for Regular Local Mesh Refinement", por R. E. Bank, A. H. Sherman and A. Weiser en "Scientific Computing", editado por R. S. Stepleman, North-Holland (1983)
- [5] "Multi Level Adaptive Solutions to Boundary-Value Problems", por Achi Brandt, en Math. of Comp., vol. 31, nro. 138, pgs. 333-390
- [6] "Some A Posteriori Error Estimators for Elliptic Partial Differential Equations", por R.E.Bank & A.Weiser, Math. of Comp., vol 44, nro 170, pgs. 283-301 (1985)
- [7] "A Posteriori Error Analysis of Finite Element Solutions for One-Dimensional Problems", por I.Babuska & W.Rheinboldt, Siam J.Numer.Anal., vol 18, (1981)
- [8] "On Multigrid and Related Techniques for Solving Discrete Elliptic Systems", por R. A. Nicolaides, Journal of Comp. Phys., vol 19 (1975), pgs. 418-431
- [9] "Applications of Multi-Grid Methods for Transonic Flow Calculations", por W.Schmidt & A.Jameson en [2]
- [10] "Multigrid Methods for Integral and Differential Equations", editado por D.J.Paddon & H.Holstein, Clarendon Press-Oxford (1985)
- [11] "Multigrid acceleration of an Iterative Method with Application to Transonic Potential Flow", por Z.Nowak & P.Wesseling, Comp. Meths. in App. Sci. Eng., vol VI, editado por R.Glowinski & J. L. Lions, North-Holland (1984)
- [12] "Une méthode multigrille pour la résolution des équations de Navier-Stokes compressible par des éléments finis", por Françoise Angrand y Pénélope Leyland, Rapport de Recherche N° 593 del INRIA (Rocquencourt, Francia) (diciembre 1986)