

REDES ADAPTADAS A UNA SOLUCION.  
SU INCLUSION EN PROGRAMAS DE CÁLCULO.

Marcelo J. Vénere

Seccion Mecánica Computacional  
Centro Atómico Bariloche - CNEA  
8400 - Rio Negro. Argentina.

RESUMEN

Se presentan dos métodos para densificar redes de Elementos Finitos en forma automática, adaptandola a una solución calculada previamente. El primero consiste en desplazar los nodos interiores de la red de forma de densificar las zonas con alto error a costa de disminuirla en las que el error es bajo. El segundo método consiste en insertar nuevos elementos en las zonas en que se necesita densificar.

Se describirá la forma de incluir estos métodos en programas de cálculo, sus ventajas y desventajas, y se mostraran algunos ejemplos de aplicación práctica.

ABSTRACT

Two methods for self-adaptive refinement of finite element grids are presented. In the first one, the non-boundary nodes are displaced according to an error estimator, so that the regions with high error become more densified than those where the error is small. In the second one, new elements are inserted in the regions where densification is needed.

A discussion follows on practical implementation of both methods in finite element software. Numerical results are shown.

## INTRODUCCION

Cuando se trabaja con Elementos Finitos, el problema de generar redes densificadas adecuadamente para el problema que se desea resolver, es en general una tarea difícil, aun si se dispone de un buen generador de redes. El problema es aun peor en los casos en que no se conoce a priori cual es la zona donde se espera tener problemas, o en los que esta zona cambia durante el cálculo.

Otro problema al que se suele enfrentar el usuario de Elementos Finitos es saber si la red con la que esta trabajando es lo suficientemente densa, y no esta introduciendo un error de truncamiento elevado. Típicamente esto se soluciona con experiencia del usuario o con redes desproporcionadamente densas.

A excepción quizás del método de multi-mallas, no existen en la actualidad programas que independicen al usuario de estos problemas, y es probable que una completa automatización este aun distante de lograrse. En este trabajo se presentan dos métodos que permiten manejar estos problemas, si bien no en forma automática y "a prueba de bobos", si en forma sencilla y con poco trabajo para el usuario.

Estos métodos pueden incorporarse a un generador de redes sin dificultad, pero su inclusión en programas de cálculo resulta un poco más compleja, y va a depender además de la ecuación diferencial que se este resolviendo. En este trabajo se describirá la implementación dentro del programa CUARM (Basombrio 1977), que permite resolver el siguiente problema:

$$(b.U_x)_x + (b.U_y)_y = g \quad (I)$$

con:  $b = b(x,y,U)$

$g = g(x,y,U)$

Para poder realizar una densificación adaptada a la solución va a ser necesario estimar el error de esta, e identificar así las zonas con problemas. Lo que se necesita es estimar el error en el elemento en

terminos relativos a los demas elementos, y en forma poco costosa en tiempo de computación.

Una de las formas mas sencillas y "baratas" computacionalmente es usar como estimador al gradiente de la solución. Una mejor opción para la ecuación (I) es utilizar el siguiente estimador (Rivara 1984):

$$|e|_t = A_t * (C1 * J_s + C2 * |g|_t)$$

s t

Donde: t → elemento  
s → lado de t  
A<sub>t</sub> → area del elemento  
C1, C2 → Constantes  
J<sub>s</sub> → Diferencia entre el gradiente del elemento "t" y  
y el elemento vecino por el lado "s"

#### DESPLAZAMIENTO ADAPTIVO

Esta idea consiste en mover los nodos libres de la red de forma de densificar las zonas con alto error a costa de disminuirla en las que el error es bajo. Por nodos libres se entiende a aquellos que no pertenecen a la frontera (donde hay condiciones de contorno aplicadas).

Existen muchas transformaciones que pueden cumplir con este objetivo; la implementada en este trabajo es una variante de las llamadas transformaciones Laplacianas, donde el coeficiente de la ecuación no es considerado constante, sino dependiente de las coordenadas (ecuación cuasi-armonica).

$$\begin{aligned} (b.X_u)_u + (b.X_v)_v &= 0 & \text{(II)} \\ (b.Y_u)_u + (b.Y_v)_v &= 0 \end{aligned}$$

con:  $X_{\text{frontera}} = X_0$   
 $Y_{\text{frontera}} = Y_0$

y  $b = b(u,v)$

Para que el desplazamiento de los nodos se adapte a la solución calculada se puede tomar:

$b = |e|$  donde  $|e|$  = error estimado

Las ecuaciones (II) se resolvieron utilizando Elementos Finitos, y la técnica de almacenamiento "Sky-line"; y para resolver el sistema de ecuaciones resultante se utiliza el método de eliminación de Gauss.

En la figura 1 se puede ver el esquema de la rutina que realiza este desplazamiento adaptivo. Obsérvese que el ensamblaje y la triangularización de la matriz solo se realiza una vez, ya que los dos sistemas solo difieren en los valores de las condiciones de contorno.

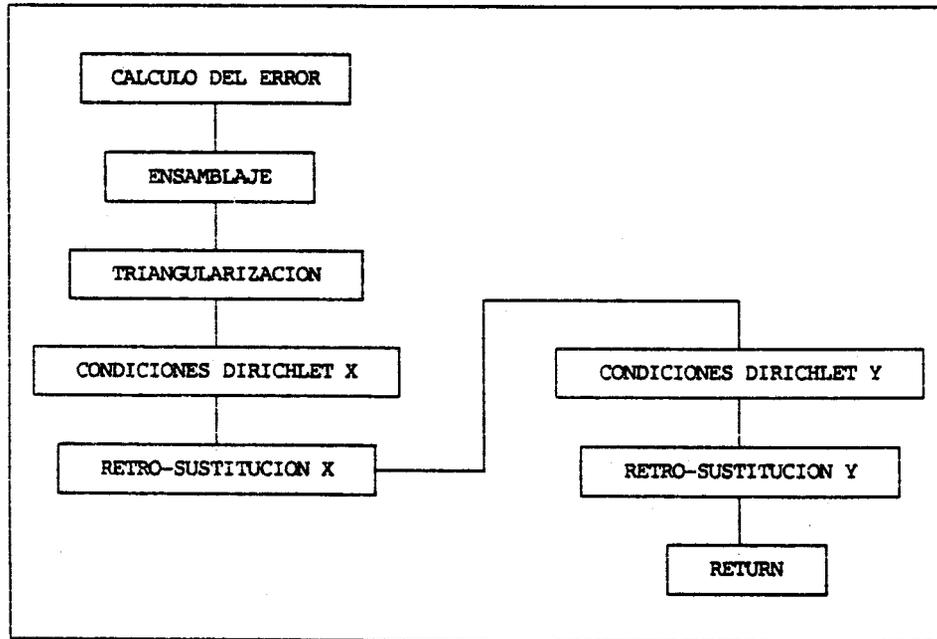


Figura 1: Procedimiento de cálculo para desplazamiento adaptivo.

La inclusión dentro de un programa de cálculo es muy sencilla ya que solo hay que llamar a esta rutina (DESADP) la que devolverá los nuevos vectores X, Y de coordenadas. La conectividad de la red seguirá siendo la misma, y por lo tanto tampoco hay que modificar las propiedades de los elementos. Esta es quizás la principal ventaja de este método.

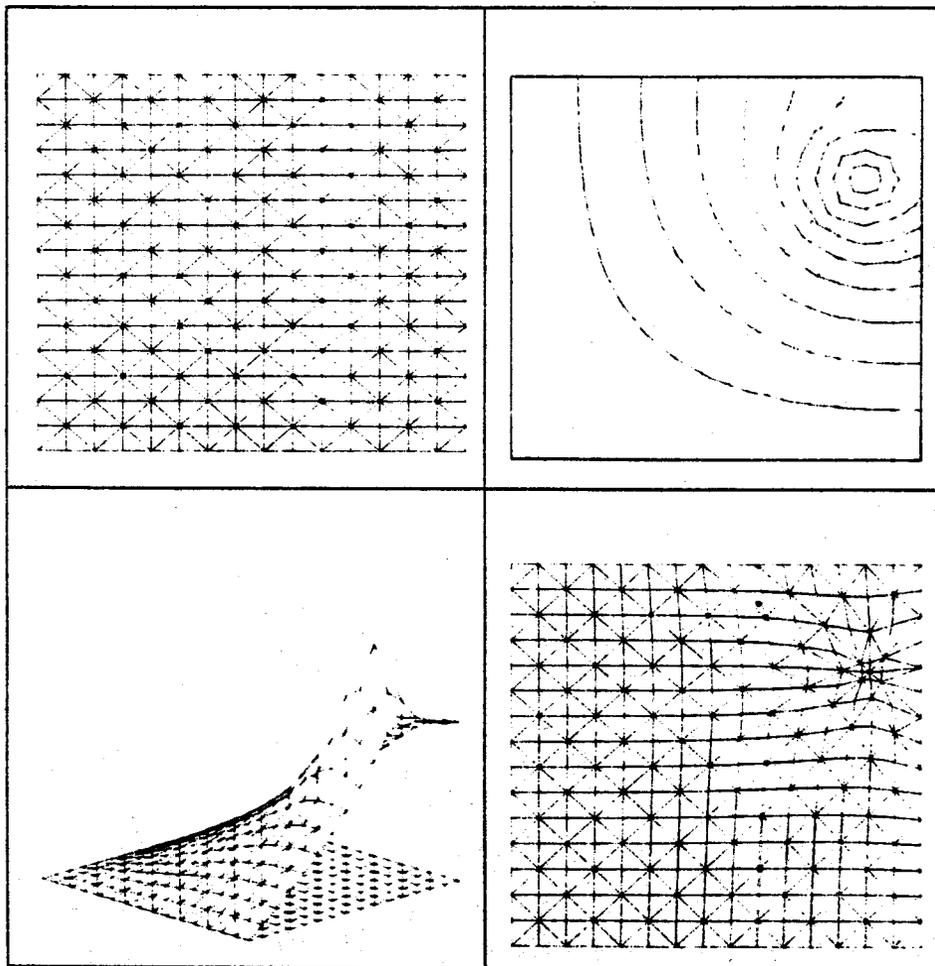


Figura 2: Ejemplo de utilización del método de desplazamiento adaptivo.

En la figura 2 se puede ver un ejemplo sumamente sencillo: Una placa cuadrada con condición de Dirichlet constante en dos lados, condición de Neumann homogénea en los otros dos, y una fuente puntual.

Se muestra la red inicial, la solución vista como una superficie en el espacio y con líneas de nivel, y la nueva red obtenida.

El inconveniente más grave que presenta este método es que la forma de los elementos se degenera, y para grandes desplazamientos es posible que aparezcan elementos de área negativa. Esto se puede solucionar en forma parcial, limitando los valores en el coeficiente "b" de la ecuación (II). En el ejemplo de la figura 2 este coeficiente fue normalizado a 1.0 y luego acotado en la siguiente forma:

$$\text{IF (b .LT. 0.1) b = 0.1}$$

En la figura 3 se muestra el mismo ejemplo anterior pero donde no se acotó este coeficiente. Se puede ver que hay elementos cuya forma ya no es aceptable.

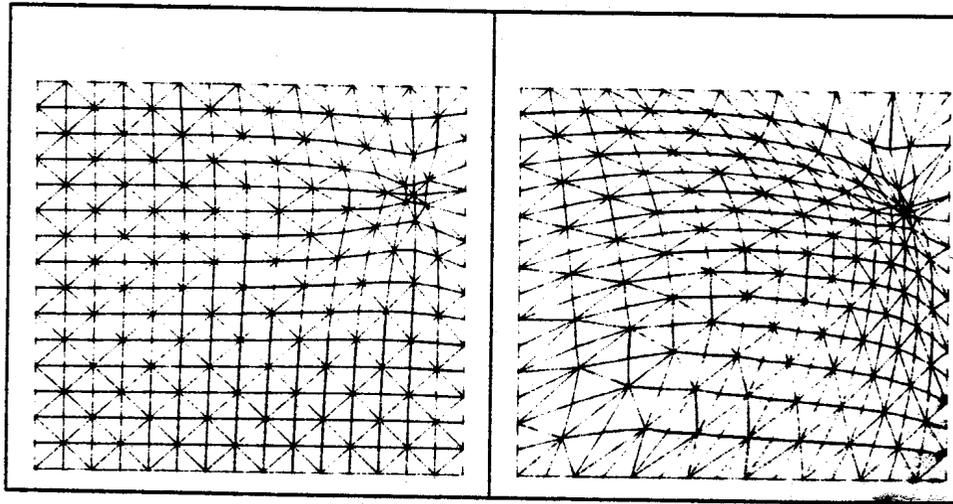


Figura 3: Comparación de la red obtenida acotando y sin acotar el coeficiente "b"

### INSERCIÓN DE ELEMENTOS

Indudablemente los mayores esfuerzos realizados en los últimos tiempos en el área de generación de redes es en el tema de redes no estructuradas. Estas en general se obtienen mediante la inserción de nuevos elementos en algunas zonas de la red (Para mayores detalles ver referencia 2). Si bien han sido propuestas numerosas formas de dividir elementos no triangulares, en este trabajo limito su utilización a los triángulos (lineales o cuadráticos) por ser los únicos que garantizan que la forma de los elementos no se va a degenerar, condición que ya se vio es necesaria para automatizar la densificación.

No todas las divisiones de triángulos van a garantizar que la forma de los nuevos elementos va a ser buena; para esta implementación se eligió la que se muestra en la figura 4, propuesta por Rivara (1987). Al dividir de esta forma a un elemento, se está agregando un nodo al elemento vecino, por lo tanto es necesario realizar luego un proceso de conformación, donde estos elementos son divididos de forma apropiada; nuevamente esta división debe ser tal que garantice la no degeneración de los elementos. En la misma figura 4 se puede ver como se realizan estas nuevas divisiones.

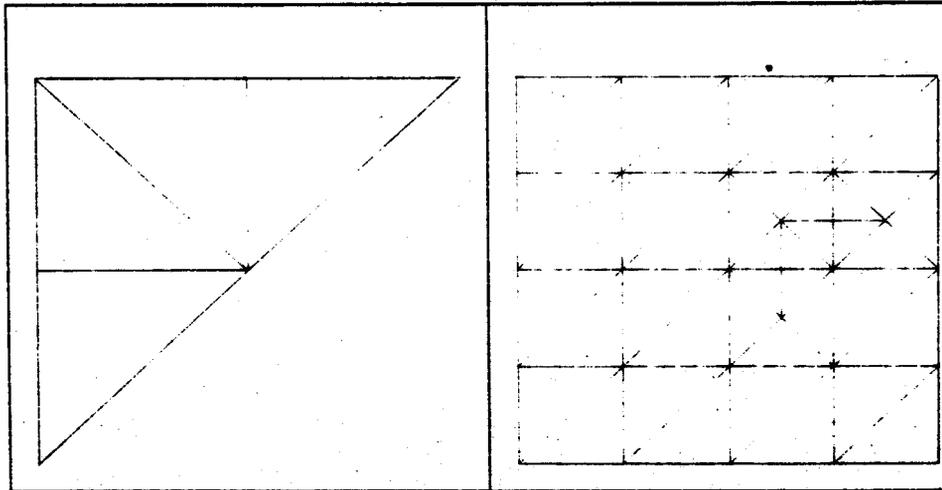


Figura 4: División de elementos triangulares

Esta técnica permite obtener redes de excelente calidad, aun en casos en que sea necesario hacer densificaciones excesivamente altas. Por ejemplo en la red final de la figura 6 hay elementos con área  $10^3$  veces más chica que otros de la misma red, y sin embargo dada la cantidad de nodos de la misma (121), se la puede considerar una red chica. Esto es imposible de lograr con las tradicionales redes estructuradas.

La implementación de este método es mucho más compleja que en el caso anterior, ya que entre otros problemas se están agregando nuevos grados de libertad, lo cual implica cambiar toda la información de entrada del programa de cálculo, y por lo tanto la implementación va a depender fuertemente del programa sobre el que se está trabajando. En la figura 5 se muestra los pasos a seguir en este método para realizar un proceso de inserción.

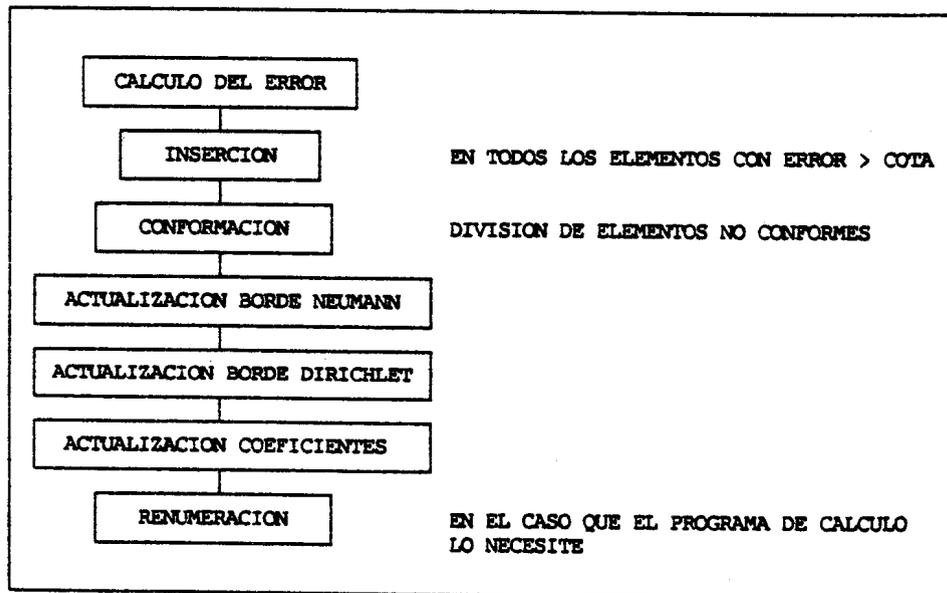


Figura 5: Procedimiento de cálculo para inserción de elementos

En la figura 6 se muestra un ejemplo donde se utilizó este método. Se ha modelado el potencial entre una placa a  $-1316.0$  V y dos cables de

distinto diámetro a 0.0 V. En primera instancia este cálculo se había realizado con una red de  $41 \times 41$  (1681 nodos, 3200 elementos), lo cual muestra una situación típica, ante la inseguridad en la calidad de los resultados, el usuario opta por sobre-dimensionar el problema y evitarse probables complicaciones que pueden llegar a ser difíciles de detectar. La idea, con este método, es que el usuario prepare una red con pocos nodos, y el programa se encargue de densificar si es necesario, y solo donde haga

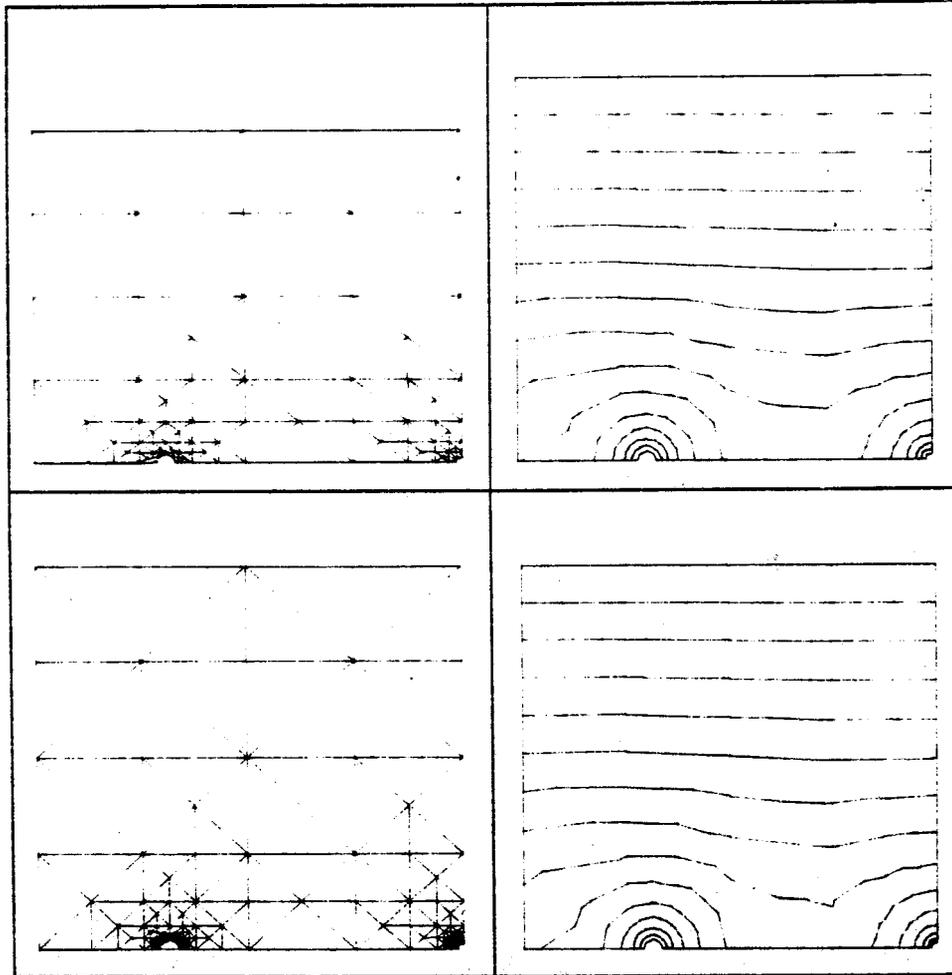


Figura 6: Ejemplo de utilización del método de inserción de elementos

falta, consiguiéndose así no solo un ahorro en tiempo de cálculo, sino también confianza en los resultados obtenidos. En este ejemplo la red inicial ya era bastante buena, y con solo un proceso de inserción bastó para tener un error aceptable.

#### INCORPORACION AL PROGRAMA CUARM

Este programa fue implementado en su primera versión por Basombrio F. y Cruz B. en 1977 con el nombre de NOLICUARM. Posteriormente fue re-programado por Vénere M. en 1986, y por su simpleza resulta ideal para incorporar y probar estos nuevos métodos. Su estructura esta esquematizada en la figura 7.

También se puede ver aquí como se agregaron los módulos de densificación adaptiva. Obsérvese que al módulo de inserción adaptiva es posible llamarlo varias veces durante un cálculo si el error no es aceptable, por ejemplo en el caso que se muestra en la figura 8 se realizaron cinco procesos de inserción hasta alcanzar el error deseado. No se implementó la misma facilidad para desplazamientos adaptivos, ya que se encontró muy poca diferencia entre la red obtenida con un proceso a la que se obtiene iterando algunas veces.

El control para activar un módulo o el otro, se controla con tres parámetros que da el usuario:

- IDESP → Indica si se activa o no el desplazamiento adaptivo.
- NINS → Número máximo de procesos de inserción que se realizarán.
- COTA → Error máximo tolerable. Si  $ERROR > COTA$  inserta elementos.

También es posible combinar ambos métodos en un mismo cálculo; en la figura 9 se puede ver el mismo ejemplo de la figura 2, donde después de desplazar, se realizaron dos procesos de inserción.

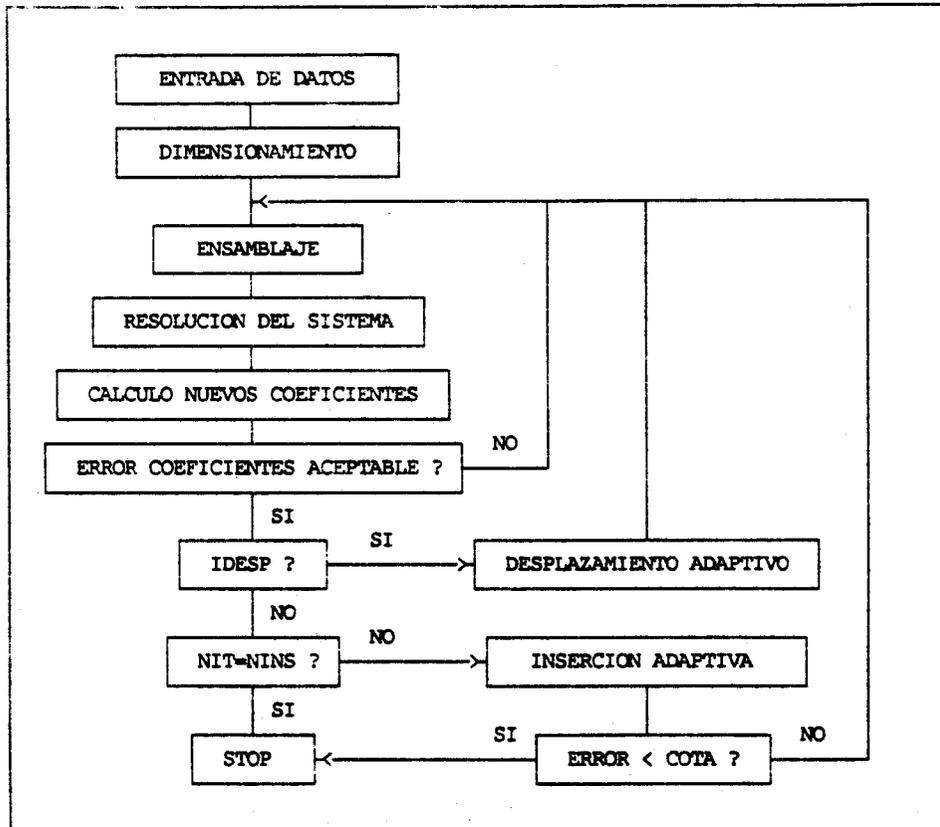


Figura 7: Esquema del programa CUARM-ADAP

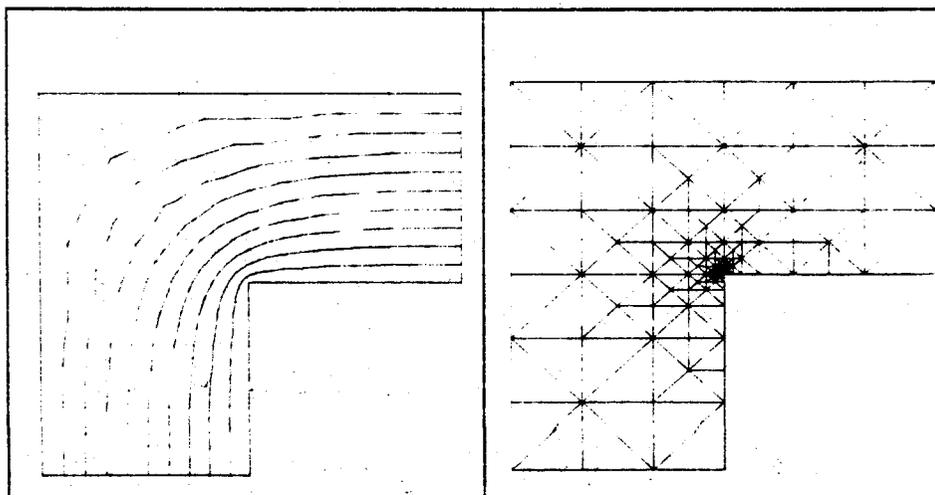


Figura 8: Ejemplo de inserción adaptiva

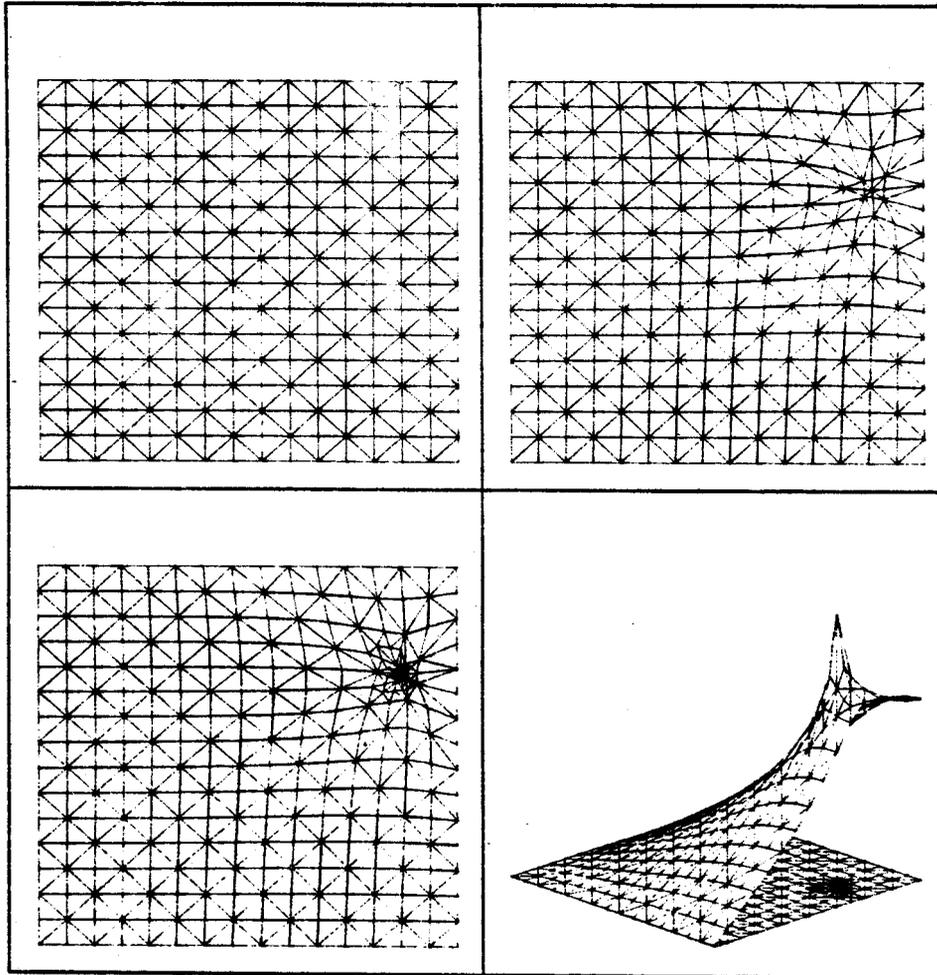


Figura 9: Utilización combinada de ambos métodos.

#### PROBLEMAS NUMERICOS

Estos métodos de densificación adaptiva permiten reducir en forma drástica el tamaño de las redes, y problemas antes reservados a super-computadoras pueden ser ahora tratados en máquinas más accesibles.

Sin embargo es necesario tener ciertos cuidados. Un problema que puede surgir es la precisión con que se trabaja. Típicamente un programa de cálculo implementado en una super-computadora trabaja en doble precisión, sin embargo ese no es el caso si uno trabaja en una VAX o menos aun en una PC.

Si bien las redes van a ser pequeñas, y por lo tanto también el volumen de cálculo, hay magnitudes (como el area de los elementos) que pueden diferir en varios órdenes de magnitud en la misma red. Esto puede introducir errores de redondeo importantes, y es necesario tenerlos en cuenta.

Para analizar este problema se corrió un caso con solución exacta, con distintas redes, en simple y doble precisión. Se encontró que el parámetro que influye mas directamente sobre el error es la dimensión del elemento respecto a las coordenadas en que se encuentra ubicado. Esto es:

$$D_e = \text{Longitud característica} / \text{Coordenadas del centro}$$

Este resultado tiene su explicación en que si bien estas redes tienen pocos nodos, y por lo tanto el volumen de operaciones es pequeño, se estan haciendo operaciones computacionalmente dificiles, como diferencias entre numeros casi iguales, donde el error de redondeo crece muy rapidamente.

En la figura 10 se ha graficado el error relativo máximo en la red, en función de " $D_e$ ". Para graficar el error de redondeo se tomó la diferencia entre la solución exacta y la solución en doble precisión (o sea que se considera que esta última no tiene error de redondeo).

Se puede ver que cuando " $D$ " está por debajo de 0.02, o sea cuando el lado del elemento es 50 veces mas chico que las coordenadas donde se encuentra, y se trabaja en simple precisión ya no es conveniente seguir densificando. Esta cota es la que necesita el usuario para saber cuando debe abandonar la precision simple.

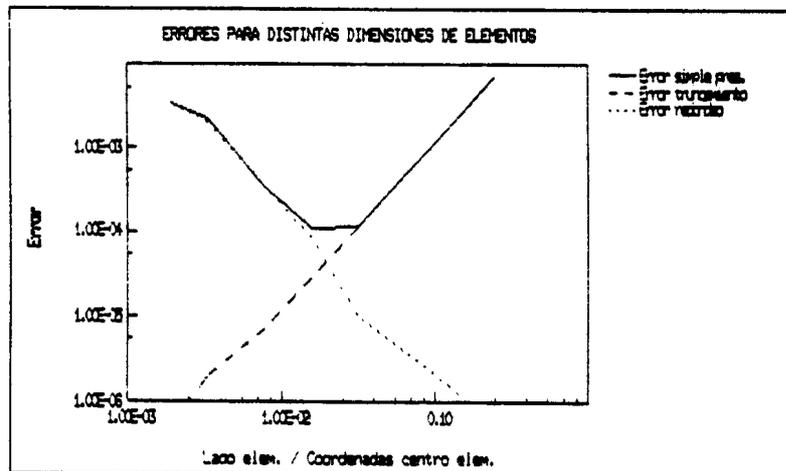


Figura 10: Error de truncamiento y redondeo para distintas redes.

#### REFERENCIAS

- [1] Bascombrío, F., "Resolución por elementos finitos de la ecuación cuasiarmónica bidimensional". CNEA-NT 31/78.
- [2] Rivara, M.C., "Adaptive multigrid software for the finite element method". Doctoral dissertation, K. U. Leuven, Belgium (1984).
- [3] Rivara, M.C., "A grid generator based on 4-triangles conforming mesh-refinement algorithms". Int. Journal for Num. Methods in Eng. Vol 24, pag. 1343-1354.
- [4] I. Babuska and W.C.Rheinboldt, "Error estimates for adaptive finite element computations", SIAM J. Numer. Anal. 15, pag 736-754.