

ADAPTATIVIDAD TRIDIMENSIONAL CON MALLAS NO ESTRUCTURADAS

MARCELO J. VÉNERE*

ROBERTO SALIBA*

y

RAUL FEIJÓO**

* *Centro Atómico Bariloche, CNEA*
8400 - San Carlos de Bariloche, Argentina
Tel.: + 54-944-45107 Fax: + 54-944-45299
E-mail: venere@cab.cnea.edu.ar

** *Laboratório Nacional de Computação Científica*
Río de Janeiro, Brasil

RESUMEN

En el presente trabajo se muestran algunas experiencias realizadas en el área de adaptatividad para problemas tridimensionales. Basados en problemas de elasticidad, se estudiaron dos formas bien diferenciadas de modificar la discretización, de manera de aumentar el número de grados de libertad en las zonas del dominio que presentan mayor error de discretización. La primera está basada en la idea de división de elementos, mientras que en la segunda se regenera nuevamente la malla. Se describirán brevemente ambas técnicas asimismo como las estrategias adoptadas para redefinir la discretización en base a un indicador del error sobre una solución previamente calculada. Por último se muestran algunos ejemplos de aplicación concretos y se discuten las ventajas y desventajas de ambas metodologías.

TRIDIMENSIONAL ADAPTIVITY WITH NON STRUCTURED MESHES

SUMMARY

In this work some experiences on adaptivity for three dimensional problems are presented. Based on a linear elasticity solver, two different approach for the discretization changes are analysed: subdivision of simplexes and remeshing. With the first one, those elements with an estimated error greater than an acceptable value are divided. With the second, the estimated error is used to define a new element size at the centre of each element and a new mesh is generated. Both methods are described and some application examples are presented. Finally the advantage and problems of using these adaptivity techniques are discussed.

Recibido: Febrero 1997

INTRODUCCIÓN

Una de las etapas que presenta mayores dificultades y exige grandes esfuerzos de parte del usuario en el proceso de modelado computacional por elementos finitos de problemas tridimensionales, es la definición de la discretización. Causa de ello son la falta de herramientas de generación robustas, y los problemas inherentes a especificar las características de la discretización en el espacio.

Las técnicas adaptativas se presentan como la mejor solución a estos problemas. La idea básica de las mismas es automatizar la búsqueda de convergencia en red que debe realizarse en todo modelado por elementos finitos. Esta convergencia usualmente se obtenía en forma bastante cualitativa, repitiendo los cálculos utilizando mallas cada vez más densas, hasta que la solución obtenida no cambia demasiado entre un cálculo y otro. Este proceso requiere por un lado un gran esfuerzo de parte del usuario, y además los resultados dependerán fuertemente de la experiencia del mismo.

El primer paso hacia una automatización de este proceso se dió con los trabajos sobre estimación a posteriori del error de discretización en soluciones obtenidas con el método de elementos finitos. Gracias a los mismos era posible conocer, en forma relativa, que error se cometía en cada elemento. Destacan en este campo los trabajos de Babuška^{1,2}. Con esta información y algún método que permita modificar la discretización, es en principio posible realizar la convergencia en red en forma automática. Los primeros trabajos en esa dirección^{5,6,7,8} utilizaban los algoritmos de división de símplexes, y se limitaban al caso bidimensional. Se trataba de una metodología muy simple y dió excelentes resultados.

Simultáneamente aparecieron las técnicas automáticas de generación de mallas (las más importantes son los métodos Delaunay^{10,11} y Frontal¹²), con las cuales lo único que se debe especificar para construir una malla es el contorno de la geometría y la discretización deseada en todo punto del dominio. No demoraron mucho en aparecer los trabajos que utilizaban esta herramienta y un estimador del error para regenerar la malla en forma adaptativa (entre los primeros se encuentra¹³).

Sin embargo, hasta el momento no son tan numerosas las publicaciones que aplican técnicas adaptativas a problemas tridimensionales, y prácticamente no existen para el caso de división de símplexes. En este ámbito sobresalen los trabajos aplicados a problemas de fluidodinámica 3D utilizando el método Frontal^{14,16}.

Más recientemente han aparecido algunos trabajos donde se presentan algoritmos que permiten modificar una dada malla mediante procesos de optimización de la calidad de los elementos^{17,18,19}. Se destaca aquí el trabajo de T. Coupez¹⁷, donde se aplica esta idea a problemas de conformado de piezas, en los que la geometría sufre grandes cambios de forma durante el cálculo.

En el presente trabajo se analizan las dos primeras técnicas aplicadas a casos tridimensionales. En primer lugar se mencionan brevemente los diferentes esquemas adaptativos y se describen los algoritmos de división de símplexes y de remallado para modificar la discretización. Posteriormente se presentan los resultados obtenidos al aplicar estos métodos a problemas de elasticidad lineal y finalmente se discuten los pros y contras de los mismos.

ESQUEMAS ADAPTATIVOS

La Figura 1 muestra un esquema con las diferentes etapas de un proceso de cálculo utilizando el método de elementos finitos, donde se han incluido las posibilidades que fueron mencionadas en la introducción para realizar una adaptación automática de la malla. El mismo corresponde a un análisis estacionario, sin cambios de geometría durante el cálculo y la generación de la malla está basada en métodos automáticos orientados a mallas no estructuradas*. Para este trabajo en particular se han analizado los procesos 1 y 2 de la figura.

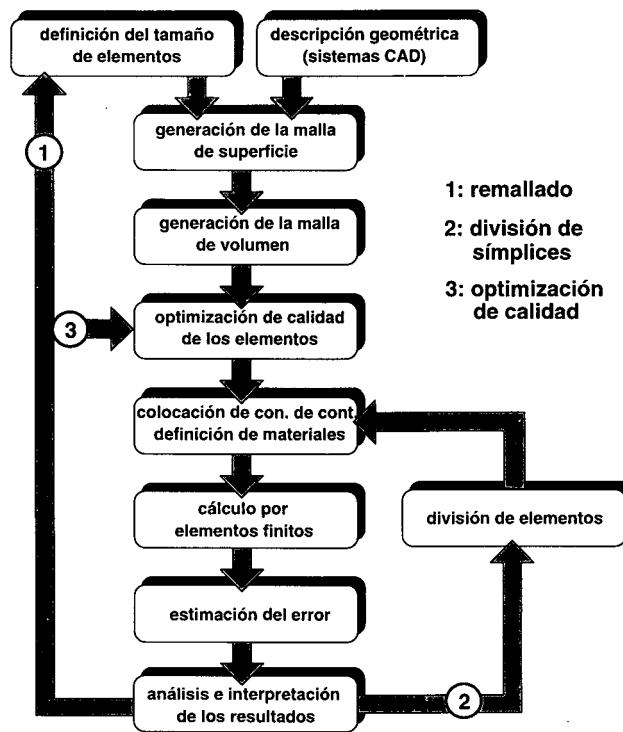


Figura 1. Esquemas adaptativos posibles

Por supuesto este esquema no es único y pueden realizarse variantes. Por ejemplo juntar la etapa de definición de materiales y colocación de condiciones de contorno con la de definición de la geometría, reduciría considerablemente el trabajo del usuario en un proceso adaptativo.

Este esquema presenta algunas etapas con alto costo computacional, y otras con alto costo humano (trabajo del usuario). Entre las primeras están el cálculo por elementos finitos y la optimización de la calidad de los elementos, mientras que las

* Se utilizó el sistema GENESIS 3D, desarrollado en forma conjunta entre investigadores del Centro Atómico Bariloche, Argentina, y del LNCC, Río de Janeiro, Brasil.

que requieren mayor tiempo de parte del usuario son la definición de la geometría, la colocación de condiciones de contorno y el análisis de resultados. Dependiendo de la aplicación puede resultar más interesante utilizar uno u otro esquema, sin embargo la tendencia parece ser cada vez mayor a reducir el trabajo del usuario a expensas de un mayor costo computacional. En realidad, el ideal sería que el usuario se concentre sólo en la etapa de análisis, y todo lo demás se realice automáticamente.

ALGORITMOS PARA MODIFICAR LA DISCRETIZACIÓN

División de símplices

Como se mencionó en la introducción, la primer técnica para modificar la discretización que se empleó en un contexto adaptivo estaba basada en la división de elementos tipo triángulo. El motivo de ello es que en el caso bidimensional, este método presenta una serie de virtudes muy atractivas que se pueden sintetizar en los siguientes puntos:

- Facilidad para obtener una implementación robusta.
- Bajo costo computacional.
- Permite acotar la calidad mínima de los elementos de la malla, independientemente del grado de densificación (relación entre el tamaño de elemento mínimo y máximo en la red.)
- Produce mallas encajadas lo que facilita el proceso de interpolación entre la malla original y la malla hija.

Esto explica la gran cantidad de trabajos que aparecieron en la década del '80 donde se utilizaban diferentes implementaciones de este método. En ese momento se obtuvieron resultados que no tenían precedentes en la época, ya que se conseguían mallas con relación en el tamaño de los elementos de $1/1000$ y más, sin degradar la calidad de los mismos.

Sin embargo la extensión al caso tridimensional de los algoritmos para 2D no son directas y simples. Por ejemplo, no existe una partición de cualquier tetraedro que permita obtener tetraedros semejantes al original, tal como ocurre en 2D. Además aparecen nuevos problemas que dificultan notablemente la obtención de algoritmos robustos, como son la posibilidad de tener aristas no conformes (además de nodos como en 2D) o la no existencia de pruebas matemáticas que permitan garantizar calidades de elementos razonables.

En la Figura 2 se muestran algunas posibilidades para dividir tetraedros obtenidas al extender las particiones más utilizadas en 2D: El caso a) es el más sencillo y consiste en bisectar al tetraedro agregando un nuevo nodo en el medio de la arista más larga. El caso b) es una extensión del algoritmo propuesto por Rivara¹⁹ y se obtiene en tres pasos: dividiendo al tetraedro en dos agregando un nodo en su arista más larga, dividiendo luego cada uno de los dos tetraedros generados agregando un nodo en la más larga de las tres aristas que pertenecían al tetraedro original, y finalmente dividiendo en dos a cada uno de los cuatro tetraedros obtenidos agregando un nodo en el medio de las aristas originales que quedan. En el caso c) se colocan nodos en el medio de las

seis aristas del tetraedro y se forman en primer lugar los cuatro tetraedros semejantes correspondientes a las cuatro esquinas. Lo que resta es un octaedro que puede dividirse en cuatro tetraedros de tres formas diferentes. Entre estas tres formas se puede elegir aquella que permita obtener mejor calidad de elementos.

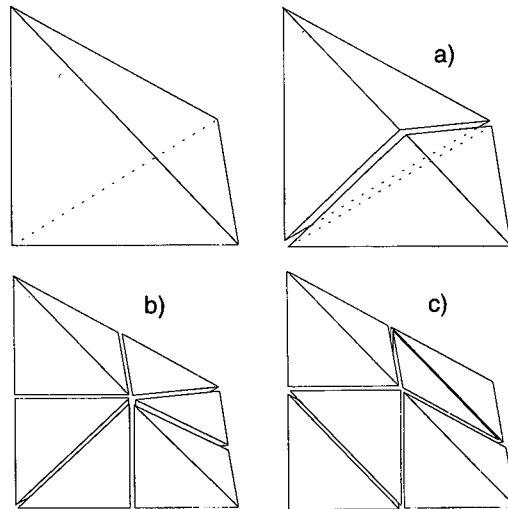


Figura 2. Algunas posibilidades de división de tetraedros

Apoyado en la experiencia obtenida en casos bidimensionales se implementó el algoritmo de la bisección por la arista más larga, el cual puede esquematizarse en la siguiente forma:

```

Crear lista tipo cola con los elementos a dividir
Mientras existan elementos en la cola
  {
    Tomar el primer elemento de la cola
    Cálculo de su arista más larga
    Si la arista no había sido dividida
      Crear un nuevo nodo en el punto medio de la misma
      Dividir al elemento en dos
      Si alguna de las dos caras divididas no estaba marcada como ya dividida
        Agregar al correspondiente elemento vecino a la cola y marcar su cara
        como ya dividida
      Si alguno de los dos nuevos elementos tiene caras no conformes
        Agregar al nuevo elemento a la cola
      Borrar al elemento dividido de la cola
    }
  
```

El algoritmo así planteado funcionará correctamente para mallas donde la arista máxima en cada tetraedro es única. Si esto no es así (por ejemplo un tetraedro

regular, que es el ideal de obtener, tiene todas sus aristas igual es), pueden generarse no conformidades de tipo arista, ya que una misma cara puede dividirse de una forma vista desde un tetraedro y de otra vista desde su vecino. Por ello a la hora de implementar computacionalmente este algoritmo, es preciso tener en cuenta una consideración adicional: *cuando se agrega un nuevo elemento a la cola a causa de que se creó una no conformidad en alguna de sus caras, será necesario indicar también en que forma se debe dividir esta cara.*

Remallado

En este caso la idea es construir nuevamente la malla, pero utilizando un nuevo requerimiento de discretización. Si se observa la Figura 3, puede verse que la generación de una malla implica las siguientes etapas: a) Definición de la geometría, b) especificación de la discretización, c) generación de la malla de superficie, d) generación de la malla de volumen y e) optimización de la calidad de los elementos. Desarrollar estos pasos excede las posibilidades de este trabajo (en 20 puede encontrarse una razonable descripción), sin embargo vale la pena prestar atención a la etapa b), ya que generando esta información a partir del error estimado es posible obtener una malla adaptada a la solución.

Al generar una malla de elementos finitos, ya sea de superficie o de volumen, es preciso especificar el tamaño de los elementos en cada punto del espacio. La metodología más aceptada para ello es definir una función diámetro del elemento $h(x,y,z)$ en un número finito de puntos y extenderla a todo el dominio mediante interpolación. Para realizar esta interpolación, utilizamos el método sugerido en¹³. La idea es en primer lugar generar la malla Delaunay del conjunto de puntos en que se especificó la función $h(x,y,z)$, la que deberá incluir completamente al dominio de interés (normalmente será necesario agregar ocho puntos que sean vértices de una caja que englobe al dominio). Una vez disponible esta malla base, que por construcción es convexa, para calcular el valor de la función h en un punto arbitrario se busca cual es el tetraedro que contiene al punto en cuestión y se interpola linealmente dentro del mismo. Para determinar si el punto está dentro de un dado elemento, se puede calcular el volumen de los cuatro tetraedros formados por las caras del elemento y el punto. Si los cuatro volúmenes son positivos, entonces el punto es interior al tetraedro.

Para hacer el remallado adaptativo, en base al indicador del error en cada elemento se calcula un tamaño deseado h_e y se lo adjudica al centroide del elemento; la malla base que se genera con estos puntos tendrá entonces tantos nodos como elementos haya en la red original. En estas situaciones es importante que la búsqueda del tetraedro que contiene al punto sea muy eficiente. La implementación trivial, que consiste en buscar secuencialmente sobre todos los elementos de la malla base, resulta inaceptable ya que tiene un costo computacional $O(Nb)$ (Nb : número de elementos de la malla base), lo que conduce a un costo global del proceso de generación $O(Nb.N)$, donde N es el número de elementos de la malla a generar, y como $Nb \cong N$, entonces el costo final es $O(N^2)$. Como regla general, en el ámbito de generación de mallas tridimensionales, un costo mayor o igual a $O(N^2)$ resulta prohibitivo, a no ser que nos limitemos a mallas no mayores que unas pocas decenas de miles de elementos.

Aprovechando la propiedad de que la malla base es convexa, es posible obtener un costo de búsqueda $O(Nb^{1/3})$ en 3D, realizando una búsqueda direccional. Para ello se comienza con un elemento cualquiera; si el mismo no contiene al punto (es decir que al menos uno de los cuatro tetraedros formados por las caras del tetraedro analizado y el punto en cuestión tiene volumen negativo) se continúa analizando el elemento vecino por la cara que dió el volumen más negativo. En esta forma se puede continuar hasta encontrar al elemento que lo contiene.* En la Figura 4 se muestra una búsqueda bidimensional donde se han sombreado los elementos que fue necesario analizar hasta encontrar al que contiene a un dado punto. En promedio puede esperarse que la trayectoria desde el elemento inicial al elemento que contiene el punto sea aproximadamente recta, por ello el orden en 3D es $O(Nb^{1/3})$ y en 2D $O(Nb^{1/2})$.

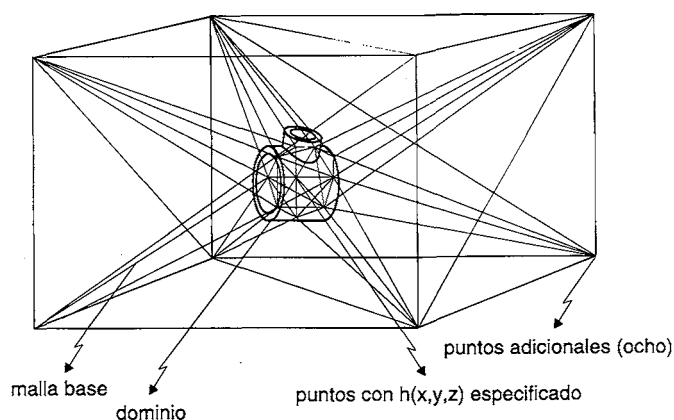


Figura 3. Malla base utilizada para definir el tamaño de elemento en todo el espacio

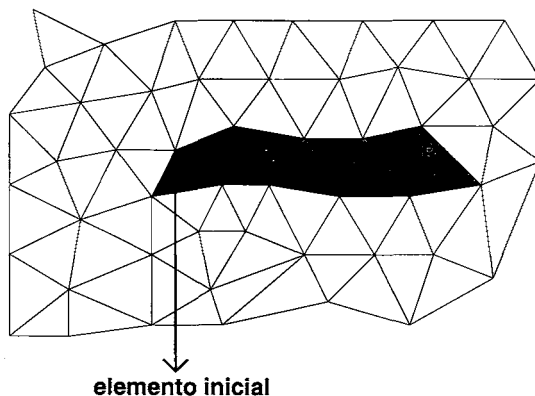


Figura 4. Búsqueda direccional del elemento que contiene un punto dado

* Es importante preveer problemas por los errores de redondeo.

Si bien el orden $O(Nb^{1/3})$ para una búsqueda puede resultar aceptable, es posible obtener un costo inferior recurriendo a técnicas de clasificación geométrica. En²⁰ se estudian diferentes alternativas y se concluye que la mejor opción es utilizar una estructura de datos tipo árboles (quadtrees en 2D y octrees en 3D) y almacenar en cada terminal del mismo, el conjunto de elementos que lo intersectan. El árbol debe ser dividido hasta que queden sólo unos pocos elementos en cada terminal. En esta forma un proceso de búsqueda se divide en dos partes: búsqueda del terminal que contiene al punto y búsqueda del elemento del terminal que contiene al punto. Con este método se consiguió obtener un costo prácticamente constante (rigurosamente el costo computacional es $O(\log Nb)$).

Es posible realizar algunas variantes a este último método que permiten ahorrar bastante memoria (principal desventaja del mismo). Por ejemplo, en vez de almacenar en cada terminal a todos los elementos que lo intersectan, puede almacenarse sólo uno, y a partir de este hacer una búsqueda direccional.

PROBLEMA RESUELTO

Sea Ω un poliedro en \mathbb{R}^3 . Consideramos el problema de elasticidad convencional, con un término adicional en las fuerzas volumétricas que facilita la incorporación de singularidades

$$\begin{cases} -\operatorname{div}(\sigma(u)) = f + \nabla g & \text{en } \Omega \\ u = \delta & \text{en } \partial\Omega_d \\ \sigma(u) \cdot n = \gamma - gn & \text{en } \partial\Omega_n \end{cases} \quad (1)$$

donde $\sigma(u) = \frac{E\nu}{(1+\nu)(1-2\nu)}(\nabla u + \nabla^t u) + \frac{E}{1(1+\nu)}\operatorname{div}(u)I$.

La frontera de Ω es la unión disjunta de $\partial\Omega_i$, $f \in (L^2(\Omega))^3$ y $g \in L^2(\Omega)$.

ESTIMACIÓN DEL ERROR

El estimador del error utilizado sigue las ideas de los trabajos de Babuška^{1,2} y en este caso resulta equivalente al error en la norma de la energía. Para cualquier tetraedro T de la malla, el error estimado (η_T) se puede escribir como

$$\eta_T = \left[\int_T g^2 + h_T^2 \int_T |R|^2 + \frac{1}{2} \sum_{l \in T} h_l \int_l |J_l|^2 \right]^{\frac{1}{2}} \quad (2)$$

donde

$$R = \text{div}(\sigma(u_j)) + f(\text{residuo en el elemento})$$

h_T = longitud de la arista más larga T

h_l = longitud de la arista más larga de la cara l

$$J_l = \begin{cases} [[\sigma(u_j) \cdot n]]_l & \text{si } l \subset \Gamma_{\text{int}} \\ 0 & \text{si } l \subset \partial\Omega_d \text{ (salto en las caras)} \\ 2[\gamma - gn - \sigma(u_j) \cdot n]_l & \text{si } l \subset \partial\Omega_n \end{cases}$$

Además se puede definir un error estimado global como

$$\eta = \left(\sum_T^{NEL} \eta_T^2 \right)^{\frac{1}{2}} \quad (3)$$

Dado el indicador de error, las estrategias para modificar la discretización varían con cada método. En el caso de división de simples, la primera idea utilizada para seleccionar los tetraedros a dividir es tomar aquellos que cumplan $\eta_T/\eta_{\text{max}} \geq \lambda$. Esto presenta dos dificultades, por un lado no es evidente que valor se debe utilizar para λ y por el otro que no hay un criterio de parada, ya que en casos con singularidades este proceso no se detiene nunca.

En este trabajo adoptamos una segunda estrategia de selección propuesta por Zienkiewicz y Zhu⁶, donde asumiendo que el error en la nueva malla estará equidistribuido, es decir

$$\eta = \left(\sum_{\text{New Elems}} (\eta_{Tadm})^2 \right)^{\frac{1}{2}}$$

y teniendo en cuenta que en este problema $\eta_T \cong o(h)$ es posible obtener una aproximación para h_{new} como

$$h_{\text{new}} = \frac{\eta_{Tadm}}{\eta_{Told}} h_{\text{old}} \quad (4)$$

Este valor de h_{new} puede utilizarse como estrategia de selección de los elementos a dividir o como criterio para requerir un nuevo tamaño de elemento.

RESULTADOS

En las experiencias que se incluyen a continuación, el modelado numérico por elementos finitos fue realizado con el código ELAS, que permite resolver problemas de elasticidad lineal en 3D utilizando elementos tipo tetraedros. Todos los casos tratan de problemas con soluciones que presentan fuertes singularidades, ya que son estos los casos donde la adaptatividad juega un papel importante, consiguiendo recuperar el orden de convergencia correspondiente al tipo de elemento que se está utilizando. Los resultados se presentan en forma separada para los dos métodos de densificación.

División de símplices

El primer caso estudiado es uno con solución analítica y que presenta una fuerte singularidad en una línea. En realidad se trata de un problema que puede ser resuelto como 2D, pero en este caso el interés era verificar que el indicador del error funcionaba correctamente y que era posible recuperar el orden óptimo de convergencia. La Figura 5 muestra una sucesión de mallas y correspondientes soluciones para este problema. La primera es la malla inicial, la segunda es la malla luego de 20 procesos de densificación y la tercera luego de 50 procesos. Si bien la malla final obtenida es excelente para el análisis que se pretende realizar, hicieron falta 50 procesos adaptivos, cada uno de los cuales involucra un análisis elástico, cálculo del estimador del error, densificación y reubicación de condiciones de contorno. En otras palabras, si bien es cierto que la malla inicial utilizada era claramente insuficiente, es recomendable que una vez que se obtuvo un indicador de error se realicen varios procesos de densificación de forma de reducir el costo global del proceso.

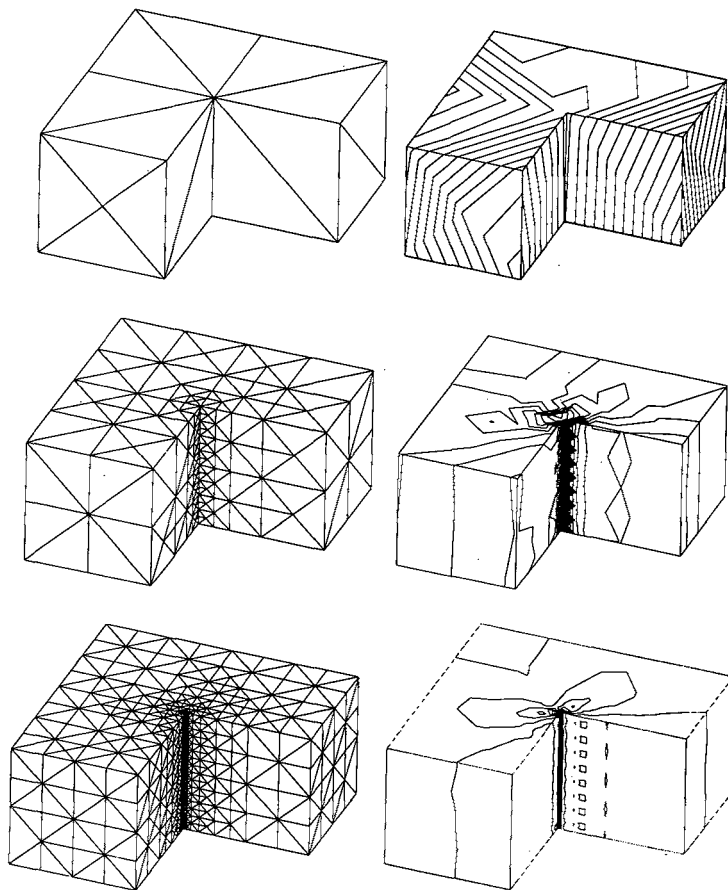


Figura 5. Densificación adaptiva en un problema con fuertes singularidades y con solución analítica.

En la Figura 6 se muestra otro análisis, en este caso netamente tridimensional. El mismo corresponde a una placa sujeta en sus bordes, que está sometida a presión en una de sus caras y que presenta un corte en la misma en forma de pirámide de base cuadrada. Para el cálculo se tomó una simetría $1/4$. En este caso se realizaron 25 iteraciones adaptativas para obtener la malla y solución que se muestran en la figura.

Un punto al que se le debe prestar mucha atención en 3D, es que ocurre con la calidad de los elementos, ya que de nada sirve aumentar el número de grados de libertad en las zonas con alto error, si con ello se generan elementos distorsionados. Por ello se hizo un estudio sobre como evolucionaba la calidad mínima con los diferentes cambios de discretización, y si optimizando esta calidad se consigue mejorar los resultados.

En el seguimiento de la calidad mínima de los elementos generados no fue posible garantizar en forma definitiva que la misma no se degrada en cualquier caso, sin embargo las experiencias numéricas muestran que esta degradación, de existir, comienza a ser importante para un número de divisiones fuera del interés práctico. La calidad media de los elementos mejora en prácticamente la totalidad de los casos.

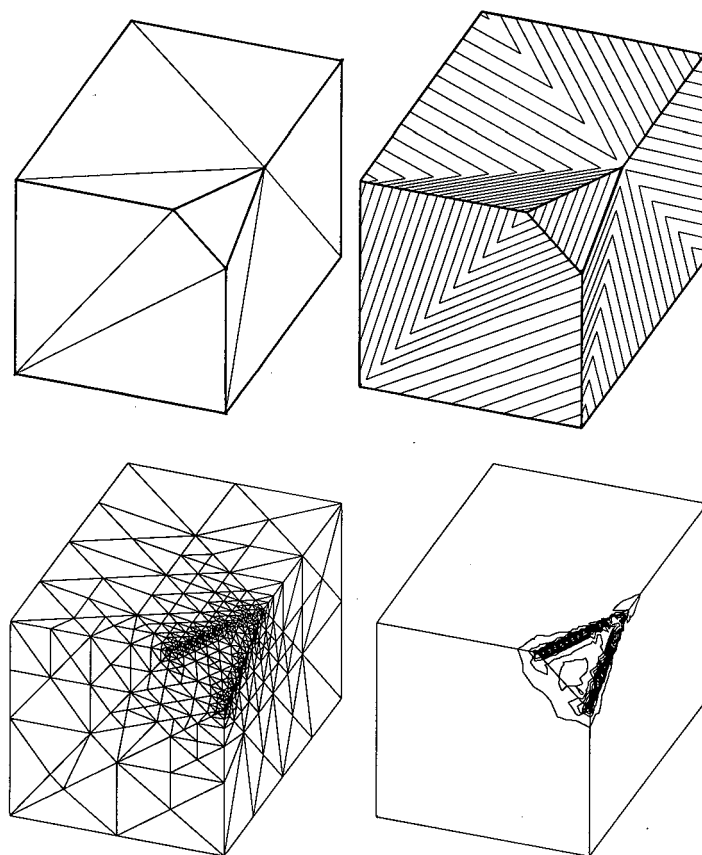


Figura 6. Densificación adaptativa en un problema 3D

En la Figura 7 se muestra como evoluciona el error real (en un problema con solución exacta conocida) durante el proceso adaptativo, con y sin optimización de calidad de elementos. Puede verse que ambas curvas de convergencia coinciden, con lo cual se puede concluir que el método de división de símplexes presenta un buen comportamiento en este aspecto.

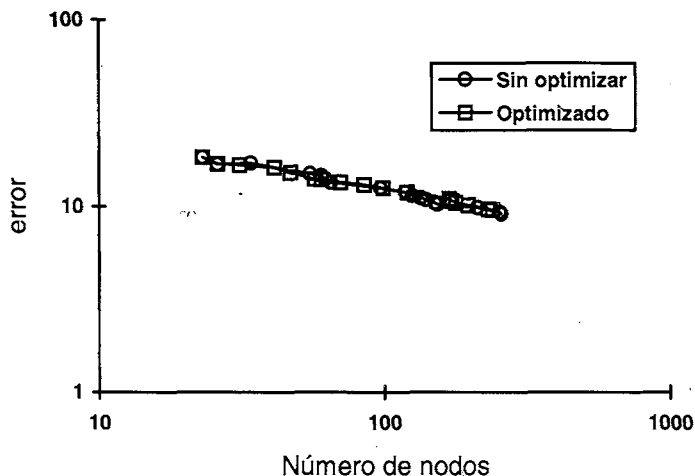


Figura 7. Comparación de la convergencia con y sin optimización de la calidad de la malla

Remallado

La Figura 8 muestra uno de los casos en que se aplicó la idea de remallado adaptativo. La geometría corresponde a una probeta de ensayo para la caracterización de materiales a la propagación de fracturas. Dadas las simetrías presentes, la malla generada corresponde a 1/4 de la pieza.

La primera malla que se generó tiene un tamaño de elemento uniforme (21303 en total). En la figura puede verse que la solución obtenida con la misma está bastante mal definida. A partir de esta última se evaluó el error estimado y utilizando la ecuación (4) se obtuvo los valores de h_{new} para cada elemento. Estos valores fueron "truncados" de la siguiente forma

$$h_{old}/10 < h_{new} < h_{old}^2$$

La cota superior es para evitar tamaños muy grandes que no permitirían representar adecuadamente la geometría, mientras que la cota inferior es a efectos de limitar el número de elementos que tendría la nueva malla.

Con estos valores, se contruyó la malla base y se generó la nueva malla que se observa en la figura (24740 elementos). A pesar de tener prácticamente el mismo tamaño, con esta malla se pudo obtener una solución notablemente mejor.

Por último, se volvió a repetir el proceso, pero con nuevas cotas para truncar el tamaño deseado

$$h_{old}/3 < h_{new} < h_{old}^*2$$

h_{old} corresponde al tamaño del elemento en la malla anterior y no a la malla inicial. Se obtuvo así la última malla con un total de 124197 elementos.

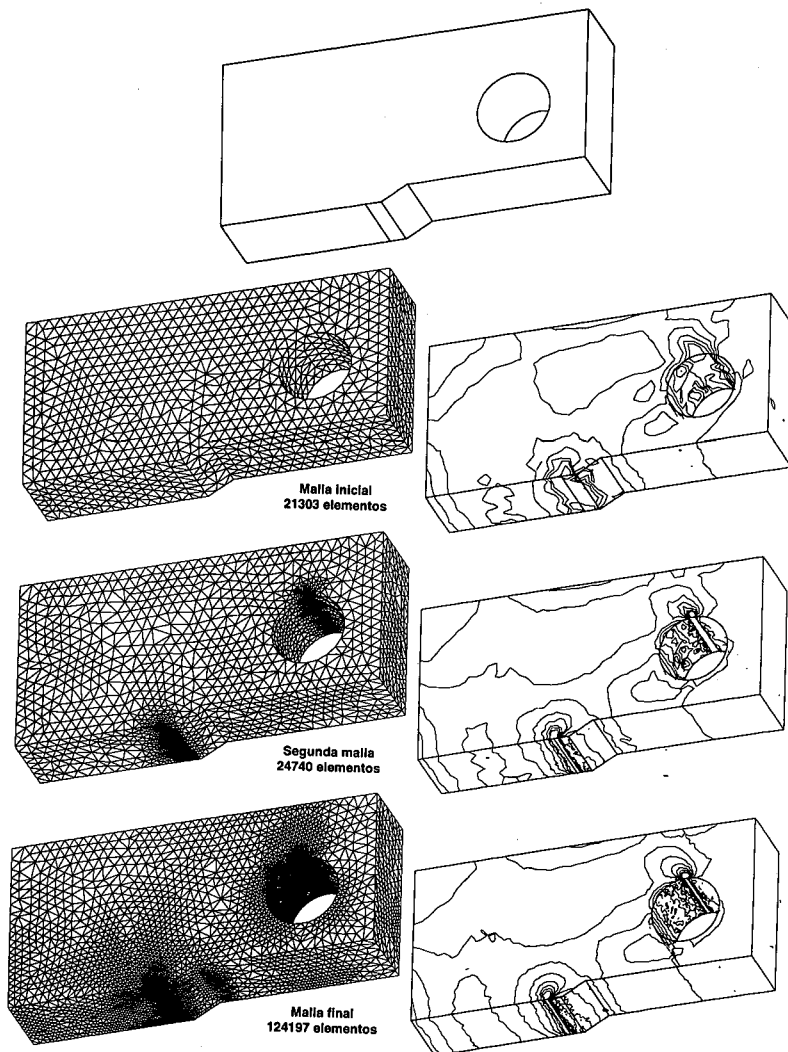


Figura 8. Adaptatividad utilizando remallado

CONCLUSIONES

En este trabajo se han presentado dos estrategias para obtener mallas adaptadas a una solución y se mostró la aplicación de las mismas a problemas concretos. De las experiencias realizadas pueden concluirse los siguientes puntos:

- En todos los casos fue posible obtener las mallas buscadas, tanto con el método de división de símlices como con el de remallado.
- En el caso de división de símlices, la principal objeción es la gran cantidad de procesos que fue preciso realizar para obtener una discretización lo suficientemente densa. Resulta evidente la necesidad de utilizar algoritmos que densifiquen más rápidamente, o en todo caso, realizar varios procesos de división entre cada análisis por EF.
- Con la técnica de remallado, una de las dificultades que surgieron es el control de la cantidad de elementos finales que tendrá la malla. A priori no es evidente como se debe definir la cota inferior para el tamaño de elemento de forma de obtener una malla final con un determinado número de grados de libertad.
- Otra dificultad de la técnica de remallado es el gran tamaño de las mallas base. En tres dimensiones las mallas tienen muchos más elementos que nodos, por lo cual la malla base que se utiliza en un remallado adaptativo es bastante más grande que la malla que se quiere generar. Esto puede solucionarse llevando de alguna forma el indicador de error por elemento a un indicador por nodos. Además como ya se mencionó, es muy importante utilizar algoritmos eficientes para el manejo de esta malla base.

Entre los futuros trabajos que resultaría interesante realizar en este área, sobresale el experimentar el esquema adaptativo basado en métodos de optimización de calidad de elementos. Finalmente, también es importante seguir perfeccionando estos algoritmos de forma de obtener herramientas cada vez más robustas y fáciles de utilizar.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Convenio Bilateral CONICET-CNPq en Mecánica Computacional.

REFERENCIAS

1. I. Babuška y C. Rheinboldt, "Error Estimates for Adaptive Finite Element Computation", *Siam Journal Numerical Analysis*, Vol. 15, pp. 736–754, (1978).
2. I. Babuška, C. Rheinboldt, "A Posteriori Error Estimator in the Finite Element Method", *Inter. J. Num. Meth. Engn.*, Vol. 12, pp. 1578–1615, (1978).

3. O.C. Zienkiewicz y J.Z. Zhu, "A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis", *Int. J. for Num. Methods in Engn.*, Vol. **24**, pp. 337–357, (1987).
4. C. Padr y, M.J. Vénere, On Adaptivity for Diffusion Problems Using Triangular Elements, *Engineering Computations*, Vol. **12**, pp. 75–84, (1995).
5. A. Dervieux, Desideri y Fezoui, "Euler Calculations by Upwind Finite Element Method and Adaptive Mesh Algorithms", *GAMM Workshop on the Numerical Simulation of Compressible Euler Flows*, Rocquencourt, June (1986).
6. O.C. Zienkiewicz y J.Z. Zhu, "A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis", *Int. J. for Num. Methods in Engn.*, Vol. **24**, pp. 337–357, (1987).
7. O.C. Zienkiewicz, Y.C. Liu y G.C. Huang, "Error Estimation and Adaptivity in Flow Formulation for Forming Problems", *Int. J. for Num. Methods in Engn.*, Vol. **25**, pp. 23–42, (1988).
8. K. Eriksson y C. Jhonson, "An Adaptive Finite Element Method for Linear Elliptic Problems", *Math. of Comp.*, Vol. **50**, pp. 736–754, (1988).
9. J. Vénere, "Técnicas adaptivas en cálculo numérico para problemas en dos y tres dimensiones", *Mathématiques Appliquées aux Sciences de l'Ingenieur*, pp. 411–428, (1991).
10. D. Watson, "Computing the n -Dimensional Delaunay Tesselation with Applications to Voronoid Polytopes.", *Computer Journal*, Vol. **24**, pp. 167–172, (1981).
11. D. Cavendish, W. Field y W. Frey, "An Approach to Automatic Three-Dimensional Finite Element Mesh Generation", *Int. J. for Num. Methods in Engn.*, Vol. **21**, pp. 329–347, (1985).
12. S.H. Lo, "A New Mesh Generation Scheme for Arbitrary Planar Domains", *Int. J. for Num. Methods in Engn.*, Vol. **21**, pp. 1403–1426, (1985).
13. J. Peraire, M. Vahdati y K. Morgan y O.C. Zienkiewicz, Adaptive Remeshing for Compressible Flow Computations", *Journal of Comp. Physics*, Vol. **72**, pp. 449–466, (1987).
14. J. Peraire, J. Peiró, L. Formaggia, K. Morgan y O.C. Zienkiewicz, "Finite Element Euler Computations in Three Dimensions", *Int. J. for Num. Methods in Engn.*, Vol. **26**, pp. 2135–2159, (1988).
15. J. Peiró, "A Finite Element Procedure for the Solution of the Euler Equations on Unstructured Meshes", Ph.D. Thesis, Dept. Civil Eng., University College of Swansea, (1989).
16. J. Peraire y J. Peiró, "Adaptive Remeshing for Three-Dimensional Compressible Flow Computations", *Journal of Comp. Physics*, Vol. **103**, (1991).
17. T. Coupez, "A Mesh Improvement Method for 3D Automatic Remeshing", *4th International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, pp. 615–626, (1994).
18. E. Dari y G. Buscaglia, "Topics on Finite Element Meshes for Problems with Moving Boundaries", *VIII International Congress on Finite Elements in Fluids—New Trends and Applications*, Barcelona, 1993, Publicado en: K. Morgan et al. (Eds.), "*Finite Elements in Fluids*", CIMNE, Barcelona, pp. 726–735, (1993).
19. M.J. Vénere, "Optimización de la calidad de mallas de elementos finitos mediante cambios localizados de topología. Aceptado para su publicación en *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, Vol. **13**, No. 1, pp. 3–13, (1996).

20. M.C. Rivara, "Mesh Refinement Processes Based on the Generalised Bisection of Simplices", *Siam J. of Num. Analysis*, Vol. **21**, pp. 604-613, (1981).
21. E.A. Dari y M.J. Vénere, "Algoritmos eficientes para la búsqueda del elemento de una red que contiene un punto dado", *Mecánica Computacional*, Vol. **10**, pp. 455-464, (1990).