# ALGORITMOS EFICIENTES PARA LA BÚSQUEDA DEL ELEMENTO DE UNA RED QUE CONTIENE UN PUNTO DADO

Enzo A. Dari
Marcelo J. Vénere
División Mecánica Computacional, DIA
Centro Atómico Bariloche, CNEA
S.C. de Bariloche, RN, Argentina

## RESUMEN

Dada una red bi-dimensional de elementos finitos y un punto en el plano, interesa encontrar en la forma más rápida posible a cuál elemento pertenece el punto en cuestión.

En este trabajo se describen cuatro técnicas que permiten reducir el orden del algoritmo de búsqueda. Son ellas: zonificación regular homogénea, zonificación regular no-homogénea, árbol cuaternario y una combinación de las primera y tercera.

Se encontró que, para un amplio muestreo de redes, los últimos dos métodos resultan más económicos computacionalmente (tanto en tiempo de CPU como en memoria).

## ABSTRACT

Given a two-dimensional finite element mesh and a point in the plane, we were interested in finding the element to which the point belongs.

In this work four techniques are described that allow to reduce the order of the search algorithm. They are: the uniform grid technique, the quasi-uniform grid technique, the quad-tree technique, and a combination between the first and third ones.

It was found that, for a wide variety of meshes, the last two methods are computationally cheaper both in CPU time and memory requirements.

## INTRODUCCIÓN

Muchas aplicaciones requieren calcular en un punto el valor numérico de una función poliédrica dados los valores en los nodos de una red. Quizás unos de los mejores ejemplos sean algunas de las implementaciones del método de las características para problemas de convección difusión [1]. El valor de la función se obtiene interpolando dentro de cada elemento de la red, por lo cual es necesario en primer lugar encontrar el elemento que contiene al punto.

Esta búsqueda sería muy sencilla y rápida en una red de forma y estructura regular, pero en general las redes utilizadas en elementos finitos son de forma muy irregular, y frecuentemente no-estructuradas.

Si bien el número de operaciones para verificar si un punto esta contenido en un elemento es reducido (por ejemplo: nueve operaciones de punto flotante para triángulos lineales y doce para cuadriláteros bilineales), la búsqueda secuencial en todos los elementos lleva a un algoritmo O(NEL) (crece linealmente con el número de elementos de la red, NEL). Esto resulta inaceptable para los problemas de interés práctico.

En la siguiente sección analizaremos cuatro métodos distintos para reducir el orden del algoritmo de búsqueda, a continuación describiremos el "benchmark" utilizado para comparar la perfomance de cada uno de ellos, y finalmente los resultados obtenidos y las conclusiones.

## CUATRO MÉTODOS EFICIENTES PARA LA BÚSQUEDA

La idea básica para reducir el orden del algoritmo de búsqueda consiste en establecer una correspondencia entre una red estructurada (donde la búsqueda es sencilla y veloz) y nuestra red de elementos finitos. La red estructurada se crea de tal manera que cada una de sus "celdas" tenga asociados unos pocos elementos de la red original.

El proceso de búsqueda ahora se divide en dos etapas: en la primera de ellas se halla la celda de la red auxiliar que contiene al punto; en la segunda se realiza una búsqueda secuencial en los elementos de la red original asociados con dicha celda.

En cualquiera de estos métodos aparece una nueva etapa en el proceso total (etapa de pre-clasificación), que consiste en la generación de la red auxiliar y el establecimiento de la correspondencia celda-elemento. Aunque frecuentemente el tiempo empleado en esta etapa se hace despreciable cuando el número de puntos a buscar es grande, la memoria ocupada por la red auxiliar puede resultar un problema en alguno de los métodos. Desde el punto de vista de la implementación es más sencillo determinar en primer lugar la matriz de incidencia elemento-celda (en formato ralo) y trasponerla para obtener la correspondencia celda-elemento.

Los distintos métodos implementados difieren en el tipo de red estructurada auxiliar que se elija.

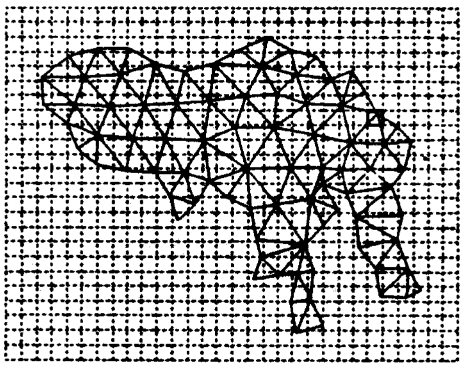
**Método 1: Zonificación regular homogénea [2]**

Se crea una malla rectangular de celdas iguales y aproximadamente cuadradas que contenga la red. Esta malla queda definida dados los extremos y el número de divisiones en cada dirección. Se utilizaron celdas cuadradas ya que se supone que los elementos de la red son de buena calidad (alta relación área/perímetro) y es deseable que la red auxiliar se ajuste a la red original para así obtener pocos elementos asociados a cada celda.

El parámetro a ajustar en este método es el tamaño de la celda. Obsérvese que si se toman celdas infinitamente pequeñas, prácticamente todas contendrían sólo un elemento, por lo tanto al determinar qué celda contiene al punto se conoce al elemento buscado. Sin embargo este ideal requiere de infinita memoria. El caso opuesto consiste en tomar una sola celda que contiene a toda la red, con lo cual el tiempo de búsqueda es nuevamente O(NEL).

En nuestro caso hemos definido el tamaño de la celda de tal manera que su área sea una fracción de la del elemento promedio de la red. Esta fracción se puede ajustar según la memoria disponible.

En la figura 1 se muestra la malla regular asociada a una red y qué elementos de la red están relacionados a una de las celdas.



Celda 505:

Elementos Asociados:

27, 16, 60, 15

**Figura 1: Red auxiliar regular homogénea**

La identificación de la celda que contiene al punto (X,Y) requiere en este caso solo cuatro operaciones de punto flotante:

$$i = \mathrm{int}(\ (X-X_{min})/DX)$$
$$j = \mathrm{int}(\ (Y-Y_{min})/DY)$$

donde: Xmin, Ymin son los mínimos de la malla regular, y DX, DY son los tamaños de las celdas.

nro de celda = j*nx + i

donde nx es el número de celdas en dirección X.

La principal desventaja de este método surge al tratar con redes cuyos elementos poseen tamaños muy dispares, ya que en tal caso para obtener pocos elementos asociados a cada una de las celdas se necesitará una malla regular excesivamente densa.

Método 2: Zonificación regular no-homogénea

Una forma de reducir la desventaja del método anterior es utilizar una red regular no-homogénea, donde las divisiones en cada eje no se realizan a intervalos iguales sino en los lugares donde hay límites entre elementos, fijando una cota mínima para el tamaño de cada celda [3]. Esta idea puede verse claramente en la figura 2.

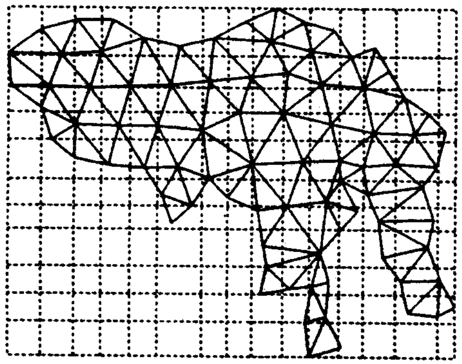


Figura 2: Red auxiliar regular no homogénea

De esta manera se logra reducir el número de elementos asociados a cada celda, aunque para redes con densificaciones puntuales el problema persiste. Además, la identificación de la celda que contiene al punto ya no es directa, sino que es por lo menos de O(log(Ni)) (Número de celdas en cada dirección).

Método 3: Árbol cuaternario

En este caso la red auxiliar está estructurada en forma de árbol cuaternario [4], lo cual le permite adaptarse muy bien a la red original, aún si ésta presenta densificaciones puntuales.

El árbol se crea de tal manera que sus elementos sean cuadrados (la raiz es un cuadrado que contiene a la red), y sus hojas se dividen hasta que cada una de ellas contenga a lo sumo un nodo de la red original; de esta manera se logra la "adaptación" de la red auxiliar a la original.

En la etapa de búsqueda es necesario conocer la estructura lógica del arbol y las coordenadas medias de cada rama, además de la matriz celda-elemento. La identificación de la celda que contiene al punto se efectúa recorriendo el árbol, con lo cual el costo computacional es proporcional al número de niveles. Este número depende del Log (tamaño del recinto / tamaño del elemento mínimo).

En la figura 3 se puede ver una red y el árbol cuaternario utilizado como red auxiliar.

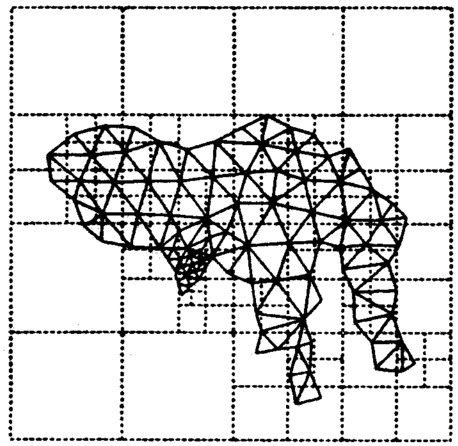


Figura 3: Red auxiliar con estructura de árbol cuaternario

Con este método la memoria utilizada depende fundamentalmente del número de elementos de la red y no de su forma y/o disparidad del tamaño de sus elementos.

**Método 4: Bosque de Árboles cuaternarios**

Este método intenta combinar las ventajas del primero y tercero anteriormente mencionados. Consiste en crear una red auxiliar regular como en el método 1, la cual es muy apropiada cuando la red original tiene elementos de tamaño aproximadamente uniforme. Cada celda de esta red regular es considerada como la raíz de un árbol cuaternario que se divide hasta que en cada hoja quede a lo sumo un nodo, con esto se logra que la red auxiliar se adapte a la original aún cuando hay densificaciones locales.

El parámetro a ajustar es nuevamente el tamaño de celda de la red regular, el mismo se toma de tal manera que el área de cada celda sea una fracción del área del elemento más grande de la red. Para redes cuasi-uniformes cada árbol contará muy probablemente con sólo un nivel de subdivisión, y la primer etapa de búsqueda se llevará a cabo en un tiempo constante. Para redes con densificaciones cada árbol se dividirá lo necesario para adaptarse a la zona de la red original que le corresponda y el tiempo de la primera etapa de la búsqueda será proporcional al Log (tamaño elemento máximo/tamaño elemento mínimo).

En la figura 4 se muestra una red de elementos finitos triangulares con densificaciones locales y la red auxiliar con estructura de bosque de árboles cuaternarios correspondiente.

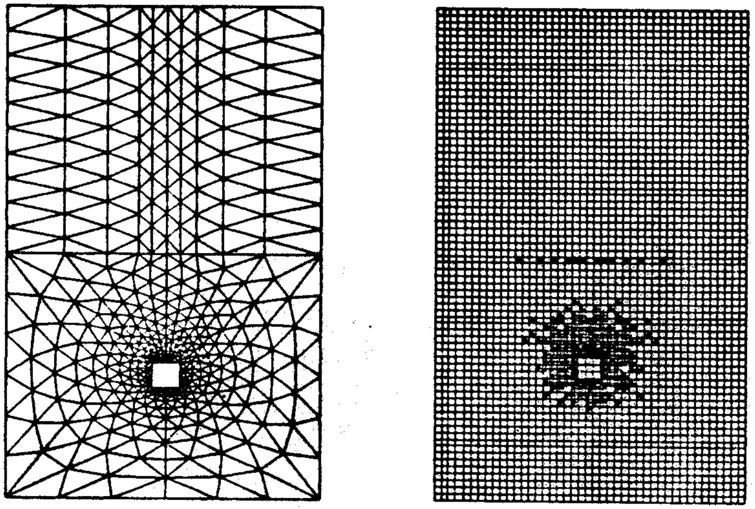


Figura 4: Red auxiliar del 4to método

EL PROBLEMA TEST

A efectos de comparar la perfomance de los distintos métodos implementados se generaron redes no estructuradas sobre un recinto en forma de "L". Como se puede ver en la figura 5 se tomaron redes de distinto número de elementos y

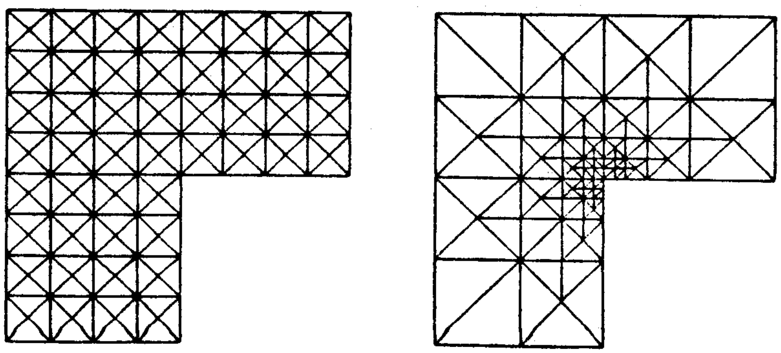


Figura 5: Recinto de cálculo y un par de redes del problema test

con distintos grados de densificación en el ángulo interior, utilizándose triángulos debido a su gran flexibilidad para densificar altamente en algunas zonas. Los puntos que se eligieron para encontrar el elemento que los contiene estaban distribuidos con una densidad igual a la de la red donde se los busca; esto se implementó haciendo que estos puntos correspondan a la posición de los nodos desplazados levemente.

## RESULTADOS

Todos los casos fueron testeados en una VAX 11-780 (fpa). Los tiempos indicados corresponden a una búsqueda, habiéndose promediando con un número significativo de puntos.

En las figuras 6 a 9 se muestran los tiempos de búsqueda para los distintos métodos, en función del número

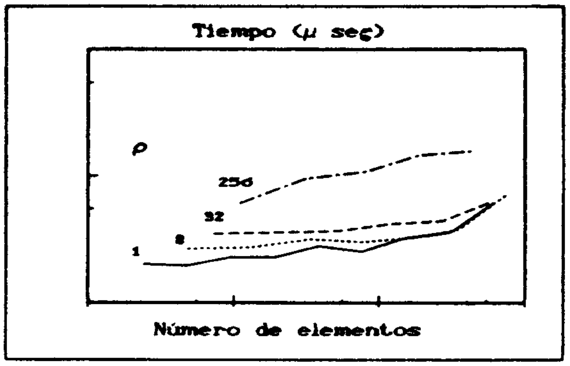



Figura 6: Tiempos de búsqueda del método 1

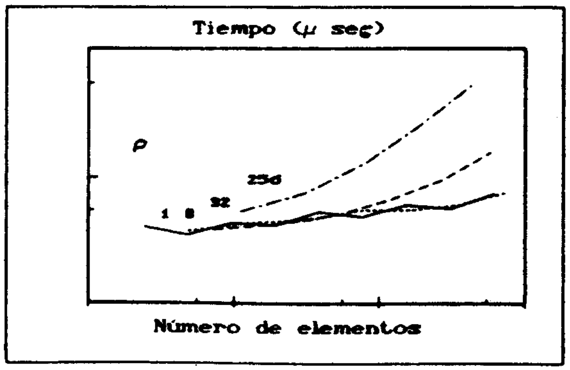



Figura 7: Tiempos de búsqueda del método 2

de elementos, parametrizados con la relación de densificación ($\rho$ = tamaño del elemento mayor de la red / tamaño del elemento más pequeño).

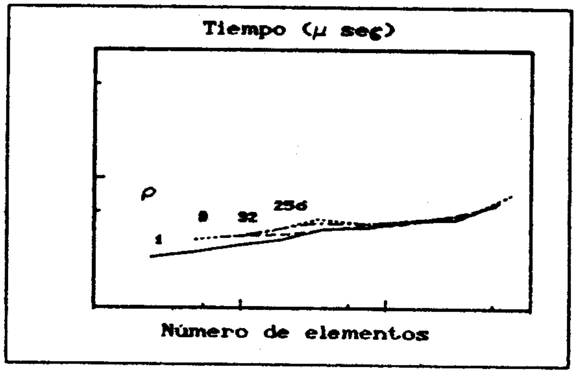



Figura 8: Tiempos del búsqueda del método 3

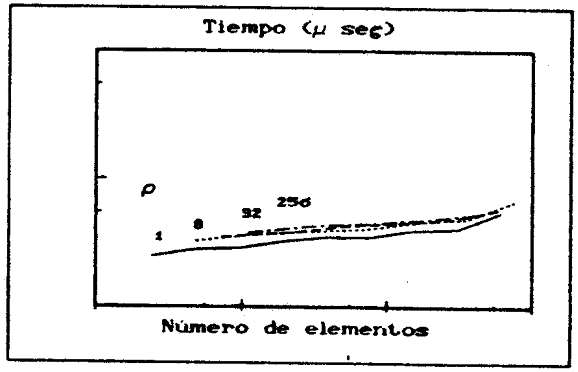



Figura 9: Tiempos de búsqueda del método 4

La memoria utilizada para el almacenamiento de la red auxiliar de los métodos 1 y 2 depende fuertemente del tamaño de celda elegido. En el método 4 la memoria no es tan sensible a este parámetro, ya que al disminuir el tamaño de las celdas "raiz" aumenta la cantidad de árboles pero disminuye el número de niveles de cada uno de ellos. En la figura 10 se muestra la memoria requerida por cada método para una relación de densificación $\rho$ = 32. Los parámetros a ajustar en cada método son los mismos que se utilizaron para obtener los tiempos en las figuras 6 a 9.

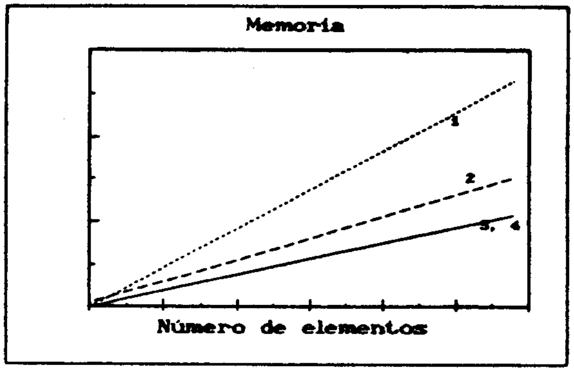



Figura 10: Memoria utilizada por cada método (palabras de 4 Bytes)

## CONCLUSIONES

Todos los métodos presentados permiten reducir en mayor o menor medida el orden del algoritmo de búsqueda.

Los dos primeros métodos (red regular homogénea y no-homogénea) son muy apropiados para redes uniformes, pero cuando la red presenta altas densificaciones puntuales su perfomance se degrada considerablemente. Además, la memoria requerida es muy sensible a la elección el tamaño de celda.

El método 3 no tiene parámetros a ajustar y se adapta con facilidad a la red original, pero para redes cuasi-uniformes de gran tamaño los métodos anteriores son muy superiores.

Por último, el cuarto método es el que mejor se adapta a todo tipo de redes. Los tiempos de búsqueda son muy poco sensibles a las densificaciones de la red e incluso al tamaño de la misma. El parámetro de ajuste (tamaño de las raíces de los árboles) no tiene una influencia significativa en la memoria requerida.

## REFERENCIAS

[1] G.C. Buscaglia, E.A. Dari, "Nascar: el método de características-Galerkin aplicado a las ecuaciones de Navier-Stokes", ENIEF '90 7mo. Encuentro Nacional de Investigadores y usuarios del método de Elementos Finitos. Mar del Plata (1990).

[2] V. Akman, W.R. Franklin, M. Kankanhalli, C. Narayanaswami, "Geometric Computing and Uniform Grid Technique", Computer Aided Design V.21 No.7 (1989).

[3] S. Pissanetzky, F. Basombrio, "Efficient Calculation of Numerical Values of a Polihedral Function", Int. Jou. of Numerical Methods in Engineering V.17 pp.231-237 (1981).

[4] R.A. Finkel, J.L. Bentley, "Quad-Trees: A Data Structure for Retrieval on Composite Keys", Acta Informatica V.4 pp.1-9 (1974).