CONGRESO EUROPEO DE FLUIDIZACION

VOLUMEN I

HIDRODINAMICA. APLICACIONES.
TRANSFERENCIA DE CALOR.

Editado por

A. Macías Machín y G. Winter.

Escuela Superior de Ingenieros Industriales de Las Palmas.

Universidad de Las Palmas de Gran Canaria.

Universitat Politècnica de Catalunya.

Universidad de Zaragoza.

I.N.P. Toulouse.

U. de Technologie de Compiègne.

APPLICATION OF AN ADAPTIVE FINITE ELEMENT METHOD TO SOLVE EVOLUTION HEAT TRANSFER PROBLEMS

A. Plaza[1], R. Montenegro[1] and L. Ferragut[2]

[1]Department of Mathematics, University of Las Palmas de Gran Canaria, Campus Universitario de Tafira, 35017-Las Palmas de Gran Canaria, Spain.
[2]Department of Applied Mathematics and Informatic Methods, Polytechnic University of Madrid, c / Rios Rosas 21, Madrid, Spain.

ABSTRACT

The adaptive refinement/derefinement processes of nested grids are very useful to solve time-dependent problems in which moving refinement areas are required. In this paper our derefinement algorithm is briefly explained. The efficiency of the algorithm is showed in a numerical example: a time-dependent convection-diffusion problem. The derefinement algorithm enables us to use the multigrid method in order to solve the equation system associated to the finite-element method. The additional computation time required by this algorithm amounts less than 1% of total execution time. The derefinement algorithm can be also used for local refining. Derefining after global refinement is thereby equivalent to a local refinement procedure, not requiring the use of error indicators. The algorithm promises to be very useful in more complex problems including non linearity where we need to adapt the mesh to a changing solution.

1. INTRODUCTION

The ability to automatically generate and adaptively control the discretizations in the numerical solution of partial differential equations over general domains is critical to the reliable application of numerical analysis techniques. In recent years there has been an increased recognition that careful consideration must be given to the interrelationship of these two areas in order to develop the most efficient procedures to adaptively solve the various classes of problems under consideration. We can say that a good discretization of the domain in which a problem in differential equations must be solved is, at lest, as important as the numerical formulation of that problem. See (Zienkiewicz and Zhu, 1991) and (Kasiyama and Okada, 1992).

There are many ways to implement an adaptive finite element method. One of the basic choice is: element subdivision or mesh regeneration? The answer is not too clear. See (Lewis et al., 1991). However, in the topic of nested meshes using local refinement, if the areas to be refined change with the time, the appearance of a large number of nodes creates a serious difficulty. Many of these nodes –though necessary in any past time– are useless at the present moment. So, in this case, it seems necessary to develop a derefinement algorithm able to remove dupe nodes, to get a good approximation of the numerical solution obtained in previous time step and to be combined with a local refinement.

We have used triangular elements with three nodes and a version of the 4-T algorithm of Rivara (Rivara, 1987) at the refining, (Ferragut, 1987). The particular election of the algorithm at refining is important because the derefinement algorithm *must be* understood as the inverse algorithm of the refinement one. We have chosen the 4-T algorithm of Rivara because it has suitable properties about the smoothness condition and non-degeneracy. Besides that, the number of possibilities that appear for an element and its sons at derefining is lesser than in other refinement algorithms.

With this combination (refinement and derefinement), we get families of sequences of nested meshes more flexible than those obtained by local refinement only and with the advantage that the number of equations does not increase so much during the whole evolution process: (Plaza et al., 1992), (Ferragut et al., 1993) and (Plaza, 1993). Besides, the fact of using nested grids enables us to use easily the multigrid method in order to solve the system of equations associated to the finite element method (Hackbush and Trottenburg, 1986). Supposing that the domain is defined by an irregular geometry, we can obtain the initial mesh by using an automatic grid generator and afterwards applying our adaptive process.

2. THE DEREFINEMENT PROCEDURE

2.1 Definitions and properties

Let $T = \{\tau_1 < \tau_2 < \ldots < \tau_n\}$ be a sequence of nested triangular grids and τ_j any triangulation of T. One node N of τ_j will be called a *proper node of* τ_j if it does not belong to any previous mesh. In other cases, N will be called an *inherited node in* τ_j. Similarly, the edges and elements are named at each level. If an edge is divided in two at refining, it is called the *father edge* of these two, and these are the *son edges* of the former. Similarly the *father elements* and *son elements* are defined. As we are using the 4-T algorithm of Rivara at refining, an element has four sons or less.

When an element is refined, some edges appear inside it. These edges are called *internal edges;* these edges are called *j-new edges* by Rivara (Rivara, 1989). On the boundary of the element some edges appear as well. Now these edges are called *external edges*. In this context, we have: i) any proper element of some triangulation either is inherited in the following triangulation or has its sons there; ii) if an element has no sons, it belongs to the finest mesh of the sequence of nested triangulations.

These definitions and properties are important because in contrast to the refinement algorithm in which only the last mesh created and the new one that is being created are involved, in the derefinement algorithm *all levels* of meshes are involved. The fundamental property of the derefinement algorithm is: iii) *only those elements without successors, i.e. elements that belong to the finest mesh, can be eliminated.*

2.2 Data Structure

The data structure allows an easy implementation of the refinement/derefinement algorithm in standard finite element codes and can be summarized as follows:

a) *Structure vectors*: *IMNODE*$(\cdot)$, *IMFACE*$(\cdot)$ and *IMELEM*$(\cdot)$ for the nodes, faces and elements, respectively. In *IMNODE* only the proper nodes of each level are kept because if one node belongs to a particular mesh, it belongs to the following meshes as well. *IMFACE* and *IMELEM* keep the global numbers of all faces and elements respectively of each level of *T*. With this data structure the implementation of the multigrid method is relatively simple.

b) *Genealogy vectors*: *IR*$[\cdot,\cdot]$ and *IXH*$[\cdot,\cdot]$. For each edge, *IR* reports the numbers of its son edges and its father edge. Similarly, for each element, *IXH* gives us the number of its son elements, its father element and the local number of its longest side.

c) *Derefinement vectors: NODES*$(\cdot)$, *NFACES*$(\cdot)$ and *NELES*$(\cdot)$. For each node, edge or element, these vectors give us the level at which it is proper and the sign of the vectors is used to control the derefinement procedure.

d) *Sack vectors: NNSAC*$(\cdot)$, *NFSAC*$(\cdot)$ and *NESAC*$(\cdot)$. In these vectors, the global number of nodes, edges and elements, that have been eliminated, are kept to be used in future refinements.

e) *Surrounding edge: IEX*$(\cdot)$. For each node *IEX* reports the number of the edge at which that node is at the middle point.

2.3 The Algorithm

Let $T = \{ \tau_1 < \tau_2 < ... < \tau_n \}$ be a sequence of nested triangular grids, where τ_1 represents the initial mesh and τ_n the finest mesh in the sequence. Our goal is to obtain another sequence after derefining T, said T'. This means that the new sequence can be written as follows: $T' = \{ \tau_1 < \tau_2' < ... < \tau_m' \}$ where $m \leq n$. The derefinement algorithm can be shortly described in this form:

INPUT: Sequence $T = \{ \tau_1 < \tau_2 < ... < \tau_n \}$

Loop in levels of T; for j = n to 2, do:

1. For each proper node of τ_j the derefinement condition is evaluated and the nodes and edges able to be eliminated are pointing out with the derefinement vectors.

2. Conformity of the arising new level j is assured.

3.a. If some proper node of τ_j stays then, new nodal connections are defined for the new level j, said τ_j^j. Genealogy vectors of τ_j^j and τ_{j-1} are modified.

3.b. In other case, the current level j is deleted in the structure vectors. Genealogy vectors of τ_{j-1} are modified.

4. The changes in the mesh are inherited to the following meshes. The structure vectors are compressed.

5. A new sequence of nested meshes T^j is obtained. This sequence is the new input in the next iteration of the loop on meshes. $T^j = \{ \tau_1 < \tau_2 < ... < \tau_{j-1} < \tau_j^j < ... < \tau_{n_j}^j \}$.

OUTPUT: Sequence $T' = \{ \tau_1 < \tau_2' < ... < \tau_m' \}$

We must take into account that only proper nodes are eligible in each mesh-level, and out of these, only those suitable to be canceled out are taken for evaluation. That is because *if N is a particular node that cannot be canceled and c is its surrounding edge*, then any node of the elements in which *c* is an edge cannot be canceled either.

Once the derefinement condition has been checked in all the eligible proper nodes of a particular mesh (inside the loop on levels), the conformity of the arising new level is assured maintaining some nodes that, otherwise, and concerning the derefinement condition, could have been canceled. In fact, if a node, said P, belongs to the longest edge of an element in which there is another node Q on any other edge that must remain, then the node P must remain too.

2.4 Derefinement indicator

A proper node may be removed if the absolute difference between the values in this node of the numerical solution and its corresponding interpolated function is less than a sufficiently small parameter $\varepsilon > 0$. That is, if u_h is the numerical solution for a given mesh and u_h^d is the interpolated function of u_h in the derefined mesh, we will get

$$\left\| u_h - u_h^d \right\|_\infty = \sup_x \left| u_h(x) - u_h^d(x) \right| < \varepsilon$$

Obviously, this derefinement indicator does not allow us to control the discretization error; in an adaptive algorithm this control is usually performed by an error indicator in the refinement process. It could be argued that the same error indicator should be used as a derefinement indicator. However when we use a time step integration scheme, a good approximation of the solution at time t_n must be kept to calculate the approximation at time $t_{n+1} = t_n + \Delta t_n$.

2.5 Conforming procedure

The procedure assuring the conformity of the mesh τ_j is summarized in the following. The concept of 1/2-non-conforming triangle is used, see (Rivara, 1987).

Input (τ_j, τ_{j-1}, Nodes)

While Conformity must be assured:

For each $t \in \tau_{j-1}$:

If t is 1/2-non-conforming:

Change the derefinement vector for the node P of its longest side.

Assure conformity.

End if.

End for.

Output (τ_j, τ_{j-1}, Nodes)

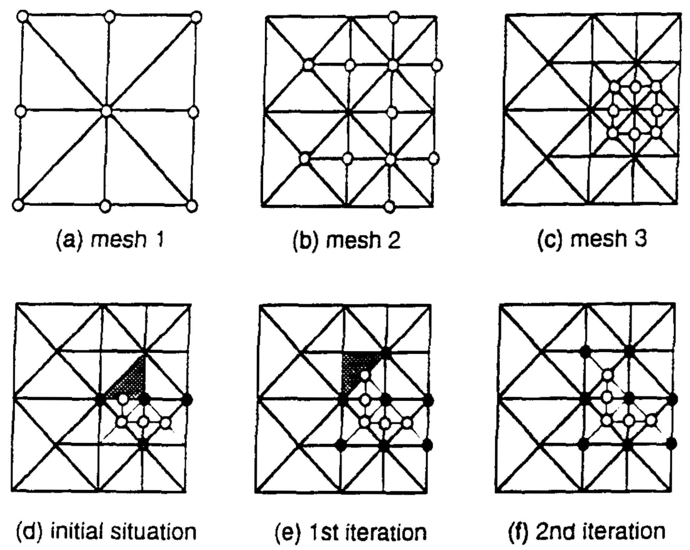


Figure 1.- Conformity of the arising new sequence.

An example of how the conformity of the arising new level is assured can be seen in Figure 1. There, the first line represents a sequence of three nested meshes in which the proper nodes are pointed out in white. The second line shows the evolution respect the derefinement vectors, when the third level is derefined. In the second line of the figure, the white nodes mean the proper nodes of the third level that will stay according to the derefinement condition (Figure 1-d) or according to the conformity of the mesh (in Figure 1-e and 1-f); the black nodes are inherited nodes in this level and these are marked because they are not suitable to be canceled out in order to assure the nestedness of the sequence of meshes. The shaded area is the non-conforming area at each loop of the iterative process of the conforming procedure applied to the third level.

Once T' is obtained, the number of equation associated to each degree of freedom/node must be redefined, maintaining the global number of the node and the previous numerical solution. At the same time the new number of equations of each level is calculated. This aspect is important to apply again the finite element solver.

3. CONVECTION-DIFFUSION PROBLEM

3.1 Semi-implicit formulation

The principal numerical aspects, presented in this section are studied in (Montenegro et al., 1989). We consider the convection-diffusion problem defined in a two-dimensional domain Ω, of boundary Γ:

$$\frac{\partial u}{\partial t} + \bar{v} \cdot \bar{\nabla} u - \bar{\nabla} \cdot \left(k \bar{\nabla} u \right) = f \tag{1}$$

where $u = u(\bar{x}, t)$ is the solution in a fluid element placed in $\bar{x} = x_1 \bar{i} + x_2 \bar{j}$ at time t, this solution may measure the temperature of the fluid element; $\bar{v} = \bar{v}(\bar{x})$ is the velocity of the medium carrying u, we consider the case in which the velocity field does not depend of time; $k = k(\bar{x}) > 0$ represents the diffusivity of u, we study the lineal model: $f = f(\bar{x}, t)$ are the external heat sources. We suppose an initial solution in Ω and boundary conditions on Γ such that the existence and uniqueness of solution is assured. Equation (1) can be written as:

$$\frac{du}{dt} - \bar{\nabla} \cdot \left(k \bar{\nabla} u \right) = f \tag{2}$$

Let be a fluid element at point $\underline{P}$, defined by $\underline{\bar{x}}$, at time t_n. After a time step Δt, this fluid element will be in position P, defined by $\bar{x}$, at time t_{n+1} ; such that $t_{n+1} = t_n + \Delta t$ and $\bar{x} = \underline{\bar{x}} + \Delta \bar{x}$. Using the following approximation

$$\frac{du}{dt} \approx \frac{u(\bar{x}, t_{n+1}) - u(\underline{\bar{x}}, t_n)}{\Delta t} = \frac{u^{n+1}(\bar{x}) - u^n(\underline{\bar{x}})}{\Delta t} \tag{3}$$

and applying an Euler implicit scheme to equation (2), we obtain:

$$u^{n+1}(\bar{x}) - \Delta t \, \bar{\nabla} \cdot \left[k(\bar{x}) \bar{\nabla} u^{n+1}(\bar{x}) \right] = \Delta t \, f^{n+1}(\bar{x}) + u^n(\underline{\bar{x}}) \tag{4}$$

In order to evaluate $u^n(\underline{\bar{x}})$ we try to write equations (4) in function of what happens at the point P at every time. For all i = 1,2:

$$\underline{x}_i = x_i(t_{n+1} - \Delta t) = x_i - v_i(\bar{x}) \Delta t + \frac{\Delta t^2}{2} \bar{v}(\bar{x}) \cdot \bar{\nabla} v_i(\bar{x}) + 0(\Delta t^3)$$

From this equation we can obtain $\Delta x_i = x_i - \underline{x}_i$ and write:

$$u^n(\underline{x}) = u^n(\bar{x}-\Delta\bar{x}) = u^n(\bar{x}) - \sum_{i=1}^{2}\Delta x_i \frac{\partial u^n(\bar{x})}{\partial x_i} + \frac{1}{2}\left[2\Delta x_1\Delta x_2 \frac{\partial^2 u^n(\bar{x})}{\partial x_1 \partial x_2} + \sum_{i=1}^{2}\Delta x_i^2 \frac{\partial^2 u^n(\bar{x})}{\partial x_i^2}\right] + 0(\|\Delta\bar{x}\|^3)$$

and then,

$$u^n(\underline{x}) = u^n(\bar{x}) - \Delta t\sum_{i=1}^{2} v_i(\bar{x})\frac{\partial u^n(\bar{x})}{\partial x_i} + \frac{\Delta t^2}{2}\sum_{i=1}^{2}\left[\bar{v}(\bar{x})\cdot\bar{\nabla}v_i(\bar{x})\right] + \frac{\Delta t^2}{2}\sum_{i=1}^{2}\sum_{j=1}^{2} v_i(\bar{x})v_j(\bar{x})\frac{\partial^2 u^n(\bar{x})}{\partial x_i \partial x_j} + 0(\Delta t^3)$$

Finally, if we introduce this last expression in (4) we get the following semi-implicit formulation that approximates the evolution convection-diffusion process, where all the terms are evaluated at the same point P, defined by $\bar{x}$:

$$u^{n+1} - \Delta t\,\bar{\nabla}\cdot\left[k\bar{\nabla}u^{n+1}\right] = \Delta t f^{n+1} + u^n - \Delta t\,\bar{v}\cdot\bar{\nabla}u^n + \frac{\Delta t^2}{2}\sum_i(\bar{v}\cdot\bar{\nabla}v_i)\frac{\partial u^n}{\partial x_i} + \frac{\Delta t^2}{2}\sum_{i,j} v_i v_j \frac{\partial^2 u^n}{\partial x_i \partial x_j}$$

Now, considering boundary conditions, it is easy to obtain the variational formulation and then apply the finite element method. We have used a consistent integration in all the terms of the final formulation.

3.2 Stability and consistency

Concerning the former formulation, we have used the principal stability results obtained in one and two dimensions by the Von Neumann method, see (Montenegro et al., 1989). About the consistency, on the lines of (Montenegro et al., 1989) and (Peraire et al., 1986), for the analyzed semi-implicit formulation, particularized in one dimension and with constant coefficients and without sources in the second member, it can be proved that it is globally second-order accurate.

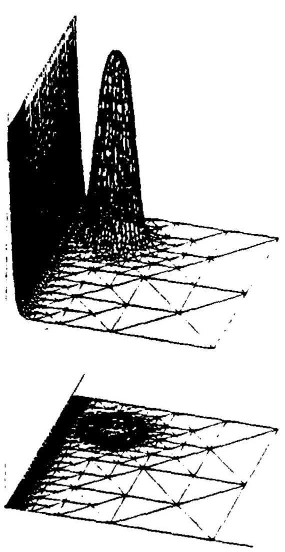
4. NUMERICAL RESULTS

4.1 A Convection-diffusion Problem

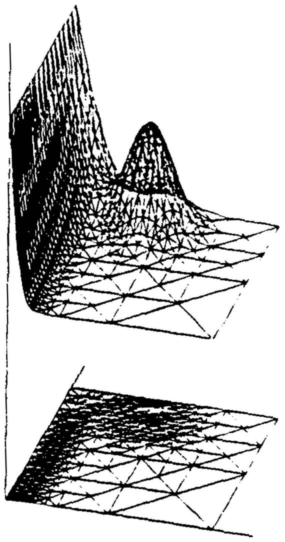
We consider the convection-diffusion linear problem (1) defined in a two-dimensional domain Ω, a unit square domain centered at the point $(0.5, 0.5)$, of boundary Γ. We suppose a rotating velocity field, with $v_1 = \omega x_1(1-x_1)(x_2-0.5)$ and $v_2 = \omega x_2(0.5-x_1)(1-x_2)$. In the present application we take $\omega = 1000$, $k = 1$ and $f = 0$. That is, a significant Peclet number about 125. On two opposite sides of the unit square domain we impose null Neumann conditions, on the other ones we suppose Dirichlet conditions, $u = 1$ and $u = 2$, respectively. The initial solution is a given function similar to the one represented in Figure 2(a) and it is captured automatically using the refinement/derefinement algorithm. This combination enables us to get a good approximation with a minimum number of nodes. In this example we have used the following refinement indicator, η_i, for an element Ω_i:

$$\eta_i = h_i|\nabla u_h|$$

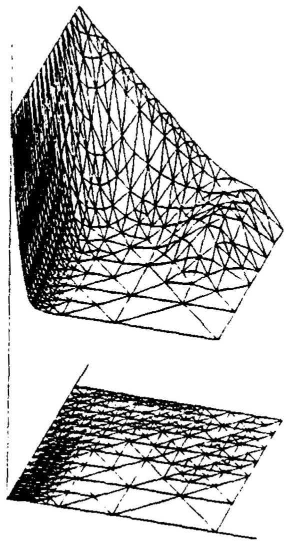
h_i being the diameter of Ω_i and u_h the linear numerical solution in the element.



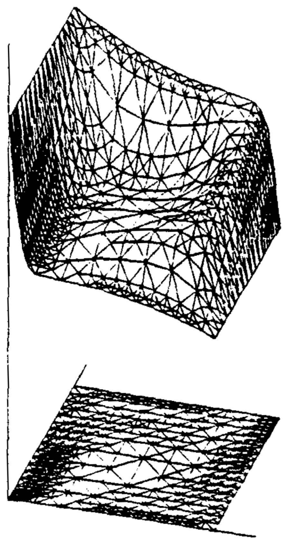
(a) t=0.00001, 1337 nodes



(b) t=0.00274, 617 nodes



(c) t=0.00815, 433 nodes



(d) t=0.05714, 595 nodes

Figure 2.- Meshes and solution for an evolutive heat transfer problem.

We know that $1 \leq u(\bar{x},t) \leq 2$ in all the evolution process, so, if we take a derefinement parameter $\varepsilon = 0.005$, we are sure that the error introduced by derefinement is less than 2.5‰ of the maximum value of the solution. The adaptive strategy was: only one refinement in ten time steps followed by the derefinement procedure. In order to evaluate the time increment, the stability condition, proposed by (Montenegro et al., 1989), has been used. In each time step, one multigrid iteration is enough to solve the system of equation associated to the finite element method. Figure 2 shows several meshes and solutions for different time steps. In order to obtain the stationary solution, approximately in Figure 2(d), 2000 time steps have been calculated.

REFERENCES

L. Ferragut, 'Una solucion al problema de la programación de métodos de elementos finitos autoadaptativos', Anales Ing. Mec. 5, 201-206 (1987).

L. Ferragut, R. Montenegro and A. Plaza, 'Efficient refinement/derefinement algorithm of nested meshes to solve evolution problems', *Comm. Num. Meth. Eng.*, in press.

W. Hackbush and V. Trottenburg (eds.), *Multigrid Methods II, Lectures Notes in Mathematics*, Springer-Verlag, Berlin, 1986.

R. W. Lewis, H.C. Huang, A.S. Usmani and J.T. Cross, 'Finite element analysis of heat transfer and flow problems using adaptive remeshing including application to solidification problems', *Int. J. Num. Meth. Eng.* **32**, 767-781 (1991).

R. Montenegro, G. Montero, G. Winter y L. Ferragut, 'Aplicación de métodos finitos adaptativos a problemas de convección-difusión en 2-D", *Revista Int. Métodos Numéricos para el Cálculo y Diseño en Ingeniería* **5**, 535-560 (1989).

J. Peraire, M. Vahdati, K. Morgan and O. Zienkiewicz, 'Adaptive remeshing for compressible flow computations', *J. Comp. Phys.*, **72**, 449-466 (1987).

J. Peraire, O. Zienkiewicz and K. Morgan, 'Shallow water problems: a general explicit formulation', *Int. J. Num. Meth. Eng.*, **22**, 547-574 (1986).

A. Plaza, R. Montenegro and L. Ferragut, 'An adaptive refinement/derefinement algorithm of structured grids for solving time-dependent problems', in Ch. Hirsch et al. (eds.), *Numerical Methods in Engineering*, Elsevier Science Publishers B.V., Amsterdam, 1992, pp. 225-232.

A. Plaza, *Derefinement Algorithms of nested meshes*, Doctoral Thesis, University of Las Palmas de Gran Canaria, 1993.

M.C. Rivara, 'A grid generator based on 4-triangles conforming. Mesh-refinement algorithms', *Int. J. Num. Meth. Eng.* **24**, 1343-1354 (1987).

M.C. Rivara, 'Selective refinement/derefinement algorithms for sequences nested triangulations', *Int. J. Num. Meth. Eng.* **28**, *2889-2906 (1989)*.

K. Kashiyama and T. Okada, 'Automatic mesh generation method for shallow water flow analysis', *Int. J. Num. Meth. Fluids* **15**, 1037-1057 (1992).

O.C. Zienkiewicz and J. Z. Zhu, 'Adaptivity and mesh generation', *Inter. J. Num. Meth. Eng.* **32**, 783-810 (1991).