A dynamical adaptive method based on local refinement and unrefinement for triangular finite-element meshes: preliminary results

Fayssal Benkhaldoun, Thierry Fernandez, Bernard Larrouturou, Pénélope Leyland

INRIA

UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1271

Programme 7
Calcul Scientifique,
Logiciels Numériques et Ingénierie Assistée

A DYNAMICAL ADAPTIVE METHOD BASED ON LOCAL REFINEMENT AND UNREFINEMENT FOR TRIANGULAR FINITE-ELEMENT MESHES : PRELIMINARY RESULTS

Fayssal BENKHALDOUN
Thierry FERNANDEZ
Bernard LARROUTUROU
Pénélope LEYLAND

Juillet 1990

RR_1271

A DYNAMICAL ADAPTIVE METHOD BASED ON LOCAL REFINEMENT AND UNREFINEMENT FOR TRIANGULAR FINITE-ELEMENT MESHES: PRELIMINARY RESULTS

Fayssal BENKHALDOUN, Thierry FERNANDEZ
Bernard LARROUTUROU, Pénélope LEYLAND

ABSTRACT: We present an adaptive refinement-derefinement method developed for solving stiff unsteady problems on a time-dependent unstructured finite-element triangulation. This procedure is applied to a stiff non linear reaction-diffusion model describing the propagation of a premixed laminar flame.

UNE METHODE DE RAFFINEMENT ADAPTATIF DYNAMIQUE POUR DES TRIANGULATIONS EN ELEMENTS FINIS: RESULTATS PRELIMINAIRES

Fayssal BENKHALDOUN, Thierry FERNANDEZ
Bernard LARROUTUROU, Pénélope LEYLAND

RESUME: Nous présentons une méthode de raffinement-déraffinement adaptatif dynamique pour la résolution de problèmes raides instationnaires sur des maillages en éléments finis triangulaires non structurés. A titre d'exemple, cette méthode est utilisée pour la résolution d'un modèle de réaction-diffusion décrivant la propagation d'une flamme prémélangée.

A DYNAMICAL ADAPTIVE METHOD BASED ON LOCAL REFINEMENT AND UNREFINEMENT FOR TRIANGULAR FINITE-ELEMENT MESHES: PRELIMINARY RESULTS

F. BENKHALDOUN[1], T. FERNANDEZ[2], B. LARROUTUROU[3], P. LEYLAND[4]

INRIA, Sophia-Antipolis, 06560 VALBONNE, FRANCE

1 INTRODUCTION

Local adapted mesh refinement algorithms for the solution of steady problems on unstructured meshes greatly improve the resolution by enhancing the accuracy of the approximation and reducing the computer storage, thus speeding up the convergence. Critical areas are refined locally in such a way that discontinuities, high gradients and boundary layers have a sufficient number of mesh points to resolve their structure, whereas zones of uniform activity are solved with a coarser distribution of the mesh points. The capture of these critical areas to be refined locally depend either on a physical criterion or a local error estimate criterion, or both. The procedure is static: starting from an initial mesh, the next grid is obtained by subdivision of those elements which are premarked according to different tolerance levels for the error criterion evaluated using a quasi-converged solution on this initial mesh.

(1) Present address: INSA de ROUEN, Place E. Blondel, 76131 MONT SAINT AIGNAN Cedex, FRANCE.
(2) Present address: Simulog, 3, avenue du Centre, 78182 SAINT QUENTIN YVELINES Cedex.
(3) Present address: CERMICS, INRIA, Sophia-Antipolis, 06560 VALBONNE, FRANCE.
(4) Present address: DMA-EPFL, Ecublens, CH-1015 LAUSANNE, SWITZERLAND.

Conformality (and sometimes regularity) requires additional divisions, a new solution is then calculated on the new mesh, and the adaptive meshing can be reapplied, up of optimal convergence of the solution and of the mesh.

Multigrid methods based on a series of unstructured grids constructed in this way have proved to be performant (see [1], [2], [5], [6], [7], [21], [28]). In this context, the successive refinement history needs be stored within the data structure in order to optimize projections, prolongations and corrections between the grid levels.

Using the same strategy for solving really *unsteady* phenomena requires *unsteady* local refinement. Adaptive methods of another type, which consist in moving the mesh nodes, can also be used (see for instance [19], [30]), but they are less easy to use in the contexts of unstructured finite-element grids and complex geometries. In this paper we present a dynamic adaptive procedure based on multi-level local refinement and unrefinement : the accuracy is obtained by locally refining the mesh in regions of particular interest, and the efficiency is obtained by the successive unrefinements whenever the regions of particular interest move out of the refined zone.

Related adaptive refinement-unrefinement procedures have also been developed by other authors. The tree data structure for the multilevel refined finite-element triangulation which we are going to use, with a "father-son-grandson" hierarchy, was used in particular by Bank et al [8] and Rivara [29]. The first step of all such adaptive algorithms consists in establishing a list $\mathcal{S}$ of those triangles which need to be refined, according to the local values of some criterion. The method of Rivara [29] then consists in dividing all triangles of $\mathcal{S}$ by joining the mid-point of the longest side to the opposite vertex, and so on up to conformality. On the other hand, Bank [6], [8] uses uniform division into four subtriangles within $\mathcal{S}$: conformality is only partially respected, and a hierarchical tree data structure is created. More recently, several authors [9], [22], [26] have developped refinement-unrefinement algorithms for transient and steady-state problems in a similar philosophy to ours, using a hierarchical data structure.

In all these works, the determination of the elements to be refined or unrefined is based on an error criterion, which is either directly related to the mathematical problem under consideration using local error estimates (which can be a priori estimates (see e.g. Babuska et al. [3], [4]) or a posteriori estimates (see e.g. Erikkson, Johnson et al. [16], [17], [18], [20]), or using a physical criterion detecting insufficient accuracy on critical quanti-

ties: entropy for shock waves, normal gradients for boundary layers, pressure gradients for weak shocks etc... (see e.g. Pouletty [27]). For instance, Löhner et al. [9], [22] use a combination of a linear finite-element interpolation estimate, as can be derived for a linear elliptic problem, and of a criterion based on some key variable (such as the density gradient, for example, in the case of multiple transient shocks). The error estimate is modulated by a noise parameter, which is related to different degrees of tolerance for the error; a protective layer is also introduced in order to smooth out large discrepancies between the refined and unrefined zones. This last point has a particular importance for compressible flow calculations on unstructured meshes, since a certain uniformity of the mesh is required to avoid disparities of adjacent flux calculations.

The former error estimates, which come from the equations and some finite-element analysis, can only be obtained for model linear equations (see [16], [18], [20]), or for particular singularities (see [24], [25]). If these error estimates are available, for a steady problem, local errors must be minimized and redistributed uniformly throughout the mesh, so that $\|u - u^h\| < \delta$, where the norm $\|.\|$ depends on the approximation space used to calculate the discrete solution u^h, and where u denotes the exact solution and δ the error tolerance level. Thus, the mesh is said to be optimal in the sense that it corresponds to a minimal number of degrees of freedom compatible with a given accuracy for a converged solution.

For time-dependent problems, temporal truncation errors must also be considered within the criterion. Error estimates for parabolic equations have been studied in particular by Ushijima [31] (which we shall closely follow for our criterion), Nochetto et al. [24] and Erikkson et al. [17], [18]. The latter authors consider discontinuous Galerkin space-time discretization, and obtain a posteriori error estimates for a linear model parabolic problem. But for non-linear or hyperbolic problems only heuristic arguments can decide upon the optimal criterion to employ. In the example presented here - a non-linear stiff parabolic system - the adaption decision is based on local a posteriori error estimates which directly come from the governing equations describing the phenomena.

This report is divided into two main parts. The first one describes the main steps of ADAPT, our refinement-unrefinement algorithm. In the second part, we apply our refinement-definement procedure to an unsteady flame propagation problem ; the difficulty here is caused by the problem stiffness,

which essentially appears through the thinness of the flame and of the reaction zone. In particular, in contrast with hydrodynamical discontinuities, a correct simulation of the flame propagation requires to accurately compute the inner structure of the thin flame (see [15], [19]).

2 THE ADAPTIVE PROCEDURE

In this section, we describe in detail the dynamic adaptive algorithm itself : we will successively present the basic principles underlying this procedure, its data structure, the general organization of the algorithm, and lastly describe the main steps in the procedure. We consider here the adaptive procedure in itself, and postpone till the next section the question of how to use this procedure for the solution of an actual unsteady physical problem.

The procedure ADAPT creates an adaptive conformal finite-element triangulation, using a multi-level hierarchy and local mesh refinement. Let us first briefly introduce the concepts of conformal mesh refinement in a finite-element triangulation and of multi-level refinement.

2.1 Conformal mesh refinement

We use here the word "conformal" in its usual sense in the finite-element context: for instance, a triangulation of a domain in $\mathbb{R}^2$ is said to be conformal if, for any pair $\{T_i, T_j\}$ of triangles with $T_i \neq T_j$, the set $T_i \cap T_j$ is either empty, or reduced to a common vertex of T_i and T_j, or reduced to a common edge of T_i and T_j. The advantage of defining a refinement algorithm that always generates conformal meshes is of course that any finite-element solver can directly be applied to the refined mesh.

As we already said before, local mesh refinement algorithms have been used by many authors (see the references quoted above). In several space dimensions, these algorithms are more or less easy to handle in practice, depending on the structure of the refined mesh. In this context unstructured finite-element simplicial meshes (that is, unstructured triangulations in two dimensions and unstructured tetrahedrizations in three dimensions) have a

definite advantage over structured or unstructured finite-volume or finite-difference grids based on quadrilaterals or hexahedra.

Indeed, let us consider the quadrilateral mesh of Figure 1.

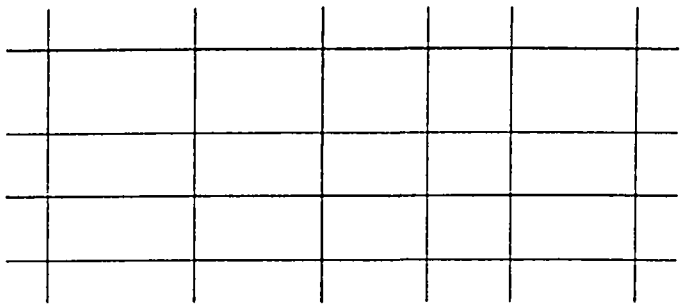

Figure 1: Structured quadrilateral mesh.

If we decide to divide one cell into four sub-cells, we obtain either the mesh of Figure 2 (a), where only one cell has been divided, but where some special approximation has to be used for the four new nodes, in the neighbourhood of which the mesh structure is singular (the mesh of Figure 2 (a) is not conformal), or the mesh of Figure 2 (b), where the refinement decision has been propagated over a full row and a full column of the initial grid, in order to keep the tensor-product structure.

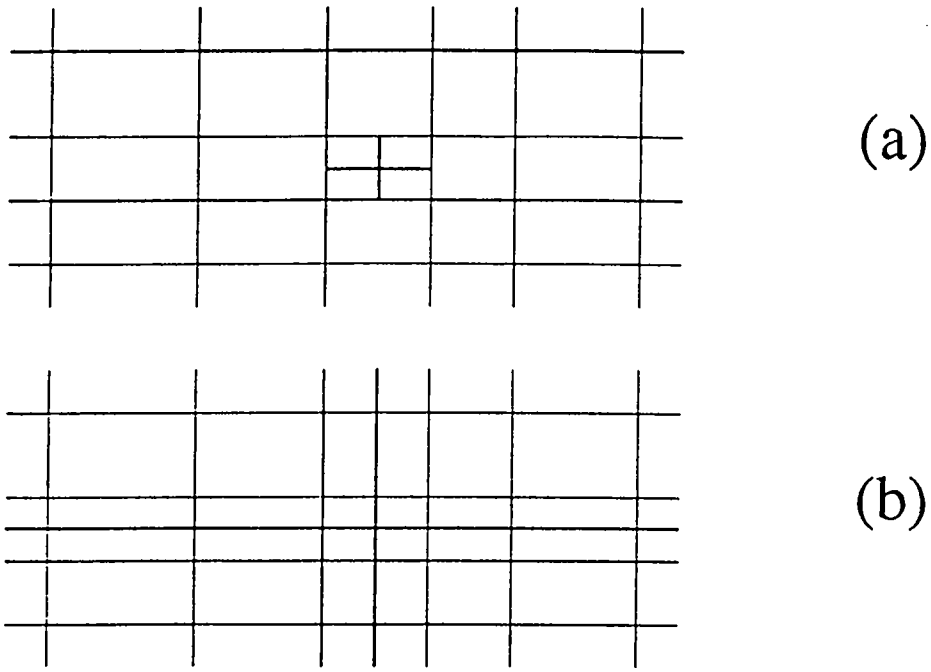

Figure 2: Local refinement of a structured quadrilateral mesh.

On the opposite, keeping a conformal mesh is an easy and cheap task if one starts with a finite-element *triangulation*. Following Bank [6], [8], we consider here as the basic refinement step the division of a triangle into four subtriangles using the middle points of the edges:

Figure 3: Basic refinement step: division into four subtriangles.

Then, a conformal triangulation can be constructed by simply examining the elements of which at least one edge is cut: once the decisions of dividing some elements into four sub-elements have been made, only a small number of additional divisions are needed to obtain a conformal triangulation. This is illustrated in Figure 4, where triangle T is not *a priori* divided into four triangles, but has either none, one, two or three divided neighbours; a conformal triangulation is obtained by dividing T into two, three of four sub-elements.

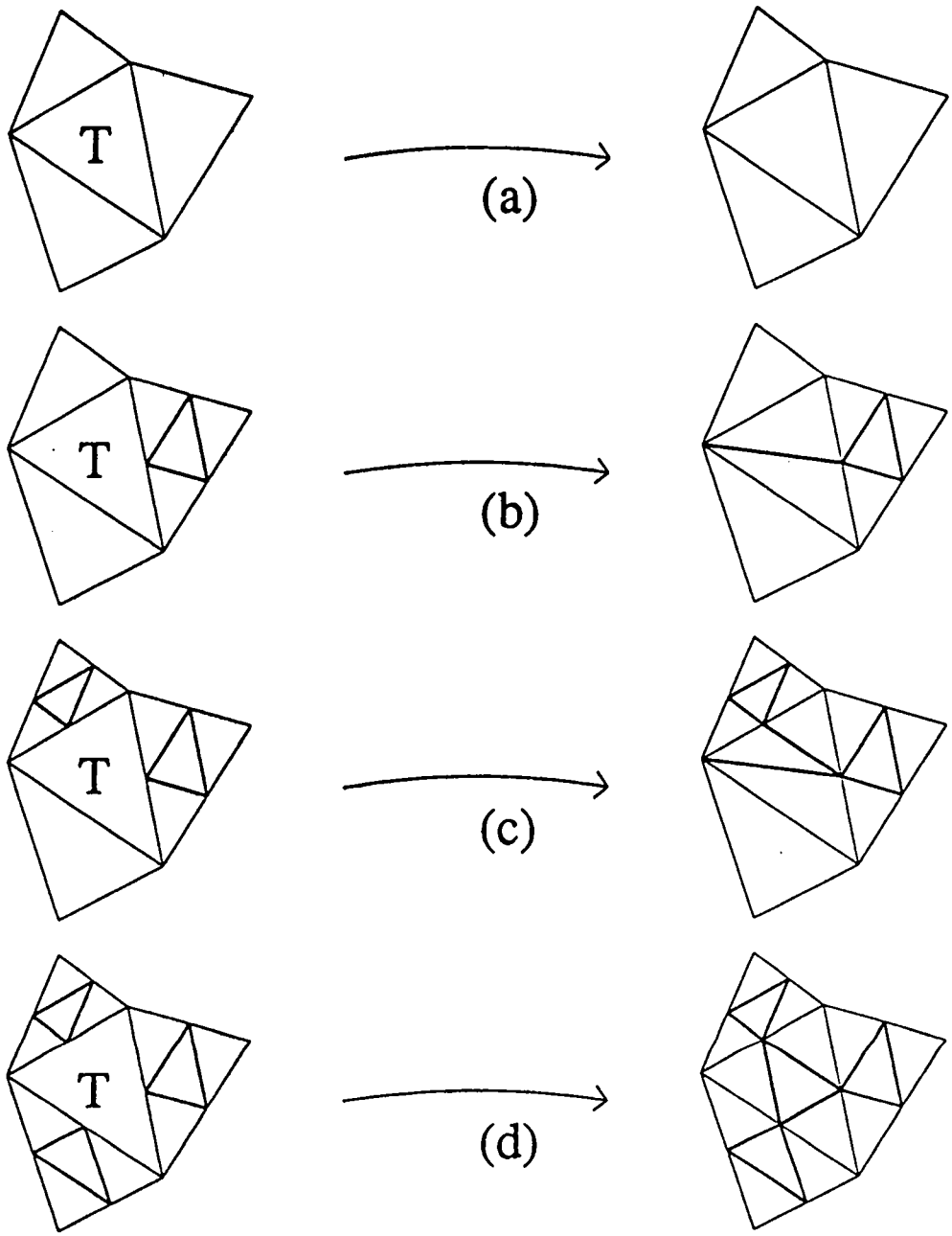

Figure 4: Additional divisions required for conformality: the triangle T is not *a priori* divided, and has (a) no divided neighbour, (b) one divided neighbour, (c) two divided neighbours, (d) three divided neighbours.

Thus several rules must be introduced within the local refinement algorithm in order to maintain conformality and mesh optimality. Optimality refers here to the minimization problem behind the error analysis, and to the regularity constraints which are classical in finite-element analysis (involving the angles and the diameter of the elements; see e.g. Ciarlet [13]). In particular, we will not exactly apply the refinement rules on Figure 4: in case (*c*) of Figure 4, we do not divide triangle T into three subtriangles, which would have too obtuse angles. Instead, we decide to divide T into four elements, and Figure 4 (c) becomes:

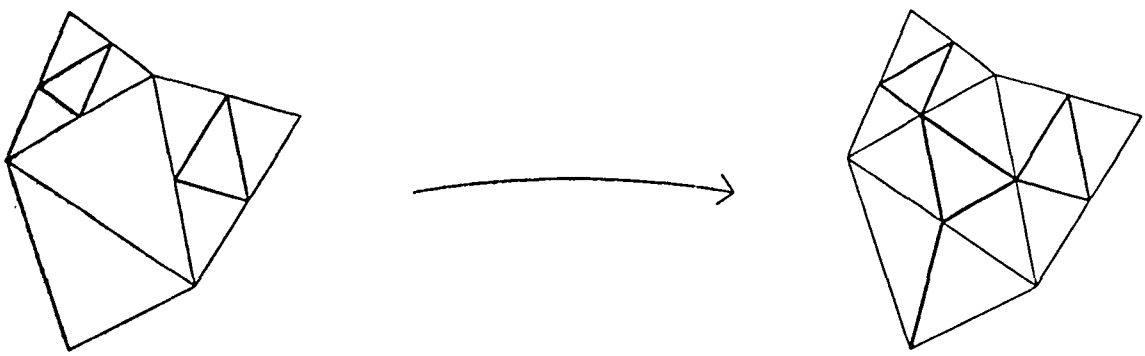

Figure 5: New rule for the additional divisions required for conformality (case (c) of Figure 4).

Thus, with this modification, all original triangles are either undivided, or divided into two subtriangles, or divided into four subtriangles.

2.2 The multi-level hierarchy

When applied several times in a row, this refinement procedure creates a multi-level hierarchy of the triangles. Indeed, we start with an original (coarse) triangulation. Then, using one time the refinement procedure, we create sub-triangles; we will say that these subtriangles are the sons of the original elements. Each original element (we will also say: each macro-element) has either no son, or two sons, or four sons. The macro-elements are also called "triangles of level 0", while their sons are "triangles of level

1".

If we again use the refinement procedure, we create "triangles of level 2", which are the sons of the triangles of level 1 (and the "grand-sons" of the macro-elements):

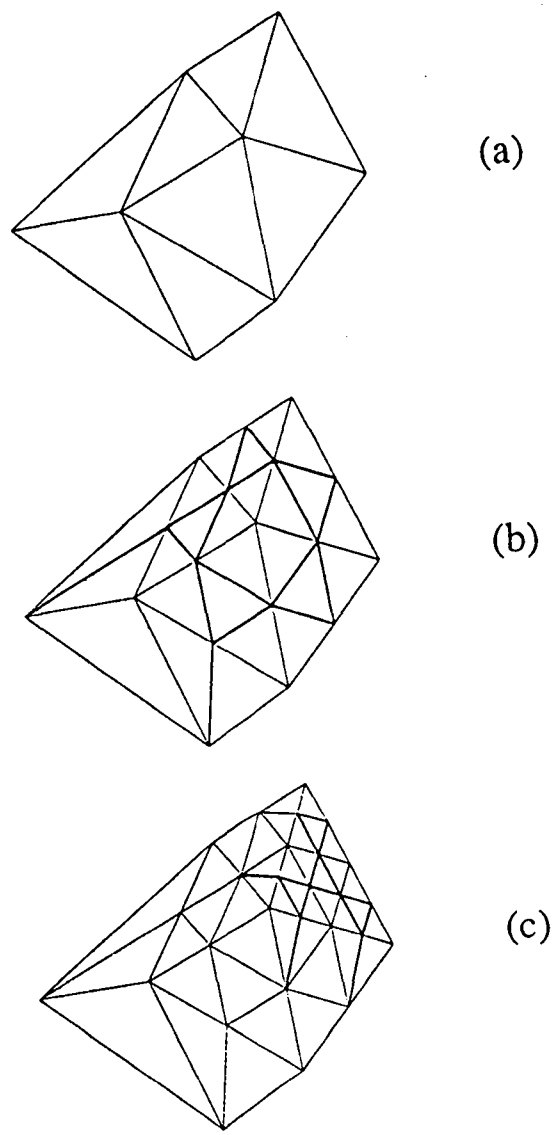

Figure 6: Multi-level refinement: the successively generated meshes (a) of level 0, (b) of level 1, (c) of level 2.

2.3 The basic principles

ADAPT is a dynamic multi-level refinement procedure. The word "dynamic" here means that ADAPT is aimed at constructing an adaptive mesh which dynamically follows the *unsteady* solution of a physical problem. At each time when the procedure ADAPT is called in such a time-dependent calculation, it constructs a mesh which is refined where refinement is needed *at that particular time*; thus, if the case may happen, ADAPT has to unrefine the grid in regions where fine cells are *no longer* needed. Therefore, ADAPT has to perform both refinements and unrefinements.

All grids constructed by ADAPT are based on the "grid of level 0", that is on the grid made of the macro-elements. This means that a macro-element can never be suppressed in an unrefinement phase; however its filiation may change: at any time level t^n, each element in the current triangulation $\mathcal{T}^n$ is either an undivided macro-element, or a subtriangle of level $m > 0$ obtained by dividing m times a macro-element.

In the procedure ADAPT, the maximal number of refinement levels is a priori prescribed; we call this number $m_{\max}$ ($m_{\max}$ can in principle be any positive integer). Then, ADAPT consists of a big loop over the number $m_{\max}$ of refinement levels : that is, starting from the macro-elements, one first takes care of the triangles of level 1 (one creates these triangles where needed, and deletes them where they have become unnecessary), then of the triangles of level 2, etc... and lastly of the triangles of level $m_{\max}$.

The refinement and unrefinement decisions are made, during this loop, in view of the values stored in an integer array called `IADIV`. This array has to be filled *for all macro-elements* before the procedure ADAPT is called. For a macro-element T_j, `IADIV` (T_j) can take any integer value m between 0 and $m_{\max}$: the assignment `IADIV` $(T_j) = m$ means that the macro-element T_j has to be divided into 4^m sub-triangles, or equivallently that T_j has to be divided into 4 sons, and that each of these sons (of level 1) has to be divided into 4^{m-1} sub-triangles. During the big loop in ADAPT, after having created the triangles of level n, ADAPT fills the array `IADIV` for these new triangles before turning to level $n+1$: if T_k (of level $n-1$) is the father of T_i (of level n) and if `IADIV` $(T_k) > 0$, then one sets `IADIV` (T_i) = `IADIV` $(T_k) - 1$. Notice here that all refinement and unrefinement decisions only depend on the values given initially in the array `IADIV` for the macro-elements.

Figure 7 gives the values of IADIV for the example of Figure 6.

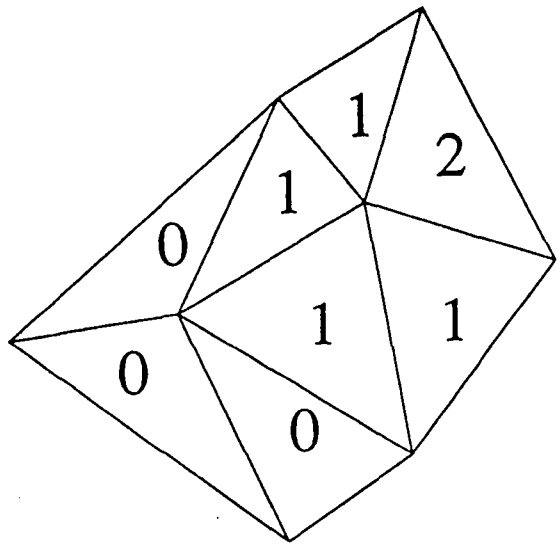

Figure 7: Multi-level refinement: the values of IADIV for the example of Figure 6.

Before turning to the precise description of the algorithm, let us add that this organization of the procedure as a big loop over the number $m_{\max}$ of refinement levels allows us to store the hierarchy of the triangles. This is necessary for unrefinement phases, since the whole filiation of a triangle has to be deleted when this triangle itself is deleted. Also, let us emphasize that ADAPT does not really reconstruct the whole refined mesh at each time it is called : for the sake of efficiency, we try to construct or delete only what need be changed with respect to the previous mesh. In other words, instead of reconstructing everything, starting from the macro-elements, we keep for the new triangulation all triangles of the old triangulation which remain valid; thus, almost nothing is done in ADAPT if the old and new triangulations are almost identical.

2.4 The algorithm

Let us now more precisely describe the different steps in the procedure ADAPT. Consider a time level t^n where the procedure ADAPT is called ; we denote $\mathcal{T}^{n-1}$ the old triangulation, and $\mathcal{T}^n$ the triangulation which ADAPT is going to construct at time t^n. Let us emphasize again here that all refinement and unrefinement decisions made in the procedure only depend on the values of IADIV (T_j) for all triangles T_j of level 0.

ADAPT begins with a big loop over the number $m_{\max}$ of refinement levels. For each level m, with $1 \leq m \leq m_{\max}$:

(a) ADAPT first uses the values IADIV (T_j) for all triangles T_j of level $m-1$ and decides to cut into two equal parts the edges of those triangles T_j of level $m-1$ such that IADIV $(T_j) > 0$. In order to obtain a triangulation which is not too distorted, ADAPT also decides to divide into two equal parts some additional edges (if two edges of a triangle have to be cut, then the third edge is also cut; see Figures 4 and 5).

(b) ADAPT then eliminates all triangles of level m of the old triangulation $\mathcal{T}^{n-1}$ which are no more valid for the new triangulation, together with their filiation: if a triangle T_k of level $m-1$ was divided in the old triangulation $\mathcal{T}^{n-1}$ and is not divided any longer, then the whole filiation of T_k is eliminated. On the opposite, ADAPT keeps all triangles of level m of the old triangulation $\mathcal{T}^{n-1}$ which remain valid for the new triangulation. In this step, the triangles are renumbered and internal flags are used to allow a dynamic storage saving for the global numbering of the triangles.

(c) Then, ADAPT actually generates the vertices at the midpoint of the edges which have to be cut from step (a), and creates the new triangles of level m.

(d) Lastly, ADAPT evaluates the values of IADIV for the triangles of level m: if IADIV $(T_j) = \hat{m} > 0$ for a triangle of level $m-1$, one sets IADIV $(T_k) = \hat{m} - 1$ for all sons T_k of T_j.

After this loop over the level index m, ADAPT globally renumbers the nodes of the triangulation (some nodes of the old triangulation may need to

be cancelled, and this could not be done inside the loop). The new triangulation $\mathcal{T}^n$ is then ready, and is automatically a conformal triangulation.

The reader is refferred to Maman [23] for more details about the implementation and the different steps in the procedure ADAPT.

2.5 Testing the procedure

Before we use the adaptive procedure ADAPT for the solution of an actual unsteady physical problem in the next section, we simply test it now on a very simple model problem: the mesh is adapted in order to follow a circular wave propagating in a square box.

More precisely, we consider a given positive even function $f \in \mathcal{C}(I\!R, I\!R)$, and a given sequence $t_0 = 0 < t_1 < t_2 \ldots < t_n \ldots$ of the successive time levels where the (initially uniform) mesh will be refined. At time t_n, for a macro-element T_k of vertices $S_{k_l} = (x_{k_l}, y_{k_l})$ $(l = 1, 2, 3)$, we set $\texttt{IADIV}(T_k) = 2$ if $\max\limits_{1 \le l \le 3} f(\sqrt{x_{k_l}^2 + y_{k_l}^2} - t^n)$ is above a given value, and $\texttt{IADIV}(T_k) = 0$ otherwise. We emphasize here that this experiment simply tests the procedure ADAPT itself, which is not coupled here to any code solving partial differential equations governing a physical phenomenon.

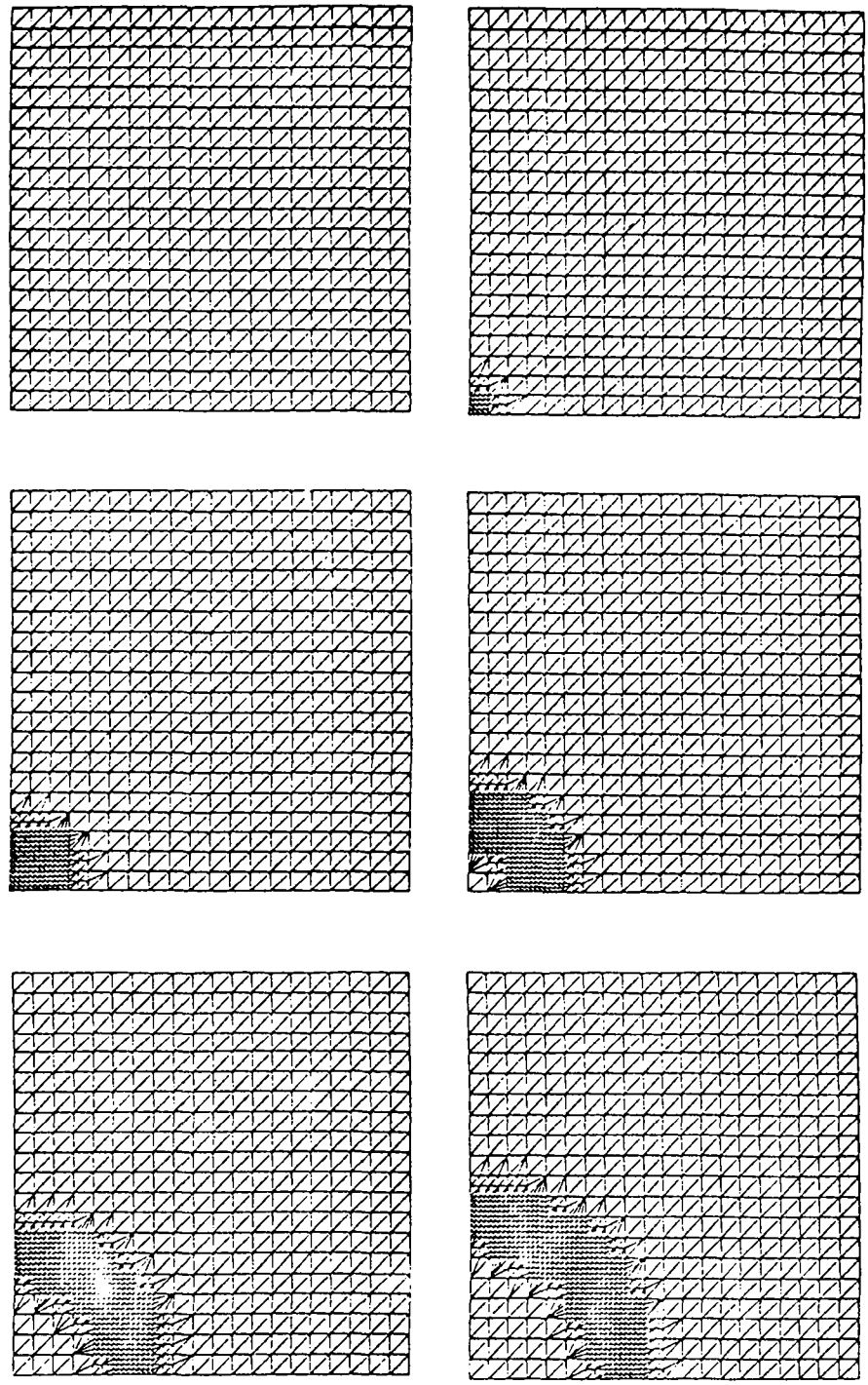

Figure 8: Testing the procedure ADAPT on a propagating circular wave.

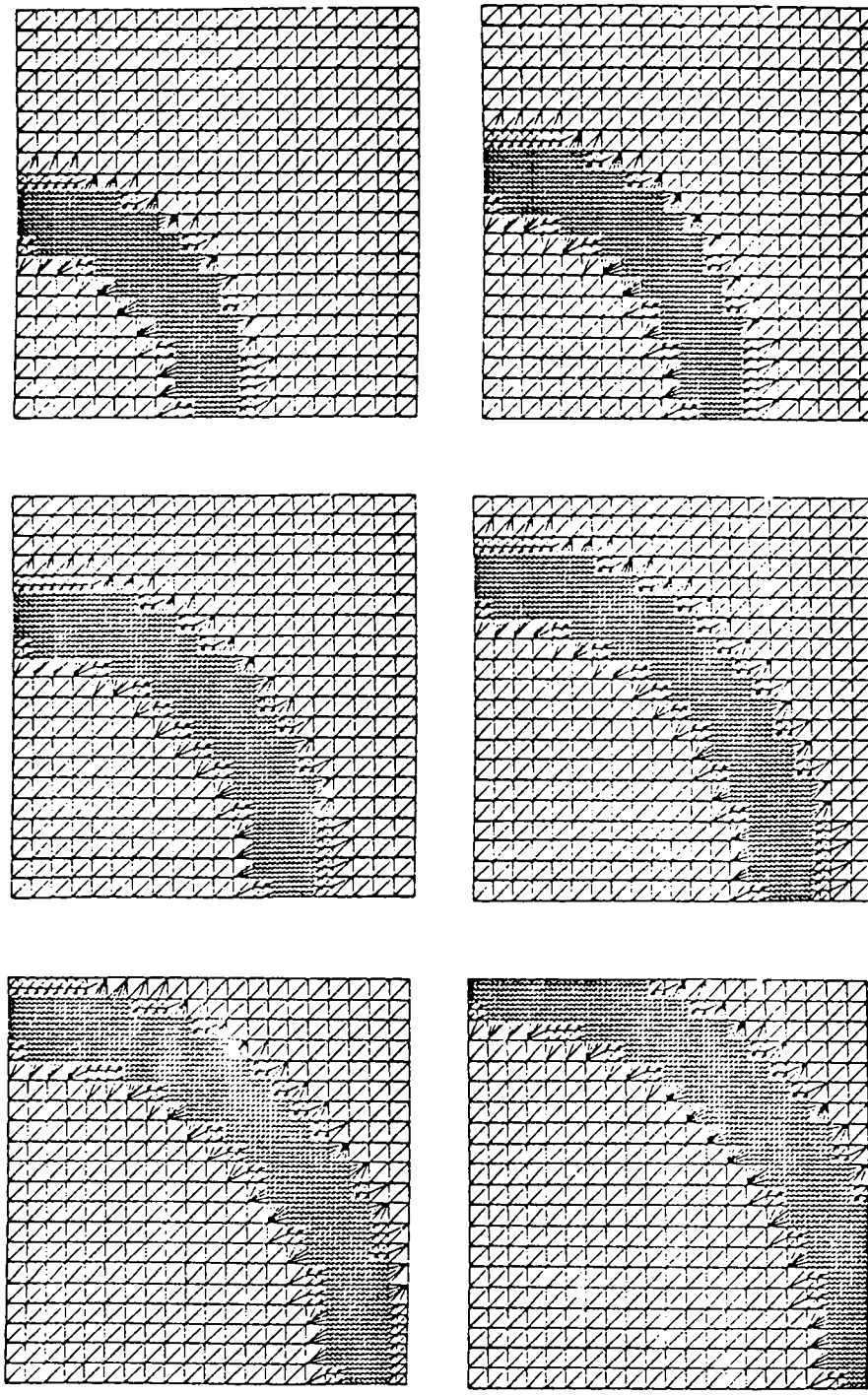

Figure 9: Testing the procedure ADAPT on a propagating circular wave.

One observes in Figures 8 and 9 that all generated meshes are actually conformal. A careful examination also shows that triangles with very small angles are created in the transition region between the refined zone and the coarse mesh region. In ADAPT, the user can avoid the creation of such triangles by the following procedure: for each pair of neighbour triangles which should be divided into two subtriangles each, one can choose to divide each triangle into four sons. These optional refinement decisions are made at the end of Step (a) in the loop of Section 2.4.

3 UNSTEADY FLAME PROPAGATION

In this section, we apply the dynamic refinement procedure ADAPT to the solution of an unsteady flame propagation problem.

3.1 The physical problem

We consider the simplified thermo-diffusive model describing laminar premixed flame propagation with one-step chemistry, in which the hydrodynamic effects are neglected, but which retains the stiffness of the phenomenon. Thus, an efficient mesh adaption will be necessary to solve the small length scales inside the flame. The model is a stiff system of two coupled non linear reaction-diffusion equations of the form:

$$\begin{cases} \dfrac{\partial \Theta}{\partial t} = D_\Theta \Delta \Theta + Q\Omega(Y,\Theta) \ , \\ \\ \dfrac{\partial Y}{\partial t} = D_Y \Delta Y - \Omega(Y,\Theta) \ , \end{cases} \tag{1}$$

with initial and boundary conditions. In these equations, Θ is the gas temperature, Y is the mass fraction of reactant, D_Θ and D_Y are positive parameters proportional to the thermal and molecular diffusion coefficients respectively, the positive coefficient Q is related to the heat released by the chemical reaction, and $\Omega(Y,\Theta)$ is the reaction rate, given by:

$$\Omega(Y,\Theta) = K_0 Y \exp\left(-\frac{\Theta_0}{\Theta}\right) \ , \tag{2}$$

K_0 and Θ_0 being positive constants. We refer to e.g. [10] for more details about the model.

The main difficulties in simulating the propagating flame described by these equations are related to the spatial stiffness of this phenomenon (i.e. the existence of disparate characteristic length scales), and will therefore concern the evaluation of an appropriate criterion guiding the refinement and derefinement decisions, and more generally the appropriate choice of all parameters involved in the adaption procedure.

4 Numerical method

For the sake of simplicity, we use for the solution of system (1) classical P1 Lagrangian finite elements, with a mass-lumped approximation and an explicit time-integration. More precisely, assuming that a conformal finite-element triangulation $\mathcal{T}^n$ is constructed in the computational domain $\mathcal{D}$ at time t^n, we consider the usual piecewise constant basis functions ϕ_i associated to this triangulation (for all node S_i of $\mathcal{T}^n$), and write the discrete integration in the interval $[t^n, t^{n+1}]$ in the form:

$$\frac{U_i^{n+1} - U_i^n}{\Delta t^n} \int_{\mathcal{D}} \phi_i = D_U \int_{\mathcal{D}} \vec{\nabla}\phi_i \left(\sum_j U_j^n \vec{\nabla}\phi_j \right) + R_U \Omega(Y_i^n, \Theta_i^n) \int_{\mathcal{D}} \phi_i \ , \quad (3)$$

for all interior node S_i of the triangulation $\mathcal{T}_h^n$ (we do not detail here neither the boundary conditions nor their numerical treatment, which is straightforward). In (3), the variable U is either Θ or Y, with the notations $R_\Theta = Q$, $R_Y = -1$. The mass-lumped integration of the time derivative allows us to avoid the storage of the complete non diagonal mass matrix and to use a fully explicit time integration. More precisely, we use a fourth-order Runge Kutta scheme, where the time step $\Delta t^n = t^{n+1} - t^n$ is taken to be:

$$\Delta t^n = \min(\Delta t_d^n, \Delta t_r^n) \ , \quad (4)$$

where Δt_d^n is the largest explicit time step for the diffusive terms:

$$\Delta t_d^n \leq \frac{(\Delta h^n)^2}{2 \max(D_\Theta, D_Y)} \ , \quad (5)$$

(Δh^n being the smallest mesh size in the triangulation $\mathcal{T}^n$), and where Δt_r^n is the largest stable time step for the reactive term of the mass fraction equation:

$$\Delta t_r^n \leq \left[K_0 \max_i \exp\left(-\frac{\Theta_0}{\Theta_i^n}\right)\right]^{-1} . \tag{6}$$

4.1 The refinement criterion

In order to solve problem (1) on a computational mesh which is dynamically refined using the procedure ADAPT, we need *(i)* to decide when the procedure ADAPT is called during the calculation, and *(ii)* to fill the array `IADIV` for all macro-elements, at each time ADAPT is called. As briefly discussed in the introduction, the latter is done using a refinement criterion, as we now describe.

Consider a time level t^n and assume that the solution (Θ^n, Y^n) has been computed on the triangulation $\mathcal{T}^{n-1}$. Then the criterion is determined using the current solution (Θ^n, Y^n) as follows: for each macro-element T_k, we first evaluate:

$$\begin{aligned}\zeta^n(T_k) = \; & \frac{\mathcal{V}}{\mu_x}\|\vec{\nabla}\Theta^n|_{T_k}\|_2 + \frac{1}{h_k}|D_\Theta[\vec{\nabla}\Theta^n]_k| + Q\|\Omega^n\|_{L^\infty(T_k)} \\ & + \frac{\mathcal{V}}{\mu_y}\|\vec{\nabla}Y^n|_{T_k}\|_2 + \frac{1}{h_k}|D_Y[\vec{\nabla}Y^n]_k| + \|\Omega^n\|_{L^\infty(T_k)} ,\end{aligned} \tag{7}$$

and set:

$$\eta^n(T_k) = area(T_k)\zeta^n(T_k) ; \tag{8}$$

in (7), h_k is the maximal height of triangle T_k, $\|\vec{\nabla}\Theta\|_2$ is the Euclidian norm of the temperature gradient, and $\frac{1}{h_k}[\vec{\nabla}\Theta]_k$ is an approximation of $\Delta\Theta$ around the triangle T_k, given by:

$$[\vec{\nabla}\Theta]_k = \frac{1}{3} \sum_{T_{k'} \; neighbor \; of \; T_k} (\vec{\nabla}\Theta|_{T_k}.\vec{n} - \vec{\nabla}\Theta|_{T_{k'}}.\vec{n}) , \tag{9}$$

$\vec{n}$ being the outward unit normal on ∂T_k. Moreover, $\mathcal{V}$ is an adjustable coefficient (for homogeneity reasons, $\mathcal{V}$ can be chosen equal to the flame

speed, which can be evaluated using the high activation energies asymptotics; see [4]), and μ_Θ and μ_Y are proportional to the differences $\max_{S_i \in \mathcal{D}} \Theta_i^n - \min_{S_i \in \mathcal{D}} \Theta_i^n$ and $\max_{S_i \in \mathcal{D}} Y_i^n - \min_{S_i \in \mathcal{D}} Y_i^n$ respectively.

Because this criterion is local, we need to smooth it out, using a standard Laplacian-type regularization. This consists in performing some iterations of:

$$\frac{\delta_{l+1} - \delta_l}{\tau} = \Delta \delta_l \ , \tag{10}$$

with $\delta_0 \equiv \eta^n$.

The values $\hat{\eta}^n(T_k)$ of the regularized criterion are then used to fill the array IADIV which governs the refinement and derefinement decisions: the real numbers $\beta_{m_{\max}} = 1 > \beta_{m_{\max}-1} > \cdots > \beta_2 > \beta_1 > \beta_0 > 0$ being given, one sets $\texttt{IADIV}(T_k) = m$ with $1 \leq m \leq m_{\max}$ if:

$$\beta_{m-1} \max_{k'} \hat{\eta}^n(T_{k'}) < \hat{\eta}^n(T_k) \leq \beta_m \max_{k} \hat{\eta}^n(T_{k'}) \ ; \tag{11}$$

if $\hat{\eta}^n(T_k) \leq \beta_0 \max_{k'} \hat{\eta}^n(T_{k'})$, the macro-element is to remain undivided and one sets $\texttt{IADIV}(T_k) = 0$. The subprogram ADAPT can then be called to construct a new triangulation.

Let us add a few remarks here before presenting our results. Concerning the criterion evaluation, it can be noticed that we add to the usual error estimate based on the second order derivative two quantities: the reaction rate Ω and the diffusive fluxes $\|\vec{\nabla}\Theta\|_2$ and $\|\vec{\nabla}Y\|_2$. A certain number of parameters are introduced in this criterion evaluation: the degree of regularization, the coefficient $\mathcal{V}$, and the thresholds β_0, β_1, $\cdots$, $\beta_{m_{\max}}$ for the refinement decisions; the choice of these parameters, and also the choice of those time levels when ADAPT is called during the calculation, remain rather critical for the success of the adaptive calculation; we refer to [12] and [11] for more details and further studies concerning these points. Lastly, we need to add here that, when the procedure ADAPT is used for the solution of an actual physical problem, as we describe now, the interpolation of the variables at the new points created by ADAPT is done inside the ADAPT procedure: at each time when a new vertex is created, the variables Θ and Y are interpolated at this new point (in the present study, we simply use here a linear interpolation on the edge which is cut by the new point).

4.2 Results

We now present how the dynamic adaptive procedure performs when applied to the flame propagation problem (1). We first consider the propagation of an initially curved flame inside an infinite rectangular open tube, with a reduced activation energy (or Zeldovich number) equal to 10. Figure 10 shows the evolution of the adaptive triangulation (with two levels of refinement) and of the reaction rate contours. We should make precise that in practice we do not adapt the mesh at each time step, but integrate over several time steps on the same triangulation, until the flame has been sufficiently moving since the last adaption.

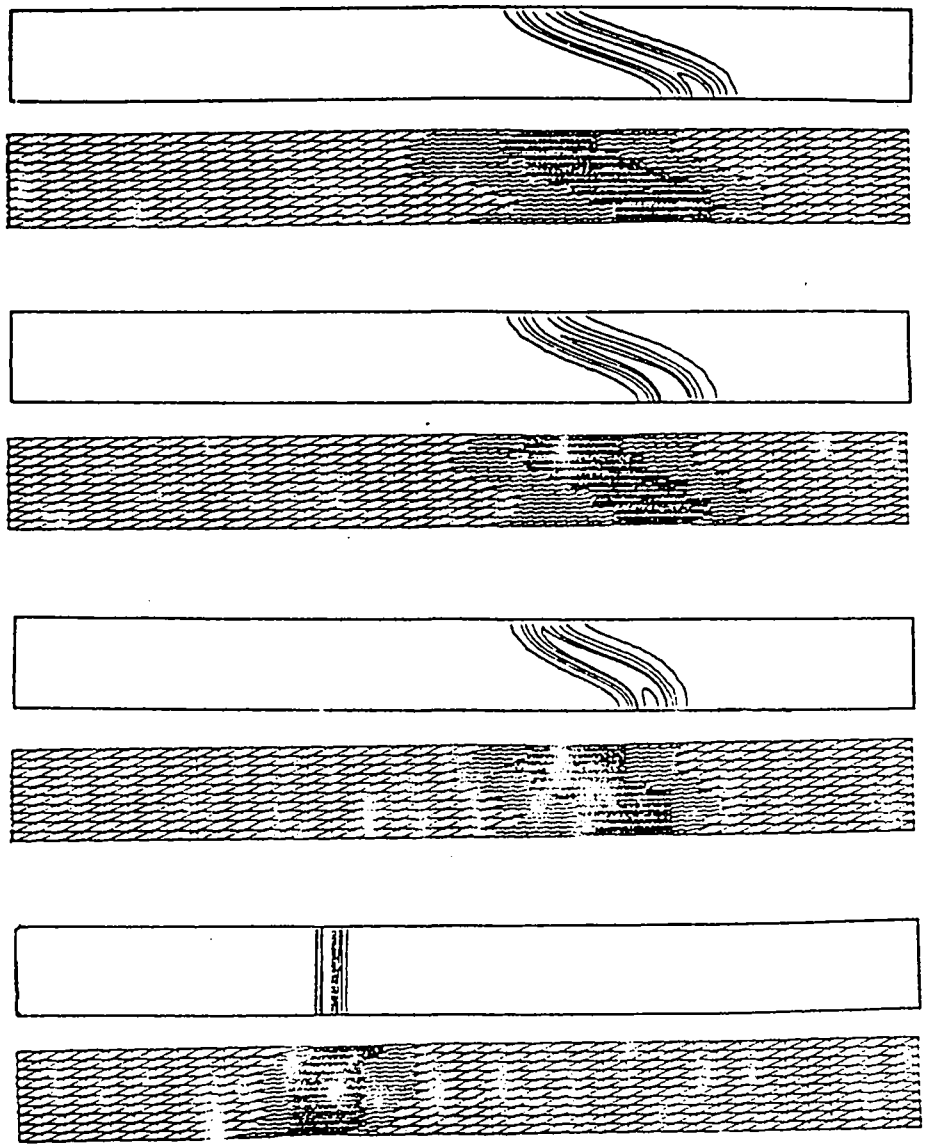

Figure 10: Curved flame in an infinite open tube; reaction rate contours and refined mesh.

A similar experiment with a planar flame with three refinement levels is illustrated in Figure 11.

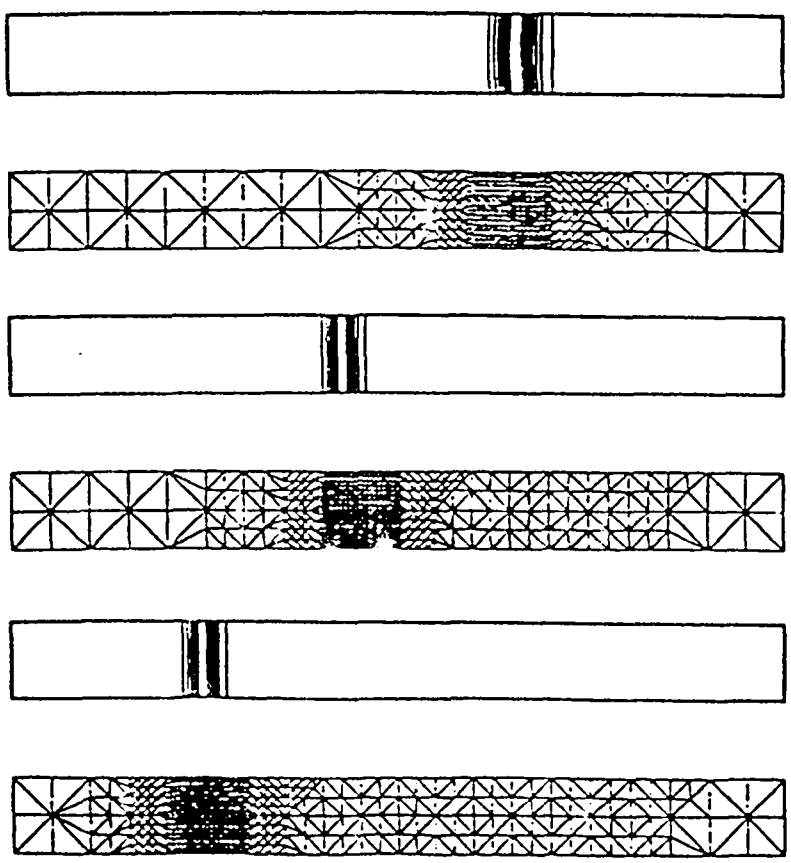

Figure 11: Planar flame in an infinite open tube; reaction rate contours and refined mesh.

Figure 12 shows a flame propagating in a closed square box initially filled with cold gases. The macro-element triangulation is a uniform 31×31 mesh. Two levels of refinement are again used, and very accurate results are obtained, as shown by the smoothness of the reaction rate contours. The CPU time for this last calculation is 400 minutes on a CONVEX C1 used in scalar mode, for 9200 explicit time steps, whereas around 1900 minutes would have been necessary to perform the same accuracy on a uniform 121×121 mesh equivalent to the finest level of refinement. This efficiency ratio, which is here larger than 4.5, is of course expected to increase when more complex systems of partial differential equations are considered. Lastly, the average number of nodes during the adaptive calculation is close to 3000; 5% of the computer time is devoted to the refinement-derefinement process, 95% being devoted to the integration of system (1).

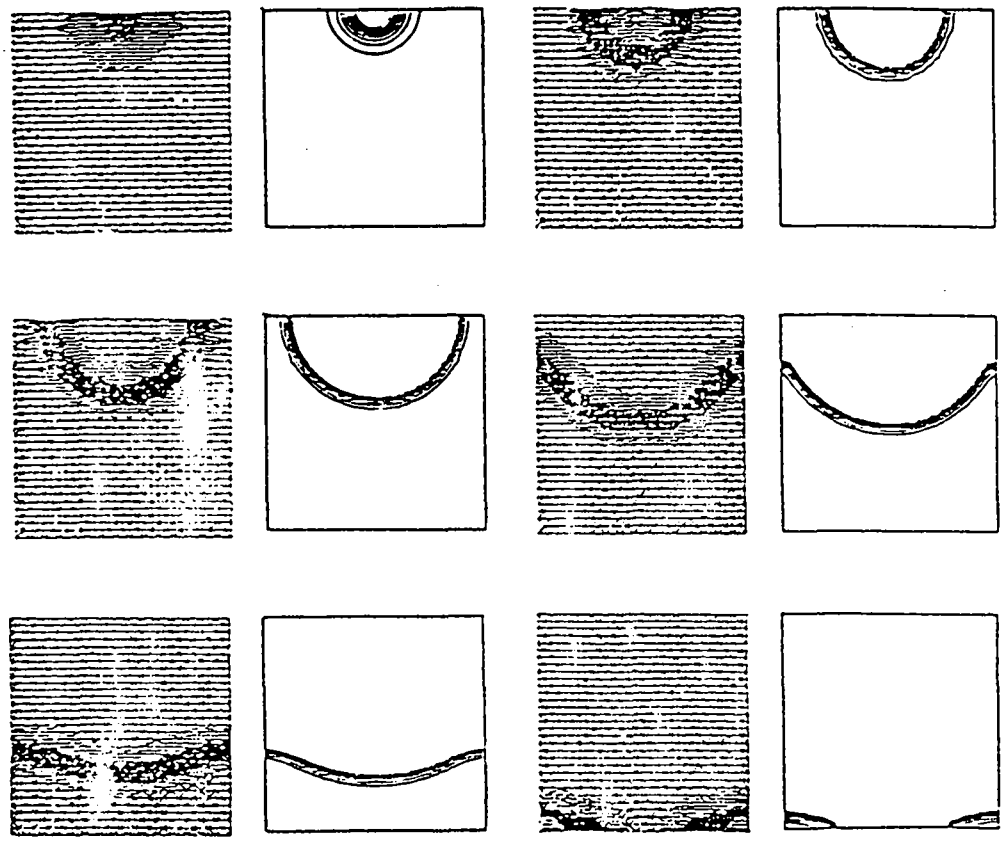

Figure 12: Curved flame in an closed square box; reaction rate contours and refined mesh.

5 CONCLUSION

The dynamic adaptive refinement-derefinement procedure presented in this paper gives efficient and accurate finite-element solutions of the stiff reaction-diffusion system (1). Further improvements are still needed, in several directions, most of them being the subject of our current research: optimal choice of the parameters involved in the criterion definition (see [11], [23]), solution of a more complex model including compressible fluid dynamics effects (see [14]), evaluation of the criterion governing the mesh adaption on the current triangulation and not only on the macro-elements (see [23]).

References

[1] F. ANGRAND & P. LEYLAND, "Schéma multigrille dynamique pour la simulation d'écoulements de fluides visqueux compressibles", INRIA Report 659, (1987).

[2] F. ANGRAND & P. LEYLAND, "Compressible viscous flow simulation by multigrid methods", Comp. Meth. Appl. Mech. Eng., **75**, pp. 167-183. (1989).

[3] I. BABUSKA & W. C. RHEINBOLT, "Error estimates for adaptive finite-element computations", SIAM J. Num. Anal., **15**, pp. 736-754, (1978).

[4] I. BABUSKA, O. C. ZIENKIEWICZ, J. CAGO & E. R. OLIVERA, "Accuracy estimates and adaptive refinements in finite element computations", Wiley, (1986).

[5] T. BAKER & A. JAMESON, "Multigrid solution of the Euler equations for aircraft configurations", AIAA 84-0093, (1984).

[6] R. E. BANK, "PLTMG User's Guide", Univ. California San Diego, (1985).

[7] R. E. BANK & A. H. SHERMAN, "An adaptive multi-level method for elliptic boundary value problems", Computing, **26**, pp. 91-105. (1981).

[8] R. E. BANK, A. H. SHERMAN & A. WEISER, "Refinement algorithms and data structures for local mesh refinement, Scientific Computing, Stepleman et al. eds., IMACS, North Holland, Amsterdam, (1983).

[9] J. D. BAUM & R. LÖHNER, "Numerical simulation of shock-elevated box interaction using adaptive finite-element shock capturing scheme" AIAA-89-0653, (1989).

[10] F. BENKHALDOUN & B. LARROUTUROU, "Explicit adaptive calculations of wrinkled flame propagation", Int. J. Num. Meth. Fluids, **7**, pp. 1147-1158, (1987).

[11] F. BENKHALDOUN, B. LARROUTUROU, P. LEYLAND & N. MAMAN, "Criteria for dynamical mesh adaption for unsteady problems", in preparation.

[12] F. BENKHALDOUN, P. LEYLAND & B. LARROUTUROU, "Dynamic mesh adaption for unsteady nonlinear phenomena - Application to flame propagation", Numerical grid generation in computational fluid mechanics 88, Sengupta et al. eds., pp. 977-986, Pineridge Press, Swansea, (1988).

[13] D. CHARGY, A. DERVIEUX, B. LARROUTUROU, M. LORIOT & N. MAMAN, "Simulation numérique de l'interaction réactive de jets gazeux transsoniques", in preparation.

[14] P. G. CIARLET, "The finite-element method for elliptic problems", North Holland, Amsterdam, (1978).

[15] P. CLAVIN, "Dynamic behaviour of premixed flame fronts in laminar and turbulent flows", Prog. Energ. Comb. Sci., **11**, pp. 1-59, (1985).

[16] K. ERIKKSON, "Adaptive finite-element methods based on optimal error estimates for linear elliptic problems", Chalmers University of Technology, (1987).

[17] K. ERIKKSON & C. JOHNSON, "Adaptive finite-element methods for parabolic problems I: A linear model problem", Univ. Göteborg, (1988).

[18] K. ERIKKSON, C. JOHNSON & J. LENNBLAD, "Error estimates and automatic time and space step control for linear parabolic problems", SIAM J. Num. Anal., to appear.

[19] J. M. HYMAN & B. LARROUTUROU, "On the use of adaptive moving grid methods in combustion problems", Complex chemical reaction systems - Mathematical modelling and simulation, Warnatz & Jäger eds., Springer series in chemical physics, pp. 222-232, Springer Verlag, Heidelberg, (1987).

[20] C. JOHNSON, "Adaptive finite-element methods for diffusion and convection problems", Comp. Meth. Appl. Mech. Eng., to appear.

[21] M. P. LECLERCQ, Thesis, Univ. Saint-Etienne, (1990).

[22] R. LÖHNER, "An adaptive finite-element scheme for transient problems in CFD", Comp. Meth. Appl. Mech. Eng., **61**, pp. 323-338, (1987).

[23] N. MAMAN, Thesis, Univ. Toulouse, (1990).

[24] R. H. NOCHETTO & C. VERDI, "An efficient linear scheme to approximate parabolic free boundary problems: error estimates and implementation", Math. Comp., to appear.

[25] R. H. NOCHETTO, M. PAOLINI & C. VERDI, "Local mesh refinements for two-phase Stefan problems in two space variables, Istituto di Analisi Numerica, Pavia, (1988).

[26] J. PERAIRE, M. VAHDATI, K. MORGAN & O. C. ZIENKIEWICZ, "Adaptive remeshing for compressible flow computations", J. Comp. Phys., **72**, pp. 449-466, (1987).

[27] C. POULETTY, Thesis, Ecole Centrale des Arts et Manufactures, Paris, (1985).

[28] M. C. RIVARA, "Algorithms for refining triangular grids suitable for adaptive and multigrid techniques", Int. J. Num. Meth. Eng., **20**, pp. 745-756, (1984).

[29] M. C. RIVARA, "A grid generator based on 4-triangles conforming mesh-refinement algorithms", Int. J. Num. Meth. Eng., **24**, pp. 1343-1354, (1987).

[30] T. STROUBOLIS, P. DEVLOO & J. T. ODEN, "A moving-grid finite-element algorithm for supersonic flow interaction between moving bodies", Comp. Meth. App. Mech. Eng., **59**, pp. 235-255. (1986).

[31] T. USHIJIMA, "Error estimates for the lumped-mass approximation of the heat equation", Mem. Num. Math., **6**, pp. 65-82, (1979).

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249 - 6399