Konrad-Zuse-Zentrum f r Informationstechnik Berlin

Jochen Fröhlich[1] Jens Lang Rainer Roitzsch

# Selfadaptive Finite Element Computations with Smooth Time Controller and Anisotropic Refinement

To be published in ECCOMAS 96 by John Wiley & Sons, Ltd.

[1] Institut für Hydromechanik, Universität Karlsruhe, Kaiserstr. 12,
76128 Karlsruhe

Preprint SC 96–16 (May 1996)

Jochen Fröhlich Jens Lang Rainer Roitzsch

# Selfadaptive Finite Element Computations with Smooth Time Controller and Anisotropic Refinement

**Abstract** We present Multilevel Finite Element computations for twodimensional reaction-diffusion systems modelling laminar flames. These systems are prototypes for extreme stiffness in time and space. The first of these two rather general features is accounted for by an improved control mechanism for the time step. The second one is reflected through very thin travelling reaction fronts for which we propose an anisotropic discretization by local directional refinement.

## 1 Introduction

Adaptive hierarchical finite element methods have been proposed to solve PDEs with highly non-uniform solution [4]. They employ a hierarchy of nested finite element spaces constructed from coarse to fine levels. By repeated application of solving, error estimation, and local refinement the final grid is well adapted to the required solution. Furthermore, computing the solution on grids of different scale speeds up the iterations on each level due to the related multilevel decomposition. The method has shown to be satisfactory in terms of flexibility with respect to the equations to be solved, the boundary conditions and the geometry.

The basic MLFEM (Multilevel Finite Element Method) concept for elliptic PDEs has been applied to linear parabolic PDEs using a time-space (TS) discretization sequence [3]. It proceeds in reversed order with respect to the Method of Lines, the classical space-time (ST) discretization sequence, and also allows additional adaptivity in time. This has further been extended to systems of nonlinear one- and twodimensional problems using linearly implicit Runge-Kutta methods [11], [10]. An adaptive time discretization involves an algorithm proposing a time increment for the next step. In the present paper we apply a method from [8] and show that an improved control mechanism of the time step yields nonnegligible savings with almost no extra cost.

Another and more difficult topic for MLFEM is the refinement strategy in the presence of an unsteady anisotropic solution. Current strategies for triangular grids ("red" refinement, longest edge bisection etc.) aim at reducing the step size of the grid in all directions simultaneously. In this way it is relatively easy to fulfil the maximum angle condition [1] required for satisfactory approximation properties of the FEM. (Bad conditioning due to decreasing minimal angle is generally remedied by appropriate preconditioning.) The resulting discretizations are versatile and robust, however, they are sub-optimal in cases where refinement is not required in all directions. A prototype example for such a situation is constituted by propagating reaction fronts which we shall consider in the present paper. They exhibit extreme steepness in normal direction, i.e. the direction of propagation, whereas the solution is almost constant in tangential direction. In such a case one would like to use a refinement technique reducing the mesh width only normal and not tangential to the front. Several strategies can be employed, but up to now this problem does not seem to be satisfactorily solved. We will briefly discuss the encountered difficulties and propose a compromise between improved discretization with a reduced number of degrees of freedom and increased computational overhead.

## 2 Basic algorithm

Let us shortly recall the main features of the method referring to [10], [7] and the references therein for further information. It has been developed and implemented in the code KARDOS for second order parabolic problems of the form

$$\begin{aligned} P(x)\partial_t u + A(\partial_x, x)\, u &= F(u), \\ u(0,x) &= u_0(x), \end{aligned} \tag{1}$$

where $u$ is the $d-$dimensional vector of dependent variables and $x \in \Omega \subset \mathbb{R}^2$. $P$ is an $x$-dependent $d \times d$-matrix and $A$ is a linear elliptic differential operator of second order with respect to the spatial variable $x$. The computational domain $\Omega$ is bounded and appropriate boundary conditions on $\partial\Omega$ are assumed to be incorporated into the operator $A$. In the present applications the functions $u$, $u_0$ and $F$ are vectors of two real functions. We consider the so-called thermodiffusive equations, see e.g. [15], describing the propagation of a laminar flame in a premixed atmosphere neglecting the gas expansion

$$\partial_t\theta - \quad \nabla^2\theta \;=\; \omega \tag{2}$$

$$\partial_t Y - \frac{1}{Le}\nabla^2 Y \;=\; -\omega \tag{3}$$

where the Lewis number $Le$ is the ratio of diffusivity of heat and diffusivity of mass. We use a simple one-species reaction mechanism governed by an Arrhenius law

$$\omega \;=\; \frac{\beta^2}{2Le}\, Y\, e^{\frac{-\beta(1-\theta)}{1+\alpha(1-\theta)}}. \tag{4}$$

With an appropriate choice of the parameters and the boundary conditions these equations constitute severe test cases for numerical solution schemes [13]. Below we use $\beta = 10,\ \alpha = 0.8$.

Employing a MLFEM to an unsteady problem it is convenient to discretize first in time then in space. This takes account of the fact that the main complexity of the problem appears in space. The latter can be dealt with more easily since the grids are independently adapted in each time step as opposed to the ST-discretization sequence. (A thorough comparison of both approaches can be found in [5].) In [11] a singly diagonally linearly implicit Runge-Kutta method of order three performed best. It reads

$$\begin{aligned}\left(\frac{1}{\gamma\,\tau_n}P - \partial_u f(u_{n-1})\right) l_j = \\ f(\,u_{n-1} + \textstyle\sum_{i=1}^{j-1} a_{ji}\,l_i\,) + \frac{1}{\tau_n}\,P\,\sum_{i=1}^{j-1}\,c_{ji}\,l_i\end{aligned} \tag{5}$$

$$u_n = u_{n-1} + \sum_{j=1}^{3} b_j\,l_j\ . \tag{6}$$

with

$$f(u) = F(u) - A(\partial_x, x)u \tag{7}$$

and suitably chosen values for the coefficients $\gamma$, $a_{ji}$, $c_{ji}$ and $b_j$ [14]. Replacing the coefficients $b_j$ in (6) by different coefficients $\hat{b}_j$ a second order solution $\hat{u}_n$ can be obtained. Employing an appropriate norm [7] the difference between both solutions $\epsilon_n = \|u_n - \hat{u}_n\|_{\Omega}$ satisfactorily estimates the error introduced by the temporal discretization. If $\epsilon_n$ is larger than some tolerance $TOL_t$, the solution $u_n$ is rejected and the step is repeated with a reduced time step $\tau_n$.

For each stage value $l_j$ in (6) an elliptic problem (5) has to be solved with appropriate boundary conditions. For reasons of efficiency the same grid is used for all stages $j = 1, 2, 3$. The MLFEM in space then starts from an initial triangulation $T_0$ resolving the geometry of the problem. In the basic algorithm the successively finer grids $T_k$, $k = 1, 2, 3, \ldots$ are then constructed by local "red" refinement, i.e. by dividing a triangle into four congruent smaller ones. We use a conformal discretization with linear trial functions and apply "green" refinement (bisection) to avoid slave nodes. "Red" refinement yields nested FE spaces. In order to avoid grid degeneration "green" closures are removed before further refinement. Of course, the nesting does no longer hold for such elements, but it causes no practical problems. On each level the solution is obtained iteratively employing BICGSTAB with SSOR preconditioning. Since an estimation procedure for the spatial error involving all stage values would be too costly, a first order solution $u^k_{Euler}$ is constructed from $l^k_1$ on the grid $T_k$ and used for this purpose. The employed error estimation is then based on the solution of local subproblems determining an approximate local residual [2]. Fig. 1 gives an overview of the basic algorithm, by means of a flow diagram.

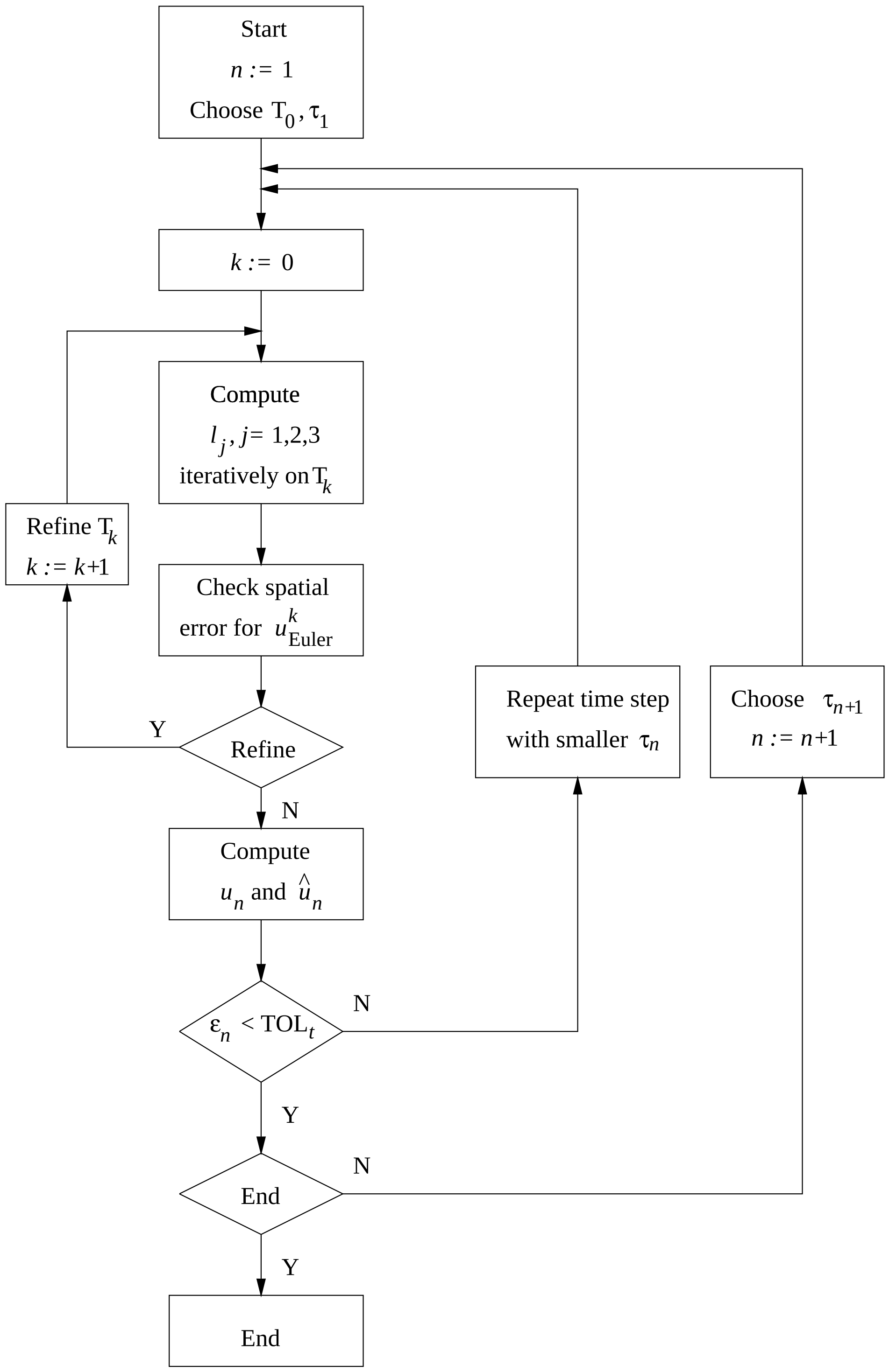

Figure 1: Flow chart for the time-space adaptive solver KARDOS.

# 3 Time control

From a global tolerance $TOL$ suitable tolerances for the error in space and the error in time are deduced through $TOL_t = TOL/2$, $TOL_x = TOL/3$ [7]. Based on the error in time $\epsilon_{n-1}$ of the previous time step a value for the new time increment is classically chosen according to

$$\tau_n \;:=\; \rho \left(\frac{TOL_t}{\epsilon_{n-1}}\right)^{1/3} \tau_{n-1}\,, \tag{8}$$

where $\rho$ denotes a safety factor. The exponent in (8) results from the cubic model for the local error in time. Unfortunately, this mechanism often leads to a nonsmooth behavior of the time integration process. For instance, after a drastic step size reduction the corresponding error $\epsilon_{n-1}$ becomes very small. Consequently, the proposed new time step will be too optimistic leading to repeated rejections. A possible remedy is to smooth the step size selection using a PI-controller as introduced in [8] for implicit methods. It takes the form

$$\tau_n := \rho\,\frac{\tau_{n-1}}{\tau_{n-2}} \left(\frac{TOL_t}{\epsilon_{n-1}}\frac{\epsilon_{n-2}}{\epsilon_{n-1}}\right)^{1/3} \tau_{n-1} \tag{9}$$

and is used in cases of more than two successively accepted time steps. If time steps are rejected, relation (8) is used with several modifications.

To illustrate the above remarks we solve equations (2), (3) in the onedimensional case with $Le = 2$. This Lewis number generates an unregularly oscillating propagation of the flame [13]. The adaptive selection of the time step in our method takes this into account and reduces or increases the step size accordingly. Fig. 2 shows the employed values over several oscillations with $TOL = 10^{-3}$.

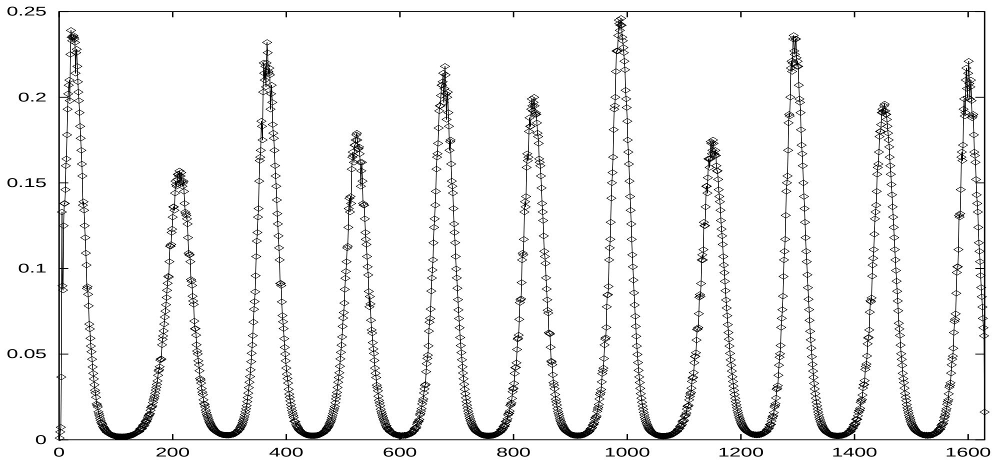

Figure 2: Selected time step $\tau_n$ versus $n$ for the onedimensional oscillating flame and $t = 0, \ldots, 100$

The time step varies over more than two orders of magnitude. Fig. 3 depicts a zoom of this plot during acceleration of the flame. It reveals that computational effort is wasted since many computed solutions have to be rejected for reasons of precision. Employing (9) instead of (8) yields a figure which is almost identical to Fig. 3, since the accepted time steps reflect the physics of the problem, but where

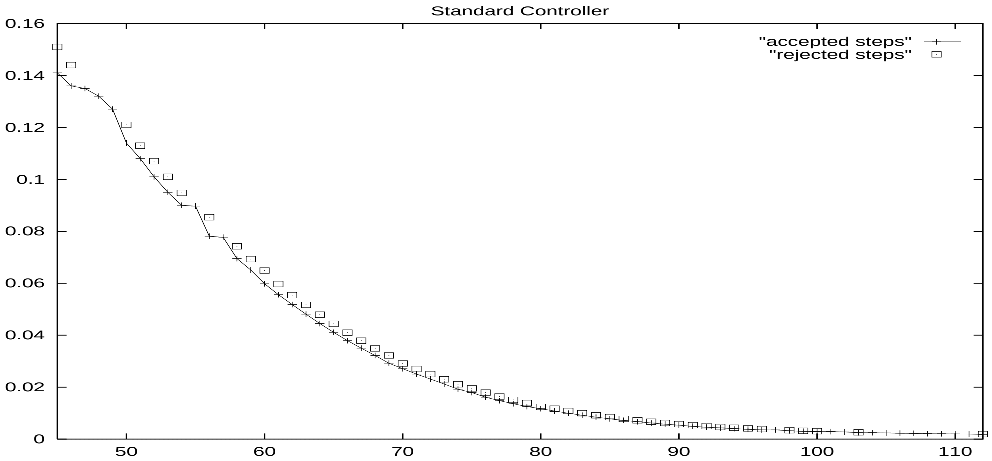

Figure 3: Selected time step $\tau_n$ versus $n$ for a critical phase of acceleration $t = 6.83, \dots, 9.03$ employing the P-controller (8). The squares indicate solutions computed with the related time step which had to be rejected. The time increment was too large and thus generated an error in time larger than the prescribed tolerance.

no computed solutions have been rejected. The resulting reduction in CPU time was about 20% for the entire run. For smaller $TOL_t$ the effect is less pronounced. Although the observed saving is not extremely large it should be recalled that it has been obtained with merely no cost and that it makes the algorithm more robust in situations with sudden changes in time.

# 4 Anisotropic refinement

Classical MLFEM algorithms start from a solution-independent coarse grid. It has to capture the geometry and has to yield a solution which permits to adequately determine the required refinement. The classical strategy furthermore does not remove or shift points or edges since aiming at nested FE function spaces. Considering an anisotropic solution in form of a front we observe the following fact. In order to respect the maximum angle condition mentioned in the introduction with narrow spacing normal to the front and wide spacing in tangential direction an optimal grid would have the points aligned in the direction of the gradient with large distances normal to the gradient. By construction such an optimal grid can not be achieved through classical subdivision of coarse triangles not containing this directional information.

Different strategies can be devised to cope with the dilemma. One is to generate a solution dependent coarse grid. For unsteady problems the solution $u_{n-1}$ of the previous time level can be used for this task. The coarse grid is then doted with directional information on the solution. This anisotropy is subsequently inherited by the finer grids generated through subdivision in the induced nonuniform metric. Since the coarse grid contains only relatively few points its generation is possible with very low cost, e.g. by the method described in [12].

Even if this strategy seems appealing we choose a different approach here for the following reason. During the solution of (5) the result of the previous time level $u_{n-1}$ has to be interpolated permanently from the old grid to the grid of the new time level (recall that it is not only required for the r.h.s. in (5) but also for the Jacobian etc.). If both grids are generated from the same coarse grid and if they use the same tree structure, just with a locally different depth, search and interpolation between both solutions can be implemented very efficiently. This is ensured by the classical refinement procedure described above. As soon as the coarse grid or the tree structure are not longer the same this task becomes much more intricated. We therefore apply a special treatment only in regions where the higher computational overhead is justified by the improved discretization. In [9] an anisotropic discretization method has been proposed and applied to the FE approximation of a discontinuous function. It consists of two parts: a) a multilevel shock fitting procedure which optimizes the location of newly inserted points in the vicinity of a shock by shifting them along the edges, b) a directional or "blue" refinement. Due to the high complexity of the shock fitting part we retain here only the directional refinement which is defined as follows (see Fig. 4). Two neighboring triangles generating a convex quadrangle are divided into four triangles by 1) removing the common diagonal, 2) selecting a direction of refinement, 3) dividing the two opposite edges pointing (approximately) in this direction and joining them by a new edge, 4) dividing the resulting quadrangles into two triangles. This is done by connecting the newly inserted points with their opposite node in order to conserve the orientation of the edges in each triangle (slave nodes are avoided by "green" or "red" closures). Note that this procedure is exactly equivalent to bisection of the original triangles with subsequent swap of the diagonal, a classical technique for grid enhancement. Furthermore, we observe that the FE spaces are not nested with this refinement rule.

In order to efficiently decide where to apply "blue" refinement we define a line along which the maximum gradient is supposed to appear (see Fig. 4) [9]. In many applications such a criterion can be furnished from physical reasoning.

In the computations (see Fig. 5) we choose a level line of the temperature $T = T_{blue} = 0.8$ as this quantity is normalized to $[0, 1]$ and the maximum slope for a onedimensional plane front is known to appear around this value for the employed parameters. Instead of determining the gradient of the solution all over the domain and then deciding where and how to apply directional refinement we first limit the region for such candidates to the triangles intersecting with the

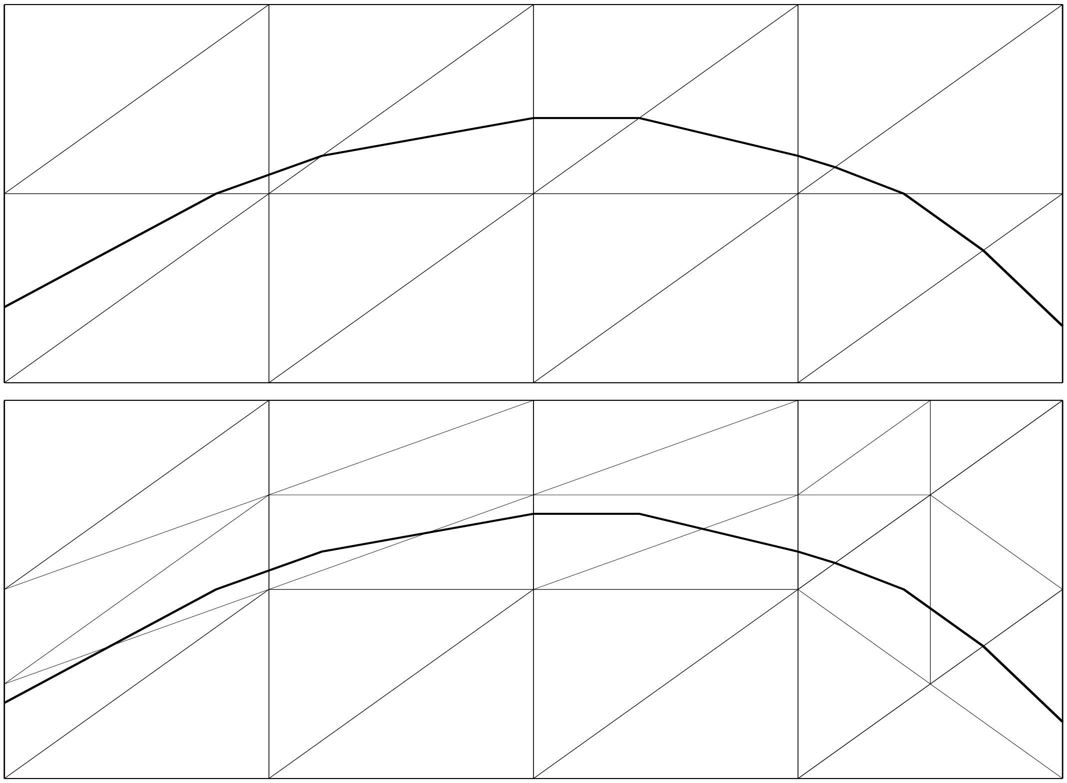

Figure 4: “Blue” refinement along a level line. Top: initial grid with a level line crossing a set of triangles that are “blue” candidates. Bottom: resulting grid after “blue” refinement.

chosen level line (if the reaction rate is not vanishing at these locations). Several criteria are then used to determine which common edge is to be removed and which edges should be refined. The whole algorithm combining isotropic (“red”) and anisotropic (“blue”) refinement from $T_k$ to $T_{k+1}$ reads as follows: 1) remove all “green” closures, 2) according to the estimated error mark a certain set of triangles $R_k \subset T_k$ for refinement, 3) search for candidates to “blue” refinement in $R_k$ according to the above criterion, 4) determine the direction of the gradient and apply the anisotropic refinement, 5) apply “red”refinement to all remaining triangles in $R_k$, 6) and avoid slave nodes by further “red” or “green” refinement.

In Fig. 5 the above refinement procedure has been applied to the computation of a twodimensional flame with $Le = 1$ traversing a cooled obstacle modelled through the boundary condition $\partial_n \theta = k\theta, k = 0.1$ [7]. The right-travelling flame is shown after leaving the obstacle. The upper part of the figure has been computed with the standard refinement procedure, the lower part with local directional refinement. The resulting number of grid points is $N = 3124$ with only the “red” refinement and $N = 2367$ with partial “blue” refinement while the achieved accuracy of the solution is similar ($TOL = 5 \cdot 10^{-3}$ in both cases). Fig. 6 depicts a zoom of the grid with directional refinement. The employed temperature level line is included for illustration. The CPU time for both computations can

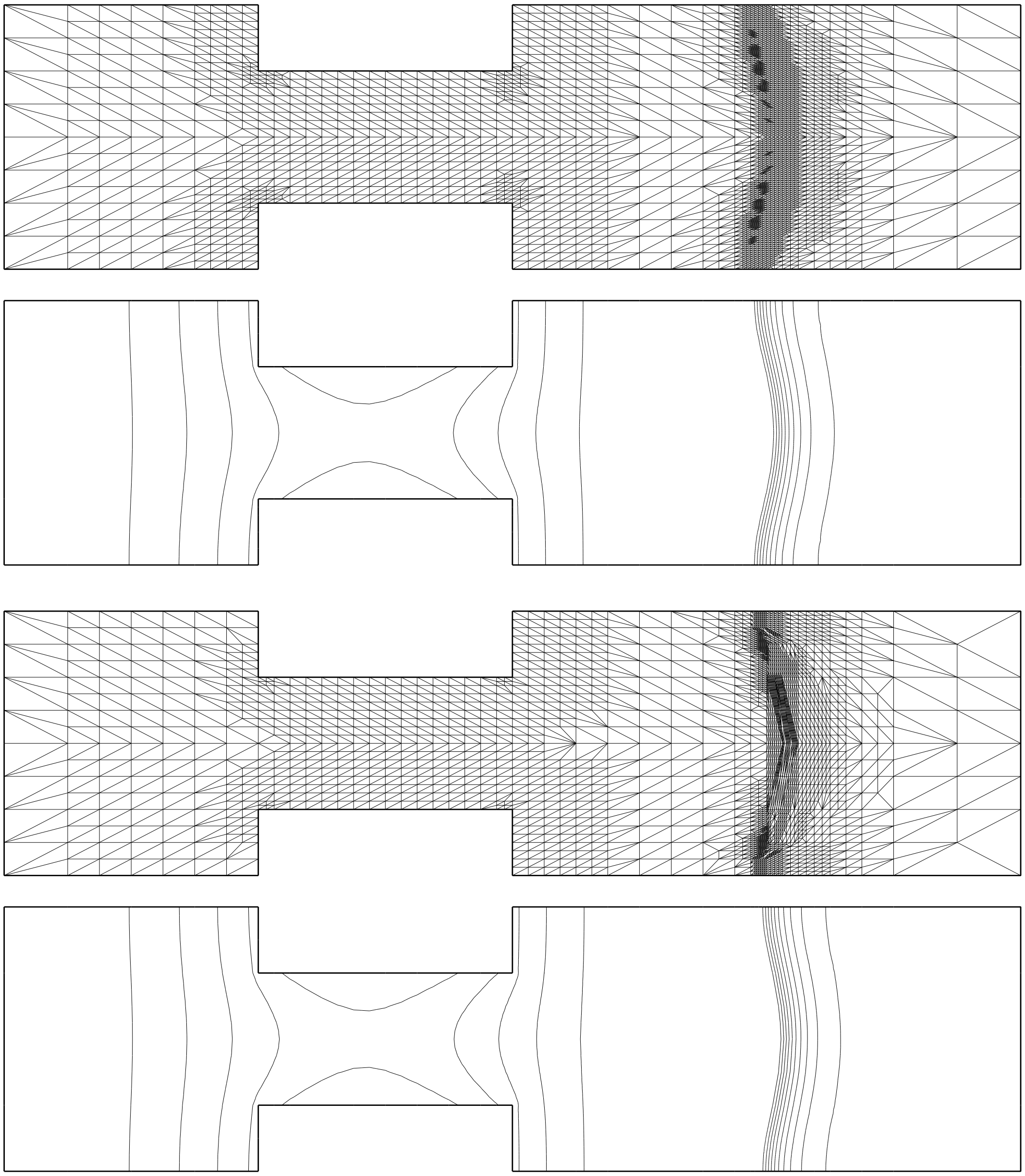

Figure 5: Flame computation with partial directional refinement. Top to bottom: a) grid with isotropic refinement at $t = 60$ with $N = 3124$, b) temperature level lines obtained with grid in a), c) grid with local "blue" refinement at $t = 60$ with $N = 2367$, d) temperature level lines obtained with grid in c).

not be compared on a fair basis since the more involved data management for the directional refinement has not yet been implemented with the same sophistication as the classical method.

In summary, the need for anisotropic discretization is obvious from the presented computations. The proposed method constitutes a step in this direction within the framework of a MLFEM. However, it still deserves improvement with

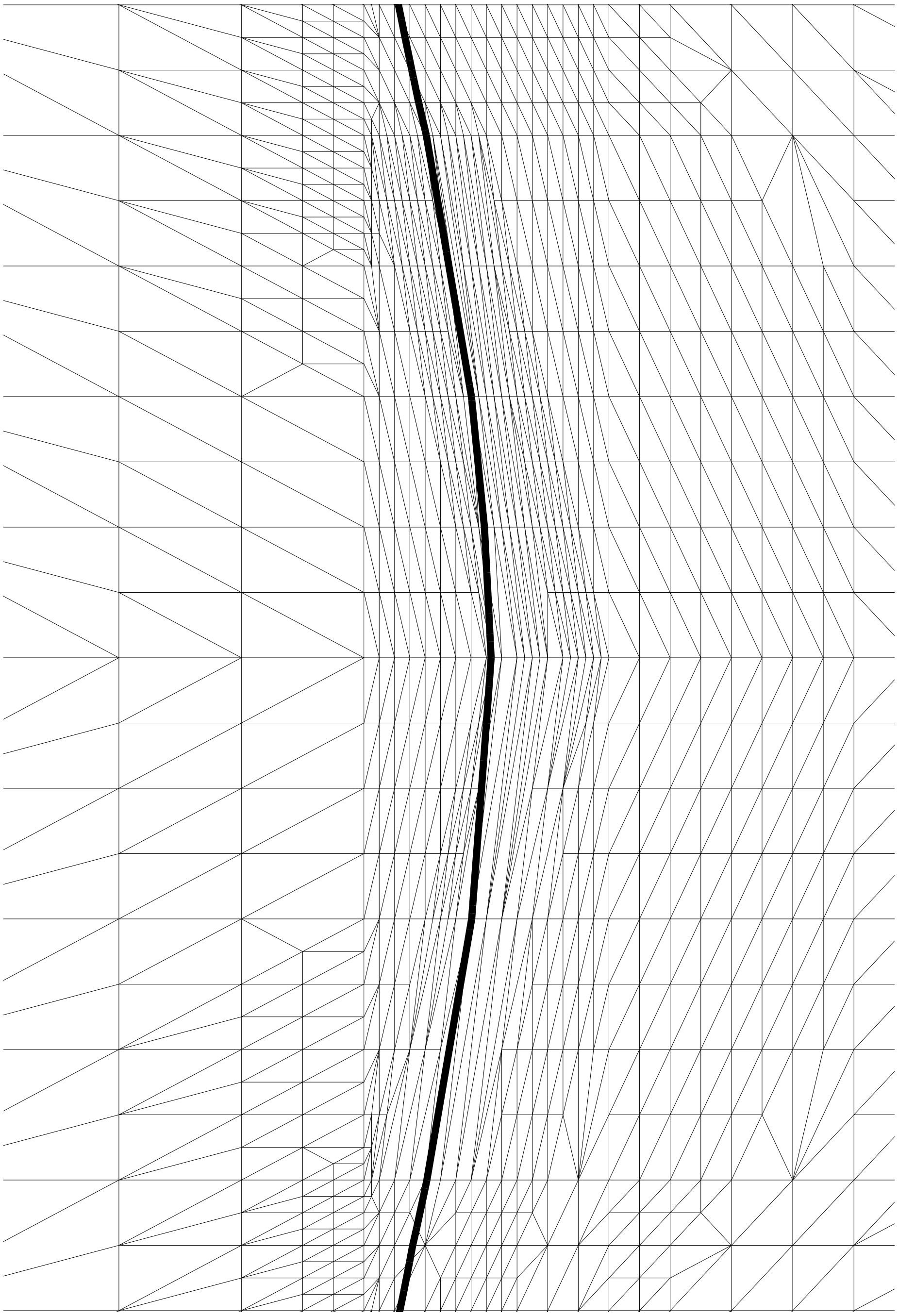

Figure 6: Zoom of Figure 5c around the flame surface in the middle of the channel.

respect to coding and to the criterion for “blue” refinement.

**Acknowledgement.** The authors are deeply indebted to P. Deuflhard for inspiring discussions on stiff ODE integrators and adaptive multilevel methods. Thanks to R. Kornhuber for keeping the interest in anisotropic refinement alive.

# References

[1] I. Babuška, A. K. Aziz. On the angle condition in the finite element method. *SIAM J. Numer. Anal. 13*, 214–226, 1976.

[2] I. Babuška and W.C. Rheinboldt. A Posteriori Error Estimates for the Finite Element Method. *Int. J. Numer. Meth. in Eng. 12*, 1597–1615, 1978.

[3] F.A. Bornemann. An Adaptive Multilevel Approach to Parabolic Equations. *IMPACT Comput. Sci. Engrg. 2*, 279–317, 1991.

[4] P. Deuflhard, P. Leinen, H. Yserentant. Concepts of an Adaptive Hierarchical Finite Element Code. *IMPACT Comput. Sci. Engrg. 1*, 3–35, 1989.

[5] P. Deuflhard, J. Lang, U. Nowak. Adaptive Algorithm in Dynamical Process Simulation. In *Proceedings of the 8th ECMI Conference Sept. 1994 Kaiserslautern.* Teubner, 1996.

[6] B. Erdmann, J. Lang, R. Roitzsch. KASKADE Manual - Version 2.0. Technical Report TR 93-5, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1993.

[7] J. Fröhlich, J. Lang. Twodimensional Cascadic Finite Element Comoputations of Combustion Problems. Preprint SC 96-5, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1996.

[8] K. Gustafsson. *Control of error and convergence in ODE solvers.* PhD thesis, Department of Automatic Control, Lund Institut of Technology, Lund, Sweden, 1992.

[9] R. Kornhuber, R. Roitzsch. On Adaptive Grid Refinement in the Presence of Internal or Boundary Layers. *IMPACT Comput. Sci. Engrg. 2*, 40–72, 1990.

[10] J. Lang. Two-dimensional fully adaptive solutions of reaction–diffusion equations. *Appl. Numer. Math. 18*, 223–240, 1995.

[11] J. Lang, A. Walter. A Finite Element Method Adaptive in Space and Time for Nonlinear Reaction-Diffusion Systems. *IMPACT Comput. Sci. Engrg. 4*, 269–314, 1992.

[12] K. Morgan, J. Peraire, J. Peiro. Unstructured Grid Methods for Compressible Flows. AGARD Report 787, Special Course on Unstructured Grid Methods for Advection Dominated Flows, 1992.

[13] N. Peters, J. Warnatz (eds.). *Numerical Methods in Laminar Flame Propagation, Notes on numerical fluid mechanics, Vol. 6.* Vieweg, 1982.

[14] M. Roche. Rosenbrock Methods for Differential Algebraic Equations. *Numer. Math. 52*, 45–63, 1988.

[15] F.A. Williams. *Combustion Theory.* Addison-Wesley, 1985.