



ELSEVIER

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

SCIENCE @ DIRECT®

Comput. Methods Appl. Mech. Engrg. 193 (2004) 405–425

**Computer methods  
in applied  
mechanics and  
engineering**

[www.elsevier.com/locate/cma](http://www.elsevier.com/locate/cma)

# Adaptive multiresolution approach for two-dimensional PDEs

J.C. Santos <sup>a</sup>, P. Cruz <sup>a</sup>, M.A. Alves <sup>b,\*</sup>, P.J. Oliveira <sup>c</sup>,  
F.D. Magalhães <sup>a</sup>, A. Mendes <sup>a</sup>

<sup>a</sup> *LEPAE–Departamento de Engenharia Química, Faculdade de Engenharia da Universidade do Porto,  
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal*

<sup>b</sup> *Departamento de Engenharia Química, CEFT, Faculdade de Engenharia da Universidade do Porto,  
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal*

<sup>c</sup> *Departamento de Engenharia Electromecânica, Universidade da Beira Interior,  
Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã, Portugal*

Received 18 March 2003; received in revised form 6 October 2003; accepted 13 October 2003

## Abstract

A multiresolution adaptive approach for the solution of two-dimensional partial differential equations (PDEs) is presented. This methodology is a multidimensional extension of that presented in a previous work [Comput. Methods Appl. Mech. Engrg. 191 (2002) 3909]. The method proposed is unconditionally stable, by incorporating convection differencing schemes with the TVD property, and the grid is dynamically adapted so that higher spatial resolution is automatically allocated to domain regions where strong gradients are observed. The two desired properties of any PDE solver, stability and accuracy, are therefore retained. Numerical results for four test problems are presented which serve to demonstrate the robustness and cost effectiveness of the method.

© 2003 Elsevier B.V. All rights reserved.

*Keywords:* Adaptive grid; CUBISTA high-resolution scheme; Multiresolution; Hyperbolic equations; Wavelet

## 1. Introduction

Solution of partial differential equations (PDEs) is a key ingredient in many problems of physics and engineering and so it is a subject of active research in numerical analysis. In a previous work by our group [1], we have introduced a multiresolution strategy coupled with high-order bounded difference schemes, which was shown to be very effective in solving one-dimensional problems involving hyperbolic PDEs with or without sharp fronts. This forms the basis for the present contribution, in which a general class of 2D non-linear advection–diffusion–reaction equations of the form

\* Corresponding author. Fax: +351-22-508-1449.

E-mail address: [mmalves@fe.up.pt](mailto:mmalves@fe.up.pt) (M.A. Alves).

$$\frac{\partial u}{\partial t} + \frac{\partial F(x, y, t, u)}{\partial x} + \frac{\partial G(x, y, t, u)}{\partial y} = v \left\{ \frac{\partial}{\partial x} \left[ Q(u) \frac{\partial u}{\partial x} \right] + \frac{\partial}{\partial y} \left[ R(u) \frac{\partial u}{\partial y} \right] \right\} + S(x, y, t, u) \quad (1)$$

with initial values  $u(t = 0, x, y) = u_0(x, y)$  is considered. The main focus will be on the hyperbolic equations that occur when Eq. (1) is dominated by advection ( $v \rightarrow 0$ ). These are the typical equations governing transport phenomena at high Reynolds (or Peclet) numbers, which are known to be difficult to solve numerically in many cases.

The methodology we propose requires the use of high-resolution schemes combined with an adaptive mesh technique inspired by wavelet theory. In this approach, higher-order accuracy is combined with a multiresolution mesh refinement technique in order to achieve a nearly constant discretization error throughout the computational domain, thus reducing the memory and CPU requirements of the solver, while keeping the same level of accuracy with respect to the equivalent uniform mesh approach. A new high-resolution scheme is also explained.

Since an introduction to the proposed multiresolution methodology was already given in our previous paper [1] we may condense here the background points. The basic idea of multiresolution analysis was put forward by Mallat [2] in the context of orthonormal wavelets. Later, Harten [3] has set a general framework for multiresolution representation, with view to reducing the number of flux computations when expensive methods, such as ENO schemes, are employed (see also [4]). There are other approaches similar to Harten's methodology such as the ones proposed by Cohen et al. [5]. Karni et al. [6] proposes a smoothness indicator for the solution of hyperbolic conservation laws. This indicator is a practical tool to identify where grid adaptation is needed.

Harten's methodology produces moderate CPU reductions but it is limited because, for each time step, the solution is still represented on the finest grid. A better strategy is to couple dynamical mesh adaptation with multiresolution and deal specifically with the restricted solution domain where sharp gradients are present. Then it is advantageous to store only the significant information and perform the time integration in this reduced set of grid points. There is, however, the need to dynamically adapt the mesh as the solution evolves but, as will be demonstrated, this overhead is not significant and globally there are significant memory and CPU time savings as compared to the corresponding fixed grid approach.

This paper is organized as follows: first, high-resolution schemes are broadly described and a new scheme, called CUBISTA [7], is introduced. Then, the multiresolution representation of data in 2D is explained together with interpolation procedures, followed by a description of the grid adaptation strategy and the way to determine spatial derivatives in the adapted grid. The overall methodology is applied to the solution of four test cases, which include hyperbolic PDEs with very strong gradients and the classical shock-tube problem of gas dynamics. The paper ends with a summary of the main conclusions.

## 2. The high-resolution scheme

A key point in the discretization of Eq. (1) is the treatment of the convection term, in order to prevent the appearance and subsequent intensification of unnatural oscillations. For this purpose, high-resolution schemes are used, as explained by Leonard in two excellent review articles on this matter [8,9]. Such schemes require normalizing the relevant advected variables and prescribing an interpolation procedure in a normalized variable diagram (the NVD). Since we have described this matter in [1], we shall give here the minimum details to make this section easy to follow. For each direction  $x$  and  $y$  we must identify upstream (U) and downstream (D) nodes relative to the central grid point (P), as represented in Fig. 1, so that the normalized cell-face value of the convected flux,  $\phi$  ( $\equiv F$  or  $G$  depending on the direction being evaluated) is given by [10]

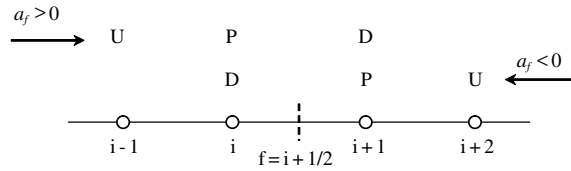


Fig. 1. Definition of local variables.

$$\hat{\phi} = \frac{\phi - \phi_U}{\phi_D - \phi_U}. \tag{2}$$

Definition of upwind and downwind directions is decided upon the orientation of the local cell-face velocity, which is defined as  $\vec{a} = d\vec{f}/d\vec{x}$ , for any flux vector  $\vec{f}$  (in relation to Eq. (1), this gives the components  $a_x = \partial F/\partial x$  and  $a_y = \partial G/\partial y$ ).

Leonard [8] has shown that, for an accuracy of up to third order on a uniform mesh, the cell-face value of the convective flux  $\phi$  is determined from a functional

$$\hat{\phi}_f = f(\hat{\phi}_p), \tag{3}$$

where the specification of  $f(\hat{\phi}_p)$  in the NVD, by means of segmented functions, identifies the many existing high-resolution schemes.

Fig. 2 illustrates the graphs of some schemes relevant to the present work. Harten [11] proposed the MINMOD scheme, a combination of second-order upwind and central differences, which Alves et al. [7] have shown to lead to iterative convergence problems. These authors have then developed an improved scheme, called CUBISTA, that exhibits better iterative properties (recall that the governing equations in fluid flow problems are solved iteratively) and is represented in Fig. 2. The high-resolution schemes in this figure are defined as

$$\hat{\phi}_f = \max[\hat{\phi}_p, \min(1.5\hat{\phi}_p, 0.5\hat{\phi}_p + 0.5)] \tag{4}$$

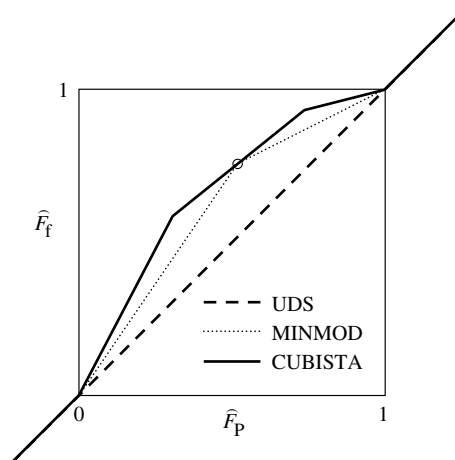


Fig. 2. NVD representation of the MINMOD and CUBISTA high-resolution schemes.

for MINMOD, and

$$\widehat{\phi}_f = \max[\widehat{\phi}_p, \min(1.75\widehat{\phi}_p, 0.75\widehat{\phi}_p + 0.375, 0.25\widehat{\phi}_p + 0.75)] \tag{5}$$

for CUBISTA. We note that these equations are written for uniform meshes and may be easily re-interpreted in terms of non-uniform meshes (see [1,7]) following the Normalised Variable and Space Formulation (NVSF) of [10].

Once the high-resolution convection scheme is specified, the task of computing the cell-face value of the dependent variable is straightforward, and is described in detail in [1].

### 3. Multiresolution representation of data

As explained in the Introduction, the idea of this work is to extend to two-dimensional grids the approach followed in Alves et al. [1].

Consider a set of dyadic grids of the form

$$V^j = \{(x_k^j, y_l^j) \in \mathbb{R}^2 : x_k^j = 2^{-j}k, y_l^j = 2^{-j}l\}, \quad j, k, l \in \mathbb{Z}, \tag{6}$$

where  $j$  identifies the resolution level,  $k$  the spatial location for the  $x$ -direction and  $l$  the spatial location for the  $y$ -direction, as illustrated in Fig. 3. Both the  $x$  and  $y$  dimensions of the computational domain have been normalized, so that the corresponding co-ordinates vary between 0 and 1.

Assume that the solution is known on grid  $V^j$  and that we want to extend it to the next finer grid  $V^{j+1}$ , which has four times more points than the previous grid (note that in 1D, the factor is only 2). For the even-numbered grid points, in both directions of grid  $V^{j+1}$ , we already know their values, which are given by

$$u_{2k,2l}^{j+1} = u_{k,l}^j. \tag{7}$$

The values in the odd-numbered grid points in  $V^{j+1}$  for  $x$ -direction,  $u(x_{2k+1}^{j+1}, y_l^j)$ ,  $y$ -direction,  $u(x_k^j, y_{2l+1}^{j+1})$  or both  $u(x_{2k+1}^{j+1}, y_{2l+1}^{j+1})$  are computed by interpolating the known even-numbered grid points in the same direction. Then, for a certain number of time steps, the solution on a given mesh  $(x_k, y_l)$  is advanced in time with the solver specified in Section 4, until a new mesh adaptation step is reached (see [1]). Mesh points will

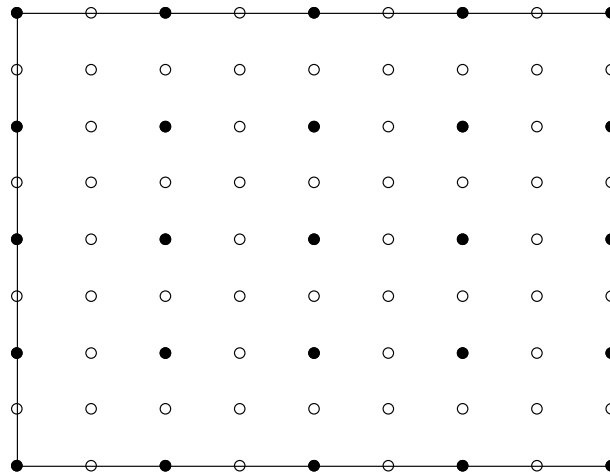


Fig. 3. Example of points in a dyadic grid: (●)  $V^2$  and (○)  $V^3$ .

be added in zones where the solution changes abruptly, according to interpolative error coefficients defined by the difference between the existing solution value  $u_{k,l}$  and the value obtained by interpolation from neighbour nodal points,  $I(u_{k,l})$  (it is noted that interpolation is only required in odd-numbered points). In two dimensions 4 types of interpolation are possible, in contrast with the single one for the 1D case of the previous work [1]. The interpolative error coefficients for the  $j$ th resolution level are obtained by the following steps:

1. For odd-numbered grid points in  $x$ -direction and even-numbered in  $y$ -direction:

$$d_{1,k,l}^j = |u(x_{2k+1}^{j+1}, y_l^j) - I_x^j[u(x_{2k+1}^{j+1}, y_l^j)]| / u_{\text{ref}}, \quad k, l = 0 \text{ to } 2^j. \quad (8)$$

Here  $I_x^j$  is the interpolating operator, based on function values along the  $x$ -direction for constant  $y$ -coordinate:  $u(x_{k-1}^j, y_l^j)$ ,  $u(x_k^j, y_l^j)$ ,  $u(x_{k+1}^j, y_l^j)$  and  $u(x_{k+2}^j, y_l^j)$  (see Fig. 4(a)). We suggest the use of the known maximum value of the dependent variable for the reference value of the dependent variable,  $u_{\text{ref}}$ .

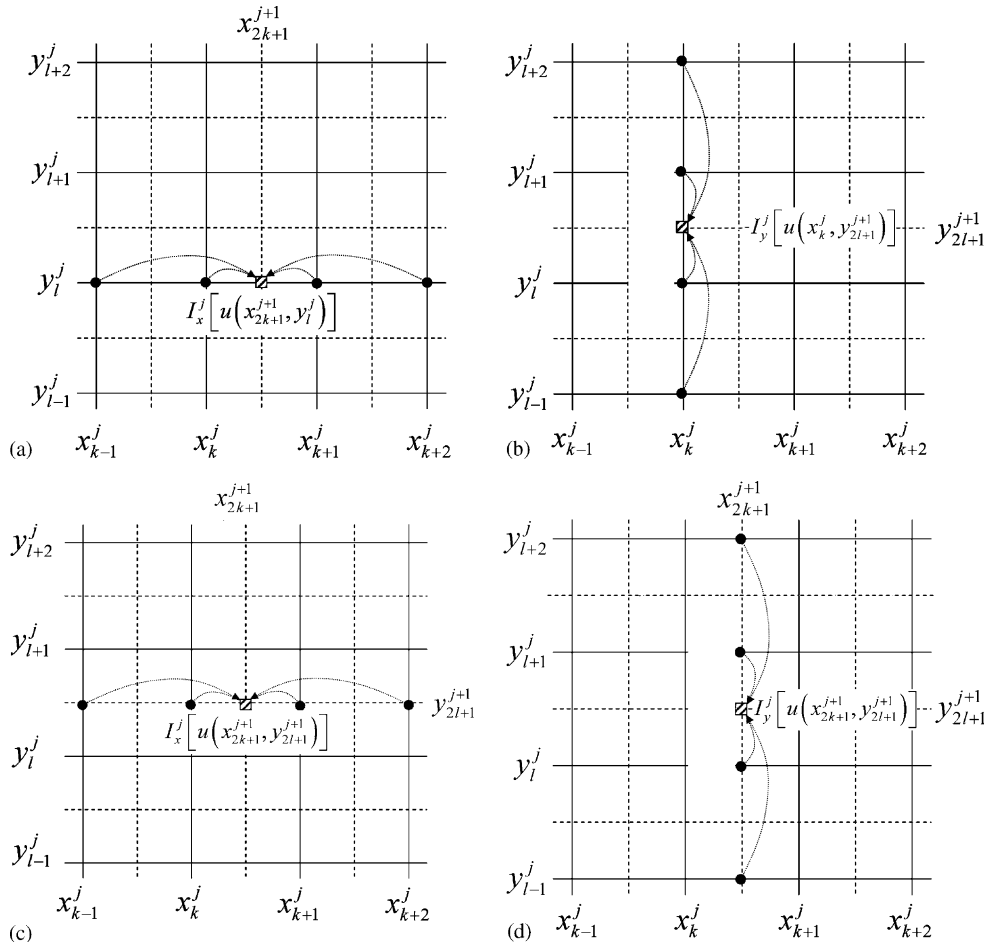


Fig. 4. Illustration of the grid points used by the interpolating operators: (a)  $I_x^j[u(x_{2k+1}^{j+1}, y_l^j)]$ , (b)  $I_y^j[u(x_k^j, y_{2l+1}^{j+1})]$ , (c)  $I_x^j[u(x_{2k+1}^{j+1}, y_{2l+1}^{j+1})]$  and (d)  $I_y^j[u(x_{2k+1}^{j+1}, y_{2l+1}^{j+1})]$ .

2. For odd-numbered grid points in  $y$ -direction and even-numbered in  $x$ -direction:

$$d_{2,k,l}^j = \left| u(x_k^j, y_{2l+1}^{j+1}) - I_y^j [u(x_k^j, y_{2l+1}^{j+1})] \right| / u_{\text{ref}}, \quad k, l = 0 \text{ to } 2^j, \quad (9)$$

where  $I_y^j$  is the interpolating operator based on the function values along the  $y$ -direction, for constant  $x$ -coordinate:  $u(x_k^j, y_{l-1}^j)$ ,  $u(x_k^j, y_l^j)$ ,  $u(x_k^j, y_{l+1}^j)$  and  $u(x_k^j, y_{l+2}^j)$  (see Fig. 4(b)).

Finally, for odd-numbered grid points in both directions:

$$3. \quad d_{3,k,l}^j = \left| u(x_{2k+1}^{j+1}, y_{2l+1}^{j+1}) - I_x^j [u(x_{2k+1}^{j+1}, y_{2l+1}^{j+1})] \right| / u_{\text{ref}}, \quad k, l = 0 \text{ to } 2^j, \quad (10)$$

$$4. \quad d_{4,k,l}^j = \left| u(x_{2k+1}^{j+1}, y_{2l+1}^{j+1}) - I_y^j [u(x_{2k+1}^{j+1}, y_{2l+1}^{j+1})] \right| / u_{\text{ref}}, \quad k, l = 0 \text{ to } 2^j, \quad (11)$$

where  $d_{3,k,l}^j$  is the interpolative error coefficient based in the  $I_x^j$  operator (see Fig. 4(c)) and  $d_{4,k,l}^j$  the interpolative error coefficient based in the  $I_y^j$  operator (see Fig. 4(d)). The interpolation designated by functions  $I_x$  and  $I_y$  above is not a simple arithmetic averaging but rather is obtained with the practice explained in the next paragraphs.

A common practice in wavelet-based methods is to utilize symmetric interpolating polynomials [12,13], but we found that it is important to retain consistency with the discretization used for the convection term. Therefore we propose using the high-resolution schemes of the previous section for interpolation.

The procedure followed consists on the following steps, where the calculation of the interpolated values is illustrated for the case of odd numbered grid points in  $x$ -direction and even-numbered in  $y$ -direction.

(i) Calculate the face velocity,  $a_f$ , to identify the local convective flux direction:

$$a_f = (a_{k,l}^j + a_{k+1,l}^j) / 2. \quad (12)$$

(ii) Calculate the normalised face value of the advected variable,  $\widehat{u}_f$ , using the CUBISTA high-resolution scheme:

$$\widehat{u}_f = \begin{cases} \max \left[ \frac{u_{k,l}^j - u_{k-1,l}^j}{u_{k+1,l}^j - u_{k-1,l}^j}, \min \left( \frac{7}{4} \frac{u_{k,l}^j - u_{k-1,l}^j}{u_{k+1,l}^j - u_{k-1,l}^j}, \frac{3}{4} \frac{u_{k,l}^j - u_{k-1,l}^j}{u_{k+1,l}^j - u_{k-1,l}^j} + \frac{3}{8}, \frac{3}{4} + \frac{1}{4} \frac{u_{k,l}^j - u_{k-1,l}^j}{u_{k+1,l}^j - u_{k-1,l}^j} \right) \right] & \text{if } a_f \geq 0, \\ \max \left[ \frac{u_{k+1,l}^j - u_{k+2,l}^j}{u_{k,l}^j - u_{k+2,l}^j}, \min \left( \frac{7}{4} \frac{u_{k+1,l}^j - u_{k+2,l}^j}{u_{k,l}^j - u_{k+2,l}^j}, \frac{3}{4} \frac{u_{k+1,l}^j - u_{k+2,l}^j}{u_{k,l}^j - u_{k+2,l}^j} + \frac{3}{8}, \frac{3}{4} + \frac{1}{4} \frac{u_{k+1,l}^j - u_{k+2,l}^j}{u_{k,l}^j - u_{k+2,l}^j} \right) \right] & \text{if } a_f < 0. \end{cases} \quad (13)$$

(iii) Calculate the interpolated value:

$$u_f = I_x^j [u(x_{2k+1}^{j+1}, y_l^j)] = \begin{cases} u_{k-1,l}^j + \widehat{u}_f (u_{k+1,l}^j - u_{k-1,l}^j) & \text{if } a_f \geq 0, \\ u_{k+2,l}^j + \widehat{u}_f (u_{k,l}^j - u_{k+2,l}^j) & \text{if } a_f < 0. \end{cases} \quad (14)$$

The interpolative error coefficients in a given direction are a measure of the local ‘‘irregular’’ behaviour of the analysed function, in that direction. If the absolute value of  $d_{1,k,l}^j$  or  $d_{2,k,l}^j$ , is below a given (small) threshold  $\varepsilon$ , then the corresponding grid point  $(x_{2k+1}^{j+1}, y_l^j)$  or  $(x_k^j, y_{2l+1}^{j+1})$  is superfluous for accurate representation of the original function, and so those values can be rejected without loss of significant information since the function can be reconstructed from the preserved information on the next coarser grid,  $V^j$ . However, in order to reject the point  $(x_{2k+1}^{j+1}, y_{2l+1}^{j+1})$  we note that both coefficients  $d_{3,k,l}^j$  and  $d_{4,k,l}^j$  must be below the given threshold  $\varepsilon$ .

A function that varies abruptly only in a narrow area of the domain will have most of the  $d_{i,k,l}^j$  coefficients close to zero and so the information can be compressed with great efficiency, and without loss of accuracy.

2D multiresolution approaches have been extensively studied in other contexts such as image compression, and the interested reader is referred to the work of Zhou [14].

Finally, it should be noted that the multiresolution approach must select the relevant grid points in a 2D tree structure. For example, if the point  $(x_{2k+1}^{j+1}, y_{2l+1}^{j+1})$  is retained in the adaptation algorithm, then the grid points in the  $x$ -direction  $(x_{i=k-m+1,k+m}^j, y_{2l+1}^{j+1})$  and  $y$ -direction  $(x_{2k+1}^{j+1}, y_{i=l-m+1,l+m}^j)$  with  $m = -1$  to  $2$  must be preserved, in order to be possible to reconstruct the function value in the next adaptation step, and evaluate  $d_{3,k,l}^j$  or  $d_{4,k,l}^j$ .

In our algorithm, the user must specify the maximum resolution level,  $J_{\max}$ , so that grid coalescence in problematic regions is avoided. The user also supplies the minimum level of resolution,  $J_{\min}$ , and all the grid points pertaining to this level of resolution are always conserved throughout the computations.

#### 4. Grid adaptation and temporal integration

The PDEs are advanced in time with LSODA [15], a public domain solver for first-order ordinary differential equation (ODE); the same solver was employed in our previous work [1]. This is a quite versatile routine which uses either Adams’s method with an accuracy of up to 12th order, or Gear’s method for the case of stiff problems, allowing for variable time steps and guaranteeing highly accurate solutions for the time evolution. The inaccuracies in the computed solution are therefore mainly controlled by the spatial discretization, and so are related to the high-resolution schemes of Section 2.

During the solution of a PDE the grid should be continuously adapted, so that it can automatically adjust to reflect modifications in time of the solution.

##### 4.1. Strategy

The grid adaptation strategy that is proposed for the solution of PDEs is now described. Given discrete function values  $u(x, y)$  at time  $t = t_1$ , proceed as follows:

1. Starting in  $j = J_{\min}$ .
2. Compute the interpolative error coefficient for odd-numbered grid points in the  $x$ -direction and even in  $y$ -direction,  $d_{1,k,l}^j$  (see Eq. (8) and Fig. 5), for  $k = 0$  to  $2^j - 1$  and  $l = 0$  to  $2^j$ .
3. Compute the interpolative error coefficient for odd-numbered grid points in the  $y$ -direction and even in  $x$ -direction,  $d_{2,k,l}^j$  (see Eq. (9)), for  $k = 0$  to  $2^j$  and  $l = 0$  to  $2^j - 1$ .
4. Compute the interpolative error coefficient for odd-numbered grid points in the  $x$ - and  $y$ -directions,  $d_{3,k,l}^j$  (see Eq. (10)) and  $d_{4,k,l}^j$  (see Eq. (11)) for  $k, l = 0$  to  $2^j - 1$ .

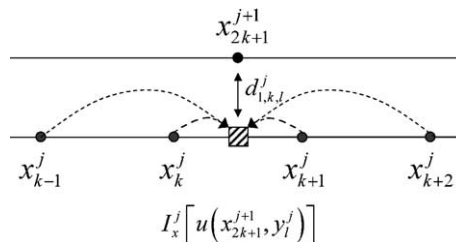


Fig. 5. Illustration of the calculation of the interpolating error coefficient,  $d_{1,k,l}^j$ .

5. Increase  $j$  by one, and repeat steps 2–4 until  $j = J_{\max} - 1$ .
6. Identify the interpolative error coefficients,  $d_{1,k,l}^j$ , that fall above the predefined threshold  $\varepsilon$ . The grid points  $(x_{2(k+i)+1}^{j+1}, y_{2l}^{j+1})$  with  $i = -N_L, N_R^{(1)}$  and  $(x_{2(2k+m)+1}^{j+2}, y_{2l}^{j+1})$  with  $m = -N_{LU} + 1, N_{RU}^{(2)}$  are included in an indicator. Variables marked with a superscript (1) and (2) are defined and discussed below.
7. Identify the interpolative error coefficient,  $d_{2,k,l}^j$ , that fall above the predefined threshold  $\varepsilon$ . The grid points  $(x_{2k}^{j+1}, y_{2(l+i)+1}^{j+1})$ ,  $i = -N_L, N_R^{(1)}$  and  $(x_{2k}^{j+1}, y_{2(2l+m)+1}^{j+2})$  with  $m = -N_{LU} + 1, N_{RU}^{(2)}$  are included in an indicator.
8. Identify the interpolative error coefficient,  $d_{3,k,l}^j$  that fall above the predefined threshold  $\varepsilon$ . The grid points  $(x_{2(k+i)+1}^{j+1}, y_{2l+1}^{j+1})$  with  $i = -N_L, N_R^{(1)}$  and  $(x_{2(2k+m)+1}^{j+2}, y_{2l+1}^{j+1})$  with  $m = -N_{LU} + 1, N_{RU}^{(2)}$  are included in an indicator.
9. Identify the interpolative error coefficient,  $d_{4,k,l}^j$ , that fall above the predefined threshold  $\varepsilon$ . The grid points  $(x_{2k+1}^{j+1}, y_{2(l+i)+1}^{j+1})$  with  $i = -N_L, N_R^{(1)}$  and  $(x_{2k+1}^{j+1}, y_{2(2l+m)+1}^{j+2})$  with  $m = -N_{LU} + 1, N_{RU}^{(2)}$  are included in an indicator.
10. Add to the indicator all the grid points associated to the lower resolution level,  $J_{\min}$ . These are the “basic” grid points that are always maintained throughout the integration.
11. Remove all the columns and rows that are not pertinent for the function representation in the opposite direction. This procedure reduces significantly the number of grid points without loss of precision, because the function can be reconstruct with the values in the opposite direction.
12. Beginning at resolution level  $j = J_{\max} - 1$ , recursively extend the indicator, so that all the grid points necessary for the calculation of the existing  $j$ th level interpolative error coefficients are included. This is necessary in order to obtain a 2D tree-like structure (see Fig. 5 in [1]).

In the above steps, the following definitions are utilized:

- (1)  $N_L$  and  $N_R$  denote the grid points belonging to the same resolution level, in both directions, and represent the number of grid points added to the left (below) and to the right (above) of the grid point which does not satisfy the interpolative error tolerance. The purpose of storing this information is to account for possible translation of the sharp features of the solution during the next time integration step.
- (2) Similarly,  $N_{LU}$  and  $N_{RU}$  are the number of points added in the resolution level immediately above the present one, respectively to the left (below) and to the right (above) of the point where the interpolative error tolerance is not satisfied.  $N_{LU}$  must be less or equal to  $N_L$  and  $N_{RU}$  must be less or equal to  $N_R$ . This accounts for the possibility of the solution becoming “steeper” in this region, at later time steps, and thus avoiding excessive work with frequent mesh adaptation steps.

For solution of a system of partial differential equations, the previous procedure must be modified in order to reflect the behavior of the solutions of all equations involved. In this case the adaptation strategy is performed for all the equations of the system for each adaptation step. The resulting indicator arrays contain the points needed for each equation. The points that will be used for the solution of all the equations of the system will be the union of the points needed by each equation. The strategy described above has demonstrated to be efficient for the solution of problems involving a single or a system of partial differential equations.

#### 4.2. Calculation of spatial derivatives in the adapted grid

Here we have two approaches, which are listed and briefly discussed hereafter. Either the spatial derivatives are calculated directly in the adapted non-uniform grid, exactly as done in our previous work [1]



or, as a second alternative, the solution is interpolated to the maximum resolution level and the calculations of the spatial derivatives are performed in that final uniform grid, as in [12]. The second approach is not recommended as it involves too many unnecessary interpolations in the finest grid, in order to obtain the values needed to calculate the spatial derivatives, thus leading to a tremendous slow down in the integration process.

## 5. Application examples

In this section we present the results obtained in four test cases, designed to assess the accuracy and robustness of the proposed strategy under different two-dimensional situations. We begin by solving the classical linear advection equation. Then we solve the Smith–Hutton problem, which is a linear advection–diffusion problem with a given velocity field, to show the ability of the method to deal with a spatially varying velocity field with curved streamlines. In the third test case we solve the classical Burgers’ equation, with and without viscosity, the prototypical non-linear transport equation. The final example is more complex and consists on the solution of the system of Euler equations for gas dynamics: a coupled set of four equations for the two Cartesian velocity components, density and pressure.

All the calculations were performed in a 1.5 GHz Intel Pentium IV<sup>®</sup> personal computer with 1 GB of RIMM memory.

### 5.1. Linear advection equation

This example considers the linear advection equation

$$\frac{\partial u}{\partial t} = -a \left( \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \right) \quad (15)$$

where, for convenience, the value of  $a = 1$  was adopted. The initial and boundary conditions correspond to a classical benchmark problem of 2D-advection, and are given by

$$u(x, y, 0) = 0,$$

$$u(0, y, t) = 1, \quad u(x, 0, t) = 0.$$

The present test case has an exact solution represented by the transport, in space and time, of the unit step  $u = 1$  applied at the left boundary ( $x = 0$ ), by the given velocity field, with equal unitary components along the  $x$  and  $y$  co-ordinates. It thus offers a convenient means of assessing the performance of the proposed adaptation strategy. A straightforward application of an unbounded discretization scheme, such as central differencing or cubic splines, always conduces to the appearance of unphysical oscillations in the solution, even with an extremely refined mesh, since the equation does not have a diffusive term. A bounded method, such as the CUBISTA or MINMOD schemes presented in Section 2, introduces some artificial diffusion in the vicinity of the step, if the mesh is not fine enough. Therefore an adaptive procedure is needed in order to save memory and computational effort.

In this example there are, initially (for  $t \leq 1$ ), two abrupt fronts: a step moving along the  $x$ -direction with a velocity of  $a = 1$ , and another moving along the  $x = y$ -direction. For  $t \geq 1$  there remains only one of the fronts (along the main diagonal,  $x = y$ ). This means that for most of the integration time there are only two localised regions where the mesh must be refined: one where the adaptation algorithm adds points only in the  $x$ -direction (this happens for the step moving along the  $x$ -direction); and a region where addition of points in both the  $x$ - and  $y$ -directions are needed (this happens with the front in  $x = y$ ).

In this example the convection term was discretized with the MINMOD and CUBISTA high-resolution schemes in the adapted non-uniform grid. Furthermore, it should be clear that, as in our previous work [1],

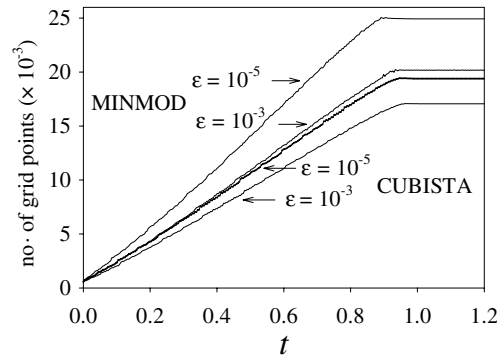


Fig. 6. Number of grid points used by the adaptation algorithm for two different threshold values ( $\varepsilon = 10^{-3}$  and  $10^{-5}$ ).

there is no need to reconstruct any values from lower levels of resolution (a very time-consuming process), except during the grid adaptation stage of the algorithm.

Fig. 6 shows the time evolution of the number of grid points used by the adaptation algorithm, when the MINMOD and CUBISTA high-resolution schemes are employed for two different threshold values ( $\varepsilon = 10^{-3}$  and  $10^{-5}$ ). The minimum and maximum levels of resolution adopted here, and in the remaining simulations of this work, were  $J_{\min} = 3$  and  $J_{\max} = 8$ , respectively, and the parameters used in the adaptation algorithm were  $N_R = N_L = 2$  and  $N_{RU} = N_{LU} = 1$ .

As mentioned in Section 3, the threshold parameter  $\varepsilon$  is a direct measure of the error involved in the approximation of the solution by a reduced set of grid points. Higher  $\varepsilon$  values conduce to fewer equations to integrate but lower accuracy. When the threshold value is decreased one hundred times, the average number of grid points increases 1.14 and 1.24 times for the CUBISTA and MINMOD high-resolution schemes, respectively. The solution obtained with the multiresolution approach corresponds to solving Eq. (15) in a fix grid of 66 049 ( $\equiv (2^8 + 1)^2$ ) mesh points. This means that at the end of the simulation ( $t = 1.2$ ), when the number of grid points attains its maximum value (see Fig. 6), the compression ratio (percentage of grid points rejected by the adaptive algorithm) obtained with the CUBISTA scheme is about 74% for  $\varepsilon = 10^{-3}$  and 71% for  $\varepsilon = 10^{-5}$ , while for the MINMOD scheme is about 69% for  $\varepsilon = 10^{-3}$  and 62% for  $\varepsilon = 10^{-5}$ . It is noted that these are minimum compression ratios during the whole integration time. The highest compression ratio is obtained at  $t = 0$  and is about 99%.

From these considerations it appears that a threshold tolerance of  $\varepsilon = 10^{-3}$  offers an adequate balance between accuracy and efficiency. It is also clear that the multiresolution approach yields very efficient variation of the number of grid points along the integration process, a fact of extreme importance and which results from using only the grid points that are necessary to attain a given precision. In this sense, the present method is more efficient and versatile than methods that use a constant number of grid points, even in the case of moving mesh methods.

The numerical solution obtained with  $\varepsilon = 10^{-3}$  (the value used in the remaining simulations) is presented in Fig. 7(a) at  $t = 0.5$  for the CUBISTA high-resolution scheme. As it can be seen the adaptation algorithm is performing well and the two fronts discussed above are resolved with high accuracy. The corresponding distribution of the grid points is represented in Fig. 7(b). The smooth regions of the solution, in either direction, are resolved by the coarse resolution levels, while the higher levels are required only near the step locations, thus illustrating the efficient data compression of the adaptive strategy. For this simple constant velocity problem, the location of the higher grid density regions could be easily predicted, but the proposed method does not need such information and works in a self-sufficient dynamic way. In Fig. 7(c) the solution obtained with the MINMOD scheme is presented, and the corresponding distribution of grid points is

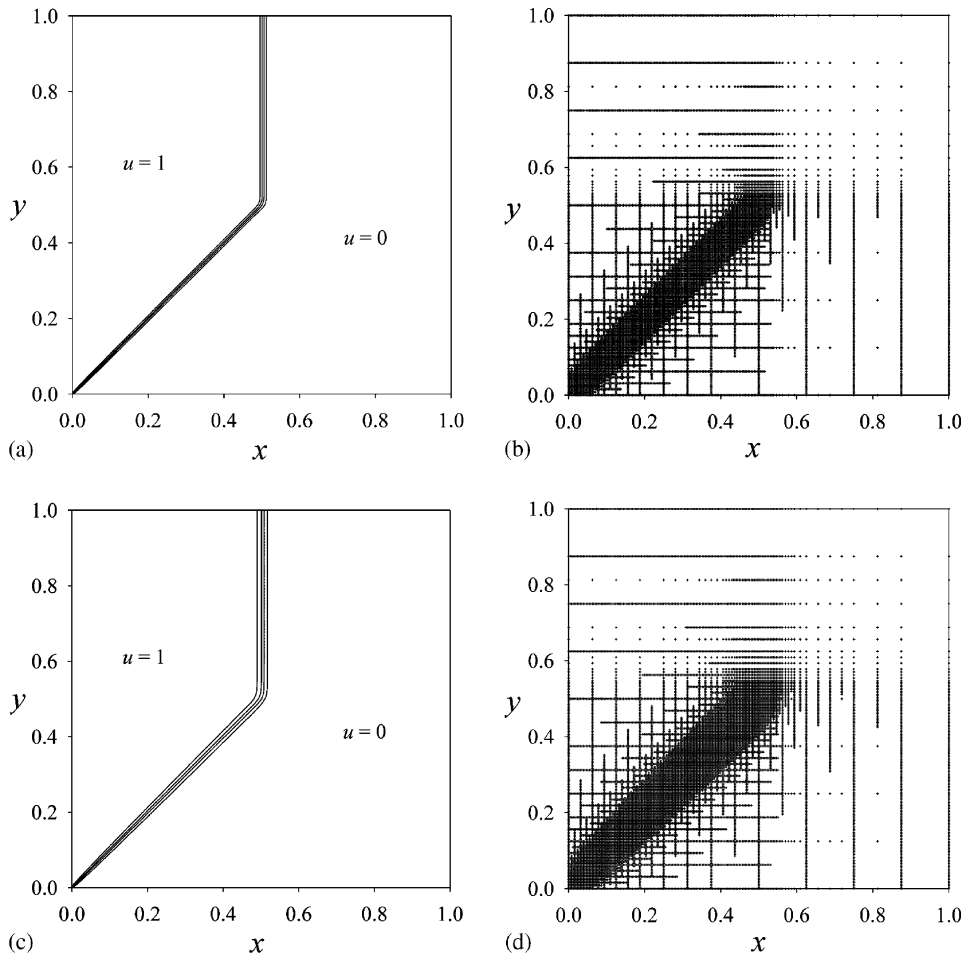


Fig. 7. (a) Solution at  $t = 0.5$  of the linear advection equation with a step profile for the CUBISTA scheme and (b) corresponding distribution of grid points in terms of spatial location. (c) Solution for the MINMOD scheme and (d) corresponding distribution of grid points in terms of spatial location.

illustrated in Fig. 7(d). As can be observed, the solution computed with the MINMOD scheme is less accurate despite using a higher number of grid points. This illustrates the superiority of the CUBISTA scheme in terms of accuracy.

The same problem was also solved on equivalent uniform meshes having  $(2^{J_{\max}} + 1)^2$  grid points, with the aim of calculating the increase rate of grid points in the adaptive approach, the CPU time and the speed-up factors, defined as the ratio of CPU times for the uniform mesh and the adaptive calculations. Here  $J_{\max}$  was varied from 3 up to 10.

The increase rate of grid points (IRGP), defined as the ratio between the number of grid points used in two consecutive resolution levels for the adaptive approach, is presented in Table 1. In a uniform mesh this number tends to 4 while in the adaptive approach tends to 2.3 for both the MINMOD and CUBISTA schemes, as it can be seen in Table 1. This means that, for high levels of resolution, when passing from one level to the next one, the CPU time should increase 4.6 times ( $= 2.3 \times 2$ , where the factor 2 results from the necessity of halving the time step, even for a variable time-step solver such as LSODA) using the adaptation

Table 1

Number of grid points (NGP), compression ratio (CR) and increase rate of grid points (IRGP) for the linear advection equation, using the adaptive approach

$J_{\max}$	Uniform mesh	Adaptive mesh					
		MINMOD			CUBISTA		
	NGP	NGP	IRGP	CR (%)	NGP	IRGP	CR (%)
3	81	81	–	0.0	81	–	0.0
4	289	234	2.89	19.0	215	2.65	25.6
5	1089	549	2.35	49.6	486	2.26	55.4
6	4225	1172	2.13	72.3	1017	2.09	75.9
7	16 641	2524	2.15	84.8	2179	2.14	86.9
8	66 049	5547	2.20	91.6	4 754	2.18	92.8
9	263 169	12 456	2.25	95.3	10 589	2.23	96.0
10	1 050 625	28 380	2.28	97.3	23 710	2.24	97.7

The values correspond to the average from  $t = 0$  to  $t = 0.5$ .

Table 2

Speed-up factors and CPU times for the linear advection equation ( $t = 0.5$ ) calculated with the CUBISTA scheme

$J_{\max}$	Uniform mesh		Adaptive mesh		
	NGP <sup>a</sup>	CPU (s)	NGP <sup>a</sup>	CPU (s)	Speed-up
3	81	0.48	81	0.48	1.00
4	289	2.74	283	2.24	1.22
5	1089	16.99	732	9.12	1.86
6	4225	117.78	1725	41.76	2.82
7	16 641	855.42	4036	204.92	4.17

<sup>a</sup> Number of grid points. For the adaptive strategy, values are the average from  $t = 0$  to 0.5.

algorithm and 8 times, using a uniform mesh. From this analysis, the maximum theoretical rate of increase of the speed-up factors that can be obtained with the present approach is about 1.7 ( $= 8/4.6$ ). It is interesting to note that the IRGP tends to the same value, as  $J_{\max}$  increases, irrespective of being calculated based on the instantaneous number of grid points, or the average value from  $t = 0$  (not shown in Table 1).

Table 2 presents the speed-up factors and CPU times for uniform and adaptive mesh approaches. The CPU times were calculated only for resolution levels up to seven, because, for higher values, the calculations with the uniform mesh approach required more physical computer memory than available. In this situation, *virtual* memory would be needed and that would distort the CPU time comparison.

The calculated speed-up factors increase exponentially with mesh refinement (almost at the rate of 1.7, as discussed before), and can be as high as we want.

From the previous tests we found that the CUBISTA scheme is more accurate and the speed-up factors are higher than those for the MINMOD scheme (due to the use of less grid points to attain a given precision). This is why, in the following examples, the CUBISTA scheme was selected for all the calculations.

## 5.2. Smith–Hutton problem

This example considers the Smith–Hutton test problem, described by the following convection–diffusion equation [16]:

$$\frac{\partial T}{\partial t} = \frac{1}{Pe} \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) - \left( \frac{\partial(a_x T)}{\partial x} + \frac{\partial(a_y T)}{\partial y} \right), \quad (16)$$

where  $Pe$  is the Peclet number,  $T$  is the temperature (the unknown variable in this case) and  $a_x$  and  $a_y$  are the velocity components in the  $x$ - and  $y$ -directions respectively. A difference with the previous test case is that now the velocity field is non-uniform, varying from point to point, and corresponds to curved streamlines with Cartesian velocity components defined as

$$a_x = 2y(1 - x^2), \quad a_y = -2x(1 - y^2). \tag{17}$$

The problem was solved for two values of the Peclet number, which measures the strength of convection in relation to diffusion, namely  $Pe = 500$  (moderate convection) and  $Pe \rightarrow \infty$  (convection dominated flow). The solution domain is  $-1 \leq x \leq 1$  and  $0 \leq y \leq 1$ , and the following initial and boundary conditions were applied, as used by Darwish and Moukalled in [17]:

$$\begin{aligned} T(x, y, 0) &= 0, \\ T(-1, y, t) &= 0, \quad T(1, y, t) = 0, \\ T(x, 1, t) &= 0, \quad T(x, 0, t) = \begin{cases} 1 & -0.5 \leq x \leq 0, \\ 0 & \text{elsewhere.} \end{cases} \end{aligned}$$

Physically, the problem can be described as follows: the temperature is initially zero everywhere and a step profile is imposed along the negative side of the  $x$ -axis ( $T = 0, x < -1/2; T = 1, x \geq -1/2$ ), this “step” is then convected by the circular velocity field, from the left quadrant ( $x \leq 0$ ) to the right one ( $x > 0$ ). The computed solution with the CUBISTA high-resolution scheme is presented in Fig. 8(a) at  $t = \pi/4$ , for  $Pe \rightarrow \infty$ . Similarly to the previous example, the use of an unbounded discretization scheme always leads to the appearance of unphysical oscillations in the solution, since the equation does not have a diffusive term (where  $Pe \rightarrow \infty$ ). The application of a high-resolution scheme is therefore imposed. The distribution of the grid points is represented in Fig. 8(b). Note that the smooth regions of the solution are resolved by a low density of grid points while high density is only required near the front location, thus illustrating the efficient data compression of the adaptive strategy.

Fig. 9(a) shows the solution at the same simulation time,  $t = \pi/4$ , for the case when some diffusion is present,  $Pe = 500$ . The purpose of this test case is to demonstrate the ability of the proposed method to deal with problems having a diffusive term. An unbounded discretization scheme could, in principle, be used, but in order to avoid the appearance of unphysical oscillations, the mesh would have to be extremely refined in the vicinity of the sharp, curved fronts. Our algorithm is however more general and can be used in both situations, as can be seen by the correct location of grid points in Fig. 9(b).

The number of grid points used by the adaptation algorithm with  $Pe \rightarrow \infty$  and  $Pe = 500$  is presented in Fig. 10. The evolution is similar to Fig. 6, but now the increase of the number of grid points is not linear with simulation time. Once again the adaptive strategy is performing well, and the compression ratios at the end of the simulation are 78% and 81% for  $Pe \rightarrow \infty$  and  $Pe = 500$ , respectively.

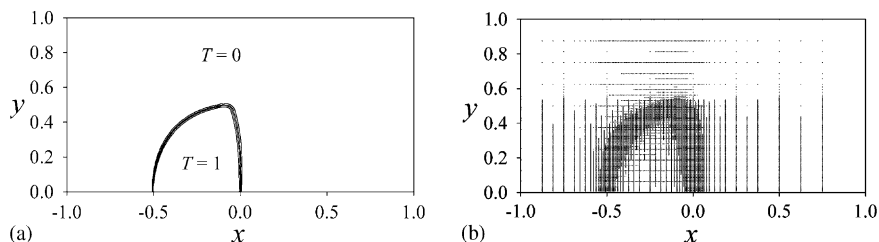


Fig. 8. (a) Solution of the Smith–Hutton problem at  $t = \pi/4$  with  $Pe \rightarrow \infty$  and (b) corresponding distribution of grid points in terms of spatial location.

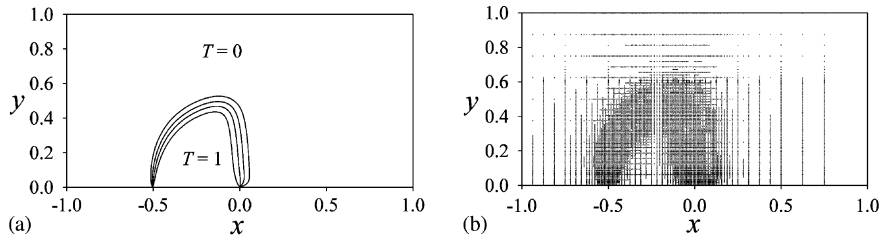


Fig. 9. (a) Solution of the Smith–Hutton problem at  $t = \pi/4$  with  $Pe = 500$  and (b) corresponding distribution of grid points in terms of spatial location.

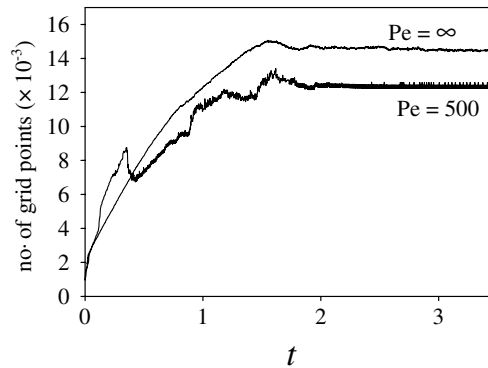


Fig. 10. Number of grid points used by the adaptation algorithm for the Smith–Hutton problem.

Table 3

Number of grid points (NGP), compression ratio (CR) and increase rate of grid points (IRGP) for the Smith–Hutton test problem for  $Pe \rightarrow \infty$ , using the adaptive approach

$J_{\max}$	Uniform mesh	Adaptive mesh		
	NGP	NGP	IRGP	CR (%)
3	81	81	–	0.0
4	289	239	2.95	17.3
5	1089	547	2.29	49.8
6	4225	1234	2.26	70.8
7	16 641	2879	2.33	82.7
8	66 049	6896	2.40	89.6
9	263 169	16 392	2.38	93.8
10	1 050 625	38 317	2.34	96.4

The values correspond to the average from  $t = 0$  to  $\pi/4$ .

Table 3 presents the increase rate of grid points (IRGP), defined as the ratio between the number of grid points used in two consecutive resolution levels for the adaptive approach. As in the previous problem the increase rate of grid points when using the adaptive strategy tends to 2.3 in contrast with the value of 4 obtained with a uniform mesh.

### 5.3. Burgers' equation

This example considers the numerical solution of the classical Burgers' equation [18]:

$$\frac{\partial u}{\partial t} = v \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - \left( \frac{\partial f(u)}{\partial x} + \frac{\partial f(u)}{\partial y} \right), \tag{18}$$

where the flux is  $f(u) = u^2/2$ . Here  $u$  may be taken as a velocity (or other advected property),  $x$  and  $y$  are the spatial co-ordinates,  $t$  is the time variable and  $v$  is the viscosity. All variables are normalized and so  $v$  may be considered as the inverse of a Reynolds number.

This equation is frequently used for testing numerical methods since it is non-linear and consequently, depending of the initial conditions chosen, discontinuities may develop from smooth solutions [19].

Eq. (18) was solved in the domain  $-1.5 \leq x \leq 1.5$  and  $-1.5 \leq y \leq 1.5$ , with either  $v = 5.0 \times 10^{-3}$  or in the inviscid limit ( $v \rightarrow 0$ ), and the following initial and boundary conditions were applied, following Kurganov and Tadmor [20]:

$$u(x, y, 0) = \begin{cases} -1 & (x - 0.5)^2 + (y - 0.5)^2 \leq 0.4^2, \\ 1 & (x + 0.5)^2 + (y + 0.5)^2 \leq 0.4^2, \\ 0 & \text{elsewhere,} \end{cases}$$

$$u(-1.5, y, t) = 0, \quad u(1.5, y, t) = 0, \quad u(x, -1.5, t) = 0, \quad u(x, 1.5, t) = 0.$$

For this specific type of initial condition there will be two cylindrical fronts which move with velocities equal to the mean of the velocities just before and after the shock front, while the trailing edges do not move at all. After a certain time gap ( $t \approx 0.307$  for  $v = 0$ ) the fronts will touch each other and will form a discontinuity in the solution, since they move in opposite directions; note that the cylinder centered in the third quadrant of the  $xy$  plane has a positive velocity. The velocity distributions at  $t = 0.5$  and  $2$ , considering the inviscid limit, are presented in Fig. 11(a) and (b) respectively. The corresponding distribution of the grid points for  $t = 0, 0.5$  and  $2$  are shown in Fig. 11(c)–(e), respectively.

When a small amount of “viscosity” is added to the equations the main features of the solution remain unchanged, except for the larger region in the original domain which is affected by the initial distribution of  $u$ . Now the trailing edges of the initial cylindrical distributions do move away from the centres due to transport by diffusion. For this case the velocity distributions at  $t = 0.5$  and  $2$ , with  $v = 5.0 \times 10^{-3}$ , are presented in Fig. 12(a) and (b) respectively. The distribution of the grid points for  $t = 0.5$  and  $2$  is presented in Fig. 12(c) and (d), respectively. The adaptation strategy proves to be capable of dealing with highly non-linear equations that include diffusive terms, as seen by the correct allocation of the grid points shown in Fig. 12(d).

Fig. 13 shows the time evolution of the number of grid points used by the adaptation algorithm. It is interesting to note that, for  $v = 5.0 \times 10^{-3}$ , at  $t \approx 0.8$  the number of grid points needed to compute the solution, with the prescribed threshold of  $\epsilon = 10^{-3}$ , decreases significantly. This result is very important since it demonstrates a virtue of the current strategy as compared to other approaches, such as the moving mesh method, where a constant number of grid points are used throughout the computations. The proposed strategy uses only the grid points that are actually necessary to attain a given precision, and so is more versatile and efficient.

Table 4 presents the increase rate of grid points (IRGP), defined as the ratio between the number of grid points used in two consecutive resolution levels for the adaptive approach. For this problem the tendency of the increase rate of grid points when using the adaptive strategy is still not very well defined at  $J_{\max} = 10$ .

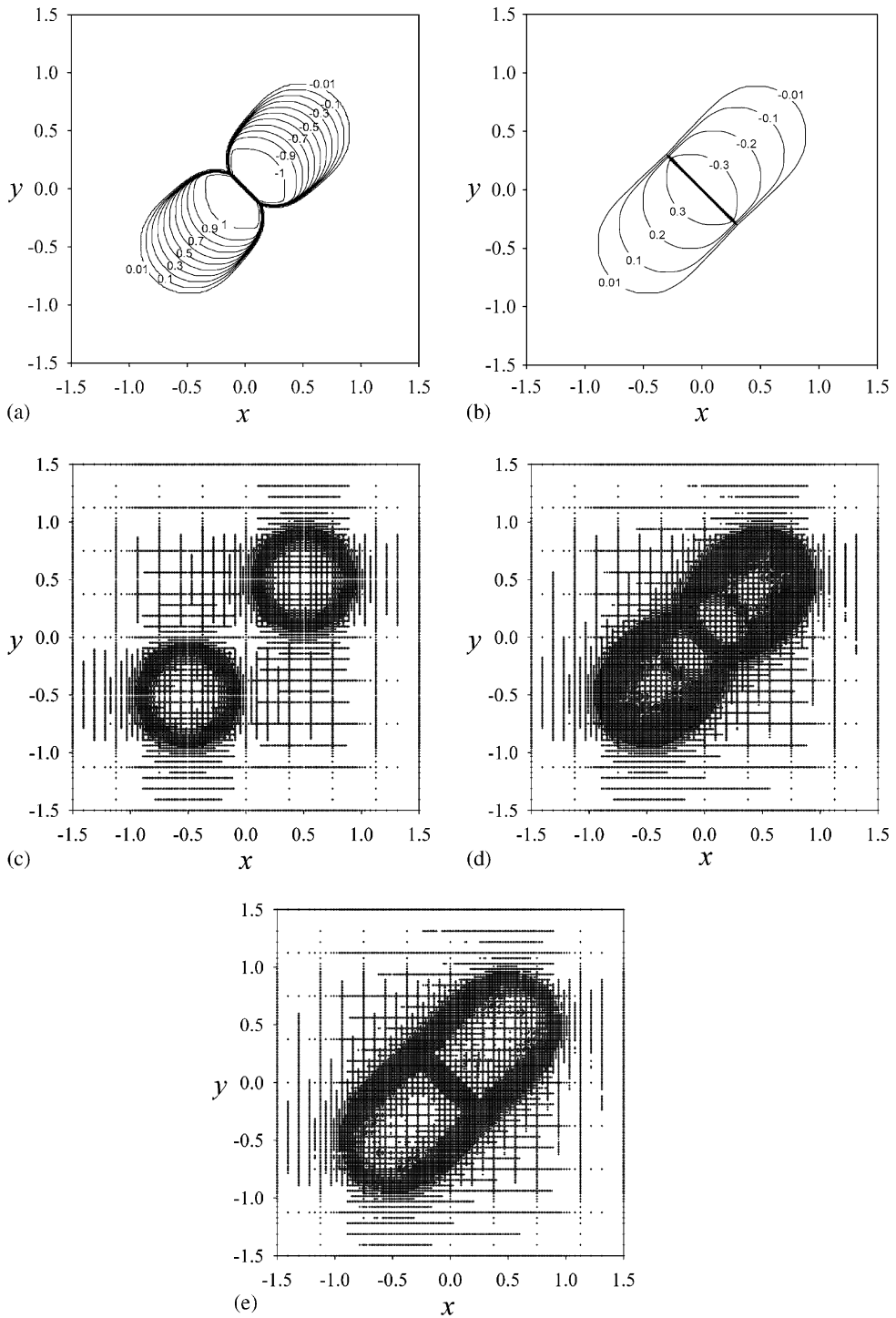


Fig. 11. Solution of the Burgers' equation, without viscosity ( $\nu = 0$ ), at (a)  $t = 0.5$  and (b)  $t = 2$ . Distribution of grid points in terms of spatial location at (c)  $t = 0$ , (d)  $t = 0.5$  and (e)  $t = 2$ .



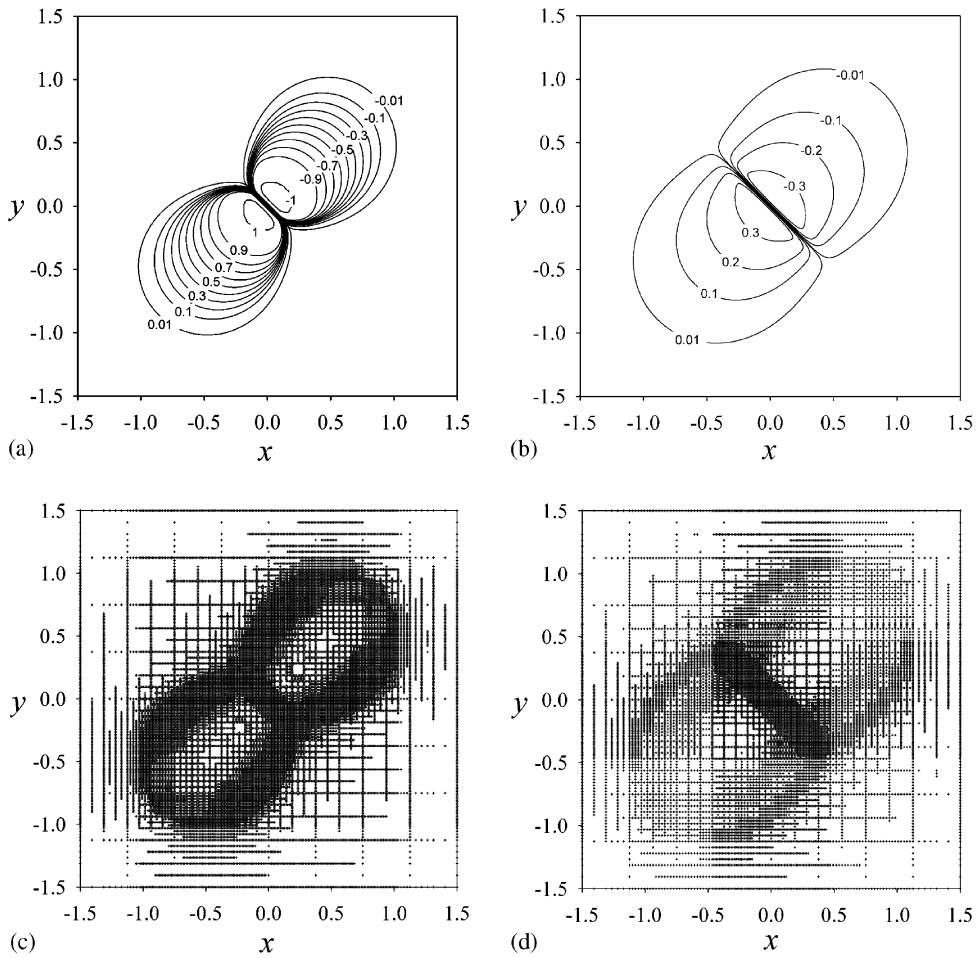


Fig. 12. Solution of the Burgers' equation with  $\nu = 5.0 \times 10^{-3}$  at (a)  $t = 0.5$  and (b)  $t = 2$ . Distribution of grid points in terms of spatial location at (c)  $t = 0.5$  and (d)  $t = 2$ .

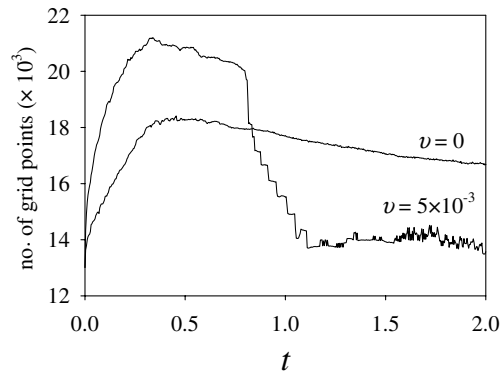


Fig. 13. Number of grid points used by the adaptation algorithm for the solution of the Burgers' equation.

Table 4

Number of grid points (NGP), compression ratio (CR) and increase rate of grid points (IRGP) for the Burgers' equation for  $v = 0$ , using the adaptive approach

$J_{\max}$	Uniform mesh	Adaptive mesh		
	NGP	NGP	IRGP	CR (%)
3	81	81	–	0.0
4	289	269	3.32	6.9
5	1089	806	3.00	26.0
6	4225	2349	2.91	44.4
7	16 641	6567	2.80	60.5
8	66 049	17 273	2.63	73.8
9	263 169	43 296	2.51	83.5
10	1 050 625	111 381	2.57	89.4

The values correspond to the average from  $t = 0$  to 2.

#### 5.4. Compressible Euler system

In this final example we test the proposed adaptive strategy on a problem involving the 2D compressible Euler system of conservation laws for gas dynamics, which is written in conservative form as follows:

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix} + \frac{\partial}{\partial x} \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E + p) \end{bmatrix} + \frac{\partial}{\partial y} \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E + p) \end{bmatrix} = 0, \quad (19)$$

where  $\rho$ ,  $u$ ,  $v$  and  $E$  are the density,  $x$ - and  $y$ -velocities and total energy of the gas per unit volume, respectively. The pressure,  $p$ , is given by

$$p = (\gamma - 1) \left[ E - \frac{\rho}{2} (u^2 + v^2) \right], \quad (20)$$

where  $\gamma$  is the ratio of specific heats ( $\gamma = 1.4$  is a good approximation for air). We solved the benchmark Riemann problem inspired by the standard 1D Sod shock-tube problem, which consists of the initial data (as in Configuration 4 of [21]):

$$\begin{bmatrix} p \\ \rho \\ u \\ v \end{bmatrix} (x, y, 0) = \begin{cases} [1.1 & 1.1 & 0 & 0]^T & x > 0.5, y > 0.5, \\ [0.35 & 0.5065 & 0.8939 & 0]^T & x < 0.5, y > 0.5, \\ [1.1 & 1.1 & 0.8939 & 0.8939]^T & x < 0.5, y < 0.5, \\ [0.35 & 0.5065 & 0 & 0.8939]^T & x > 0.5, y < 0.5. \end{cases}$$

At  $x = 0$ ,  $x = 1$ ,  $y = 0$  and  $y = 1$  reflecting boundary conditions are imposed. Several approaches can be found in the literature for the solution of Euler system of equations, either in the context of Riemann solvers [22] or pressure-based algorithms [23]. In this work we selected the semi-discrete central scheme of Kurganov and Tadmor [20], which is an extension to higher-order accuracy of the well-known Lax–Friedrichs scheme [24]. This semi-discrete central scheme requires the use of a TVD limiter in order to prevent formation of unphysical oscillations in the solution. It is here that the recently developed CUBI–STA scheme [7], explained in Section 2, shows its potential because it was devised in such a way that TVD conditions are satisfied.

Results for the gas density distribution and the corresponding location of grid points at times  $t = 0$ , 0.1 and 0.2 are presented in Fig. 14. The propagation of several fronts is seen to be well represented, with very sharp resolution, and the interaction region does not show signs of spurious oscillations in the computed fields.

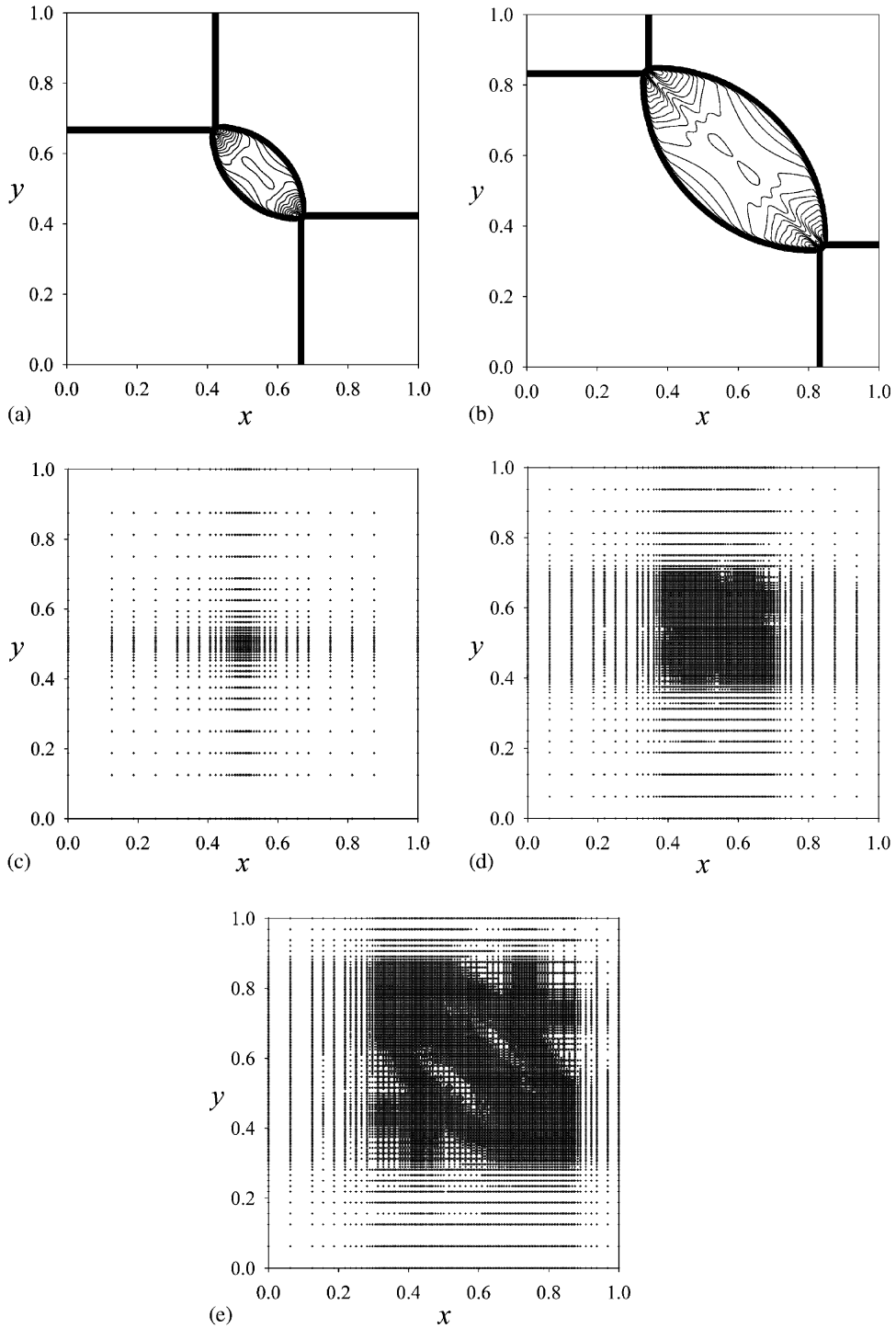


Fig. 14. Solution for the gas density of the compressible Euler system at (a)  $t = 0.1$  and (b)  $t = 0.2$ . Distribution of grid points in terms of spatial location at (c)  $t = 0$ , (d)  $t = 0.1$  and (e)  $t = 0.2$ .

## 6. Conclusions

A multiresolution approach for solving the typical PDEs of mathematics/physics has been described and applied to a number of two-dimensional test cases. Key features of this approach are the mesh adaptation strategy, whereby high density of grid points is allocated only where it is needed, and the high-resolution scheme for the convection terms, which prevents the formation of unphysical oscillations near sharp fronts.

The examples presented have demonstrated that the multiresolution approach can resolve very accurately the various moving fronts present in the solutions, for both linear and non-linear problems, involving either a single or a system of equations, while the total number of grid points required to achieve such level of resolution was seen to increase only moderately with the simulation time and the number of grid levels. In fact, compared with a fixed grid approach, in which grid doubling leads to 4 times more grid points, for the multiresolution approach the factor was found to be between 2.3 and 2.5. As a consequence the speed-up factor (ratio of CPU time for uniform mesh/adaptive mesh) was seen for the examples to increase at a rate almost around 1.7 as the level of resolution increases. Therefore, these speed-up factors quickly attain very high values; we found a typical value of 4.2 for a resolution level of 7 (which is fairly typical in current applications). These figures illustrate the efficiency of the proposed adaptive multiresolution approach.

## Acknowledgements

The work of J.C. Santos and P. Cruz was supported by FCT (grants SFRH/BD/6817/2001 and BD/21483/99, respectively). M.A. Alves wishes to thank the financial support provided by Fundação Calouste Gulbenkian.

## References

- [1] M.A. Alves, P. Cruz, A. Mendes, F.D. Magalhães, F.T. Pinho, P.J. Oliveira, Adaptive multiresolution approach for solution of hyperbolic PDEs, *Comput. Methods Appl. Mech. Engrg.* 191 (2002) 3909–3928.
- [2] S. Mallat, Multiresolution approximation and wavelet orthogonal bases of  $L^2(R)$ , *Trans. Amer. Math. Soc.* 315 (1989) 69–87.
- [3] A. Harten, Multiresolution representation of data: a general framework, *SIAM J. Numer. Anal.* 33 (1996) 1205–1256.
- [4] A. Harten, B. Engquist, S. Osher, S.R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes III, *J. Comput. Phys.* 71 (1987) 231–303.
- [5] A. Cohen, S.M. Kaber, S. Muller, M. Postel, Fully adaptive multiresolution finite volume schemes for conservation laws, *Math. Comput.* 72 (2003) 183–225.
- [6] S. Karni, A. Kurganov, G. Petrova, A smoothness indicator for adaptive algorithms for hyperbolic systems, *J. Comput. Phys.* 178 (2002) 323–341.
- [7] M.A. Alves, P.J. Oliveira, F.T. Pinho, A convergent and universally bounded interpolation scheme for the treatment of advection, *Int. J. Numer. Methods Fluids* 41 (2003) 47–75.
- [8] B.P. Leonard, The Ultimate conservative difference scheme applied to unsteady one-dimensional advection, *Comput. Methods Appl. Mech. Engrg.* 88 (1991) 17–74.
- [9] B.P. Leonard, Bounded higher-order upwind multidimensional finite-volume convection–diffusion algorithms, in: W.J. Minkowycz, E.M. Sparrow (Eds.), *Advances in Numerical Heat Transfer*, vol. 1, Taylor and Francis, 1996, pp. 1–57.
- [10] M.S. Darwish, F.H. Moukalled, Normalized variable and space formulation methodology for high-resolution schemes, *Numer. Heat Transfer B* 26 (1994) 79–96.
- [11] A. Harten, High resolution schemes for hyperbolic conservation laws, *J. Comput. Phys.* 49 (1983) 357–393.
- [12] M. Holmström, Solving hyperbolic PDEs using interpolating wavelets, *SIAM J. Sci. Comput.* 21 (1999) 405–420.
- [13] J. Walden, Filter bank methods for hyperbolic PDEs, *SIAM J. Numer. Anal.* 36 (1999) 1183–1233.
- [14] H.-M. Zhou, Wavelet transforms and PDE techniques in image compression, Ph.D. Thesis, Department of Mathematics, UCLA, 2000.
- [15] L.R. Petzold, Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations, *SIAM J. Sci. Statist. Comput.* 4 (1983) 136–148.

- [16] B.P. Leonard, J.E. Drummond, Why you should not use ‘Hybrid’, ‘Power law’ or related exponential schemes for convective modelling—There are much better alternatives, *Int. J. Numer. Methods Fluids* 20 (1995) 421–442.
- [17] M.S. Darwish, F. Moukalled, TVD schemes for unstructured grids, *Int. J. Heat Mass Transf.* 46 (2003) 599–611.
- [18] C.A.J. Fletcher, *Computational Techniques for Fluid Dynamics*, vol. I, second ed., Springer-Verlag, Berlin, 1991.
- [19] N.N. Carlson, K. Miller, Design and application of a gradient-weighted moving finite element code I: in one dimension, *SIAM J. Sci. Comput.* 19 (1998) 728–765.
- [20] A. Kurganov, E. Tadmor, New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations, *J. Comput. Phys.* 160 (2000) 241–282.
- [21] A. Kurganov, E. Tadmor, Solution of two-dimensional Riemann problems for gas dynamics without Riemann problem solvers, *Numer. Methods Partial Differential Equations* 18 (2002) 584–608.
- [22] C. Hirsch, in: *Numerical Computation of Internal and External flows*, vol. 2, John Wiley, Chichester, 1990.
- [23] J.H. Ferziger, M. Perić, *Computational Methods for Fluid Dynamics*, Springer-Verlag, Berlin, 1996.
- [24] P.D. Lax, Weak solutions of nonlinear hyperbolic equations and their numerical computation, *Comm. Pure Appl. Math.* 7 (1954) 159–193.