INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2003; **42**:297–321 (DOI: 10.1002/fld.514)

Discretization of transport equations on 2D Cartesian unstructured grids using data from remote cells for the convection terms

John S. Anagnostopoulos*,†

Fluids Section, School of Mechanical Engineering, National Technical University of Athens, Greece

SUMMARY

This paper presents a new finite volume discretization methodology for the solution of transport equations on locally refined or unstructured Cartesian meshes. The implementation of the cell-face values of the dependent variables enables the employment of data from remote cells and thus the use of higher-order differencing schemes. It also results in simple and flux-conservative multiple-scale stencils for the discretization of the governing equations. The latter are finally cast into a generalized form that does not depend on the local mesh structure. The performance of the numerical model is demonstrated on some classical 2D problems using various gridding techniques and a bounded second-order upwind scheme. A stable and efficient behaviour of the algorithm is observed in all test cases. The results indicate that the combination in the present model of both local grid refinement and second-order discretization can produce substantially more accurate solutions than each of the above techniques alone, for the same computational effort. The method is also applicable to turbulent flows and can be easily extended to three-dimensions. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: Cartesian unstructured grids; mesh refinement; finite volume method; multiple-scale stencils; square cavity flow; backward facing step

1. INTRODUCTION

A remarkable part of the research effort in CFD has been devoted to the elimination of numerical errors from the solution results of the governing equations. The use of higher-order differencing schemes and the reduction of the size of the control volumes in viscous or complex flow regions constitute the basic numerical tools that can been used towards this goal.

Unstructured triangular (2D) or tetrahedral (3D) grids offer a great geometric flexibility and the potential for automation and local refinement [1]. However the generation of such grids in regions where high aspect ratios are required (e.g. boundary layers) is quite difficult. Hybrid or generalized grids have recently appeared in the literature (e.g. References [2–4]) in which a structured mesh is usually implemented to capture the directionality of the flow in viscous

* Correspondence to: J. S. Anagnostopoulos, Fluids Section/School of Mechanical Engineering, National Technical University of Athens (N.T.U.A.), 9 Heroon Polytechniou ave., 15773, Zografou, Athens, Greece.
† E-mail: j.anagno@fluid.mech.ntua.gr

Received January 2002
Revised 6 January 2003

regions near solid components. However, the generation of such grids is a formidable task and algorithms are still being developed [1].

In case of a structured mesh, an increase of its density at some regions of high gradients of the dependent variables or near boundaries produces many unnecessary points at the rest domain. Several multiblock techniques have been developed to overcome this problem by dividing the physical region into a number of subregions, each of which is covered by a properly optimized structured grid. These individual grids can be patched together at common interfaces [5, 6] or they are allowed to overlap [7]. The transfer of information between the grids is critical for a correct simulation, and the grid generation may be very time-consuming especially for complex geometries, where the user interaction cannot be avoided although considerable progress has been made towards automatic blocking [8].

In an alternative technique, grid refinement algorithms are used to increase the resolution of a base grid at certain regions through cell subdivision [3, 9–12]. The resulting locally refined or embedded grids are usually solved together with the base grid, thus no information transfer is required. Several methods have been tried to mark cells for refinement, from a manual refinement in regions known *a priori* to be located in the vicinity of walls or in regions of high shear [11, 13], to a fully automated and solution adaptive process, based on weight functions containing combinations of first and second derivatives of some of the dependent variables [12, 14] or error estimators [3, 15].

The use of Cartesian grids is the simplest and most straightforward discretization way. For this reason several locally refined gridding methods have been applied in rectangular grids in a variety of applications [10, 12, 14, 16–18]. Cell subdivision is quite simple in such grids, although several constraints are imposed in order to reduce the truncation error; the most common is the retention of a grid interface ratio up to 2 [12]. However, there is a difficulty in finding neighbouring points, as well as in treating points at the interface of cells having different refinement levels. The standard cell-centred finite volume discretization can cause significant truncation error at the interface of such cells, which increases with the grid interface ratio [12, 16] due to the non-orthogonal differencing along the lines connecting the cell centres. For this reason the single-scale stencils for the finite difference discretization of derivatives, that are commonly used on unstructured grids, produce truncation error in general first-order in the grid spacing [3, 4, 14, 18, 19]. More accurate, multiple-scale stencils have been developed and used in recent works [10, 12, 16], but their application is restricted to interfaces with integer interface ratio, usually equal 2.

The multiple-scale stencils operation is based on the introduction of auxiliary points in the coarser cell of a coarse–fine interface in order to be used as fictitious neighbours of the finer cells. The values of the dependent variables at these points are interpolated from the adjacent cells, thus they may not appear in the final governing equations. This treatment simplifies the data structure but results in non-flux-conservative discretizations [12] or increases considerably the complexity of the discrete operator even for the simplest refinement ratio of 2 [16]. Moreover, different discretized equations are obtained for a grid node depending on the refinement level of the surrounding cells. The fictitious points are included explicitly into the final equations of Reference [10] to preserve a flux-conservative computational stencil. However, their number as well as their interpolation formula depends on the local grid structure around each node.

The traditional low-order discretization introduces excessive amounts of numerical diffusion at high Reynolds numbers or with relatively coarse grids. For this reason higher-order dif-

ferencing has become essential in most computer codes using regular meshes. However, the application of such schemes on refined or adaptive grids is very restricted, due to the difficulty in defining the needed data from remote grid cells. Hence most of the existing numerical simulators incorporating local grid refinement use first-order upwind or second-order centred schemes [2–4, 9, 10, 12] that require data only from the adjacent grid cells.

A new finite volume discretization methodology is developed and presented in this paper for the solution of transport equations in arbitrarily refined or unstructured Cartesian grids. An important advantage of the method is the potentiality to implement higher-order differencing schemes in such grids using data not only from the neighbouring but from remote cells as well. Thus the benefits of increased accuracy of both local mesh refinement and higher-order differencing techniques can be combined in the same algorithm. Key modelling feature is the calculation of cell-face values for the dependent variables at every grid-cell. The storage of these additional data increases the computer memory demands, but on the other hand exhibits considerable advantages concerning the simplicity and the generality of the algorithm. Unlike other similar methods (e.g. Reference [10]), their number is constant for every control volume (four for 2D meshes) and does not increase by the refinement ratio per local interface, hence no complexity is added to the data structure. Moreover, the above number is slightly increased (becomes six) for 3D simulations.

A multiple-scale stencil is implemented by which the auxiliary points are expressed in terms of the adjacent cell-centre and cell-face values. This treatment produces flux-conservative discredized equations of a very simple and generalized form, which does not rely on the cell refinement ratio or on the local mesh structure.

The new model is at present applicable on orthogonal domains with straight horizontal and vertical boundaries. However, in recent years there has been a renewal of interest in numerical methods with irregular stationary or moving boundaries on fixed Cartesian grids, and several techniques have been developed to treat grid cells that are traversed by a domain boundary [6, 20, 21]. Such a technique can be incorporated in the present algorithm, extending its applicability to computational domains of any complexity.

The lid-driven square cavity, the transport of a step change, and the backward-facing step flow were selected among the most classical test cases to evaluate the new methodology. A bounded second-order upwind scheme [22] was implemented in the present work and the model was first applied in rectangular grids of arbitrary topology. Three alternative techniques to obtain the requested remote data were comparatively tested in these grids. Next, the convection–diffusion equation of the step profile problem was solved to assess the gain in accuracy of the results that can be achieved by combining a second-order upwind differencing scheme and a local grid refinement technique. Finally, in order to examine the performance of the model on some more practical grid arrangements, several solution-adaptive grids were constructed and solved in the square cavity and in the backward-facing step domain. In all the above cases the corresponding results obtained by the traditional Hybrid scheme (that combines first-order upwind and central differencing) are also given for comparison.

2. DISCRETIZATION PROCEDURE

A smooth refinement of an adaptive grid is desirable in most cases and for this reason a refinement ratio of two combined with several refinement levels is the most common practice.

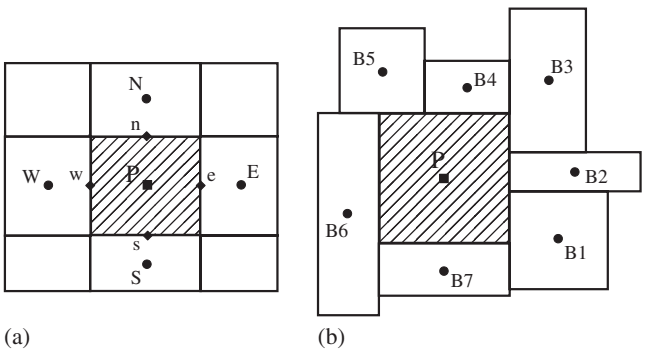


Figure 1. Example of control volumes arrangement in a Cartesian structured mesh (a), and a Cartesian unstructured mesh (b).

However, a general grid adaptation technique should allow for arbitrary refinement ratios, as well as for the application of any derefinement algorithm. Computational grids like the example shown in Figure 1(b) can thus be formed and a general flow solver for the governing equations should be applicable to them.

In such a general grid configuration the terms 'refinement ratio', 'grid level', and 'coarser' or 'finer region' have no meaning; therefore every grid cell must be treated in the same, generalized way.

2.1. Governing equations and cell-face variables

The time-averaged transport equation of a variable Φ in 2D, Cartesian co-ordinates is

$$\frac{\partial}{\partial x}(\rho U\Phi)+\frac{\partial}{\partial y}(\rho V\Phi)-\frac{\partial}{\partial x}\left(\Gamma_\Phi\frac{\partial\Phi}{\partial x}\right)-\frac{\partial}{\partial y}\left(\Gamma_\Phi\frac{\partial\Phi}{\partial y}\right)=S_\Phi \tag{1}$$

where U, V are the fluid velocities in the x and y direction, respectively, Γ_Φ is the diffusivity and S_Φ the source term. Integration over a finite control volume, Vol, having faces A_i, gives:

$$A_{\mathrm{e}}\left(\rho U\Phi-\Gamma_\Phi\frac{\partial\Phi}{\partial x}\right)_{\mathrm{e}}-A_{\mathrm{w}}\left(\rho U\Phi-\Gamma_\Phi\frac{\partial\Phi}{\partial x}\right)_{\mathrm{w}}+A_{\mathrm{n}}\left(\rho V\Phi-\Gamma_\Phi\frac{\partial\Phi}{\partial y}\right)_{\mathrm{n}}$$
$$-A_{\mathrm{s}}\left(\rho V\Phi-\Gamma_\Phi\frac{\partial\Phi}{\partial y}\right)_{\mathrm{s}}=S_\Phi\,\mathrm{Vol} \tag{2}$$

where the quantity in each bracket is calculated on the corresponding face centre (east, west, etc., Figure 1(a)). It must be noted that the more general terms 'volume' and 'face' are retained here, although in a 2D grid they are actually 'area' and 'line', respectively. The first derivatives can be obtained using central differences, e.g. (Figure 1(a)):

$$\left.\frac{\partial\Phi}{\partial x}\right|_{\mathrm{e}}=\frac{\Phi_{\mathrm{E}}-\Phi_{\mathrm{P}}}{\Delta X_{\mathrm{E-P}}}$$

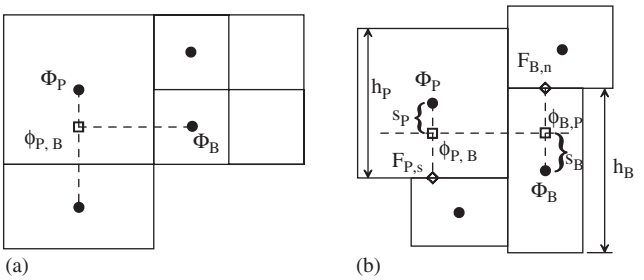


Figure 2. Location of auxiliary points in a normally refined grid (a), and an unstructured grid (b).

whereas the value of Φ on the faces can be also expressed in terms of the values of Φ at the centres of the adjacent volumes using first- or higher-order differencing schemes, as in Section 2.4. Finally, for a control volume centred at node P the discretized equation can be written in the following form [23]:

$$\alpha_P \Phi_P = \sum_{i=1}^{n} \alpha_i \Phi_i + S'_\Phi, \quad \alpha_P = \sum_{i=1}^{n} \alpha_i \tag{3}$$

where α_i are the combined convection/diffusion fluxes across the faces and $n=4$ for a regular 2D cell. The fluxes $(\rho U)_i$ and $(\rho V)_i$ are included in the linking coefficients α_i, and all the rest terms go in the source term S_Φ.

In the present model all the dependent variables are stored at the centres of the control volumes according to the collocated or non-staggered arrangement (Figure 1). For a generalized 2D grid configuration like the one in Figure 1(b), Equation (3) includes now all the neighbouring cells B1 to B7, namely the cells having a common interface with the central grid-cell. According to previous studies [10, 16], a number of auxiliary points can be defined and used in order to reduce the significant truncation error emerging from the fact that the lines joining the adjacent cell centres are, in general, non-orthogonal to the corresponding interfaces (Figure 1(b)). The usual practice is to keep a constant refinement ratio of 2:1 and to introduce 'fictitious' nodes in the coarser cell volume on the same horizontal or vertical level with the centres of the neighbouring finer cells (Figure 2(a)). The values at these 'fictitious' points are then expressed by interpolating from the values at two adjacent cell centres (e.g. Φ_P and Φ_S in Figure 2(a)).

In case of a general grid as in Figure 2(b), where no distinction to coarser or finer cell is made, auxiliary points are defined in both neighbouring cells P and B, located in the same horizontal or vertical joining line that bisects the common interface, as shown in Figure 2(b). Hence for the central node P and a neighbouring node B the term $\alpha_B(\Phi_P - \Phi_B)$ of Equation (3) can be replaced by the term $\alpha_B(\phi_{P,B} - \phi_{B,P})$. Now the values $\phi_{P,B}$ and $\phi_{B,P}$ at the corresponding auxiliary points cannot be interpolated from adjacent cell centres. Alternatively, they could be obtained using the values (if known) of the dependent variable Φ at the cell-faces ($F_{P,s}$ and $F_{B,n}$, respectively, in the example of Figure 2(b)), via the relations:

$$\begin{aligned} \phi_{P,B} &= g_{P,B} F_{P,s} + (1 - g_{P,B})\Phi_P \\ \phi_{B,P} &= g_{B,P} F_{B,n} + (1 - g_{B,P})\Phi_B \end{aligned} \tag{4}$$

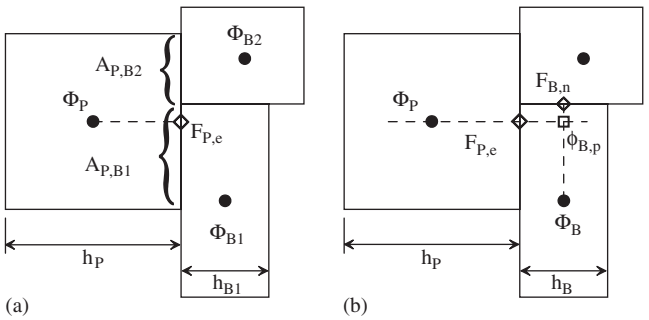


Figure 3. Definition of cell-face values using an area-weighted formula (7) (a), and a linear interpolation (8) (b).

The lowercase ϕ will be used to indicate an interpolated value of a dependent variable at an auxiliary point, whereas F will represent a value at the centre of a cell-face. The co-ordinates of the joining line determines both the proper cell-faces j (e,w,n,s) and the geometrical coefficients g for a given pair of neighbouring cells P and B:

$$g_{\mathrm{P,B}} = s_{\mathrm{P}}/(h_{\mathrm{P}}/2), \quad g_{\mathrm{B,P}} = s_{\mathrm{B}}/(h_{\mathrm{B}}/2) \tag{5}$$

where s is the distance of the cell centre (P or B) from the joining line and h the cell dimension in this direction.

Using now the expressions (4), and after some algebraic treatment, the above term becomes:

$$\alpha_{\mathrm{B}}(\phi_{\mathrm{P,B}} - \phi_{\mathrm{B,P}}) = \alpha_{\mathrm{B}}(\Phi_{\mathrm{P}} - \Phi_{\mathrm{B}}) + \alpha_{\mathrm{B}}[g_{\mathrm{P,B}}(F_{\mathrm{P},j} - \Phi_{\mathrm{P}}) + g_{\mathrm{B,P}}(\Phi_{\mathrm{B}} - F_{\mathrm{B},k})]$$
$$j\text{–}k = \text{e–w or w–e or n–s or s–n} \tag{6}$$

The first part of the resulting term is the original one, whereas the second part can be included in the source term of Equation (3), thus preserving the grid link reciprocity. In case of a non-refined grid the second part vanishes, since both geometrical coefficients become zero (joining line passes through the cell centres). Also, in case of a refinement ratio of two (Figure 2(a)), the above relation returns to the suggested treatment of previous studies [10, 16]. However, instead of the values of the dependent variables at the 'fictitious' points, the present approach uses the cell-face values. This has the advantage of reducing the number of additional points (always four per each grid cell, for any number of neighbouring cells), as well as of facilitating the interpolation process. The latter could be cumbersome if the requested point is located somewhere inside a control volume of an adaptive grid having no simple structure.

A point located on a cell-interface has two obvious reference points to correlate with, the centres of the adjacent grid-cells. Based on this concept, the value of a dependent variable at the centre of a cell-face is computed by the following method (see Figure 3(a)). At first the values at the common interface of the reference cell and all the neighbouring cells in the desired face direction are interpolated from the corresponding cell centre values that are assumed, as a first approximation, constant throughout the cell volume. Next, these values are area weighted to give the desired value at the centre of the cell-face. The mathematical

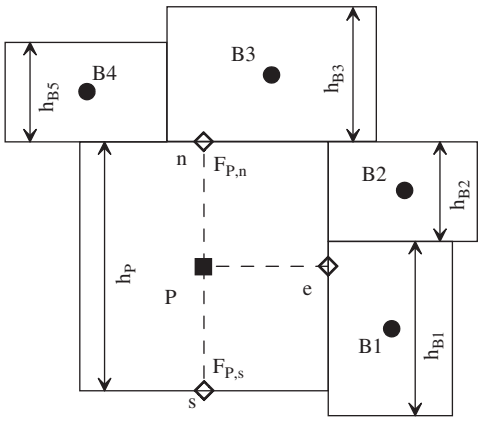


Figure 4. Sketch of arbitrary control volumes arrangement explaining the derivatives discretization equations (9).

formula can be written as follows:

$$F_{P,j} = \frac{\sum_i^m [w_{P,i}\Phi_P + (1 - w_{P,i})\Phi_i]A_{P,i}}{\sum_i^m A_{P,i}} \tag{7}$$

where $w_{P,i} = h_i/(h_P + h_i)$ is the weighting factor for the interpolation between cells P and i, $A_{P,i}$ is their common interface area and the summation i covers all the cells neighbouring to the j-face.

An alternative method was also tested, which uses only linear interpolations. In the example of Figure 3(b) the value at the eastern cell-face $F_{P,e}$ is obtained from the following formula:

$$F_{P,e} = w_{P,B}\Phi_P + (1 - w_{P,B})\phi_{B,P} \tag{8}$$

where the value $\phi_{B,P}$ at the intersection of lines P-P,e and B-B,n is interpolated between the values Φ_B and $F_{B,n}$. Since the latter constitutes a cell-face quantity as well, the method uses at first its value from the previous iteration of the solution algorithm. Next, Equation (8) is repeatedly solved throughout the calculation domain, before the next iteration of the main algorithm. It was found that two or three such internal iterations are adequate to prevent any instability problems of the algorithm. Equations (7) and (8) become identical in case of a regular Cartesian grid, but in the common case of grid refinement ratio of two they give the same result only for the face values of the coarser cell.

2.2. *Discretization of derivatives*

An additional benefit from the introduction of the cell-face values of the dependent variables is the capability to calculate the various derivatives appearing in the governing equations in a generalized and simple way. For example, the gradients $\partial\Phi/\partial y$ at the centre and the faces of a cell are expressed as follows (Figure 4):

$$\left(\frac{\partial\Phi}{\partial y}\right)_P = \frac{F_{P,n} - F_{P,s}}{h_P} \tag{9a}$$

$$\left(\frac{\partial \Phi}{\partial y}\right)_{\mathrm{n}} = \frac{\sum_i^m [\frac{\Phi_i - \Phi_{\mathrm{P}}}{0.5(h_{\mathrm{P}}+h_i)}] A_{\mathrm{P},i}}{\sum_i^m A_{\mathrm{P},i}} \tag{9b}$$

$$\left(\frac{\partial \Phi}{\partial y}\right)_{\mathrm{e}} = \frac{\sum_i^m [w_{\mathrm{P},i}(\frac{\partial \Phi}{\partial y})_{\mathrm{P}} + (1 - w_{\mathrm{P},i})(\frac{\partial \Phi}{\partial y})_i] A_{\mathrm{P},i}}{\sum_i^m A_{\mathrm{P},i}} \tag{9c}$$

where the summation i in (9b) and (9c) includes the north- or the east-face neighbours of the cell, respectively.

The cell-face values and the area-weighted summations in the above expressions result in multiple-scale stencils that, in addition to the reduced truncation error compared to single-scale stencils [12], can be applied to any mesh structure, from the commonly used grid interface ratio of two, to a completely unstructured configuration as in Figure 1(b). In case of a regular Cartesian grid the above gradients take the optimum form in terms of accuracy and efficiency [4].

2.3. Flux continuity

The total flux $\mathbf{q}_{\mathrm{P},j}$ through a cell-face j is the summation of the fluxes through the common interface of the cell and its neighbours i at this face:

$$\mathbf{Q}_{\mathrm{P},j} = \sum_i^m \mathbf{q}_{\mathrm{P},i}, \quad j = \mathrm{e,w,n,s} \tag{10}$$

where the vectorized symbols are introduced to account for the direction of the flow in the Cartesian system.

According to the technique of Rhie and Chow [24] the calculation of the cell-face mass flux is arranged to depend on both the velocity and the pressure field. This technique is adopted in the present study and furthermore, the velocities and pressures at the adjacent cell centres are replaced with the corresponding values at the auxiliary points obtained via the interpolation equations (4) (Figure 2(b)). The same is also done for the rest quantities (e.g. pressure gradient) that are involved in the flux expression [3]. With this treatment the flux through the interface of two neighbouring cells is computed for both cells via the same equation, therefore the method is flux conservative.

2.4. Higher-order discretization

Higher-order differencing schemes need in general data not only from the neighbour cells but from remote cells as well. In simple Cartesian grids this constitutes only a storage problem, but in an arbitrarily adaptive or unstructured grid the major difficulty is to define such data. For example, the second-order upwind (SOU) scheme gives for the east face of a simple grid cell P, Figure 5(a):

$$\Phi_{\mathrm{e}} = \begin{bmatrix} \Phi_{\mathrm{P}} + (\Phi_{\mathrm{P}} - \Phi_{\mathrm{W}}) \dfrac{\Delta X_{\mathrm{P-e}}}{\Delta X_{\mathrm{W-P}}} & q_{\mathrm{P,e}} > 0 \\ \Phi_{\mathrm{E}} + (\Phi_{\mathrm{E}} - \Phi_{\mathrm{EE}}) \dfrac{\Delta X_{\mathrm{e-E}}}{\Delta X_{\mathrm{E-EE}}} & q_{\mathrm{P,e}} < 0 \end{bmatrix} \tag{11}$$

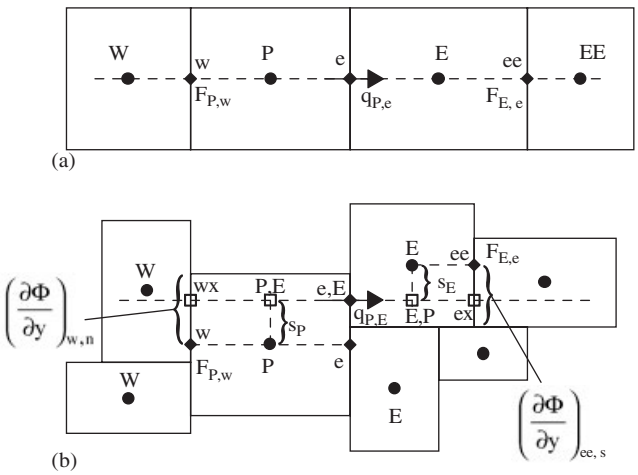


Figure 5. Location of points of a second-order upwind scheme stencil for a regular grid (a), and for an unstructured grid (b).

and the needed data at points P, W, E, and EE are readily available (ΔX_{i-j} is the distance between points i and j). On the contrary, it is rather impossible to define even the location of the corresponding points for the arbitrary grid of Figure 5(b). The technique developed in this study to overcome this difficulty is described below:

The dependent variables at points EE and W of Equation (11) can be replaced by corresponding expressions that include the values at points ee and w, respectively (Figure 5(a)), obtained by linear interpolation from the adjacent nodes:

$$\Phi_{EE} = \frac{F_{E,e}\Delta X_{E-EE} - \Phi_E \Delta X_{ee-EE}}{\Delta X_{E-ee}} \tag{12}$$

$$\Phi_W = \frac{F_{P,w}\Delta X_{W-P} - \Phi_P \Delta X_{W-w}}{\Delta X_{w-P}} \tag{13}$$

Then, after some algebraic manipulation Equation (11) takes the simple form:

$$\Phi_e = \begin{bmatrix} 2\Phi_P - F_{P,w} & q_{P,e} > 0 \\ 2\Phi_E - F_{E,e} & q_{P,e} < 0 \end{bmatrix} \tag{14}$$

where the neighbour cell-face values appear instead of the values at remote cells (i.e. EE). This relation is identical with Equation (11) providing that the values $F_{P,w}$ and $F_{E,e}$ are linearly interpolated. And using the procedure described previously (Equations (4)–(7)), the latter can be computed even for an unstructured grid, like in Figure 5(b). In this case however the joining line that bisects the common interface of the cell P and a neighbour cell E does not in general pass through the cell-face centres w and ee, but intersects the faces at points wx and ex, respectively (Figure 5(b)). Hence, the corresponding expression for the interface

value must be:

$$\Phi_{e,E} = \begin{bmatrix} 2\phi_{P,E} - f_{P,E} & q_{P,E} > 0 \\ 2\phi_{E,P} - f_{E,P} & q_{P,E} < 0 \end{bmatrix} \tag{15}$$

where the lowercase f indicates the value of a dependent variable at an auxiliary point anywhere on a cell-face, likewise the value ϕ within the cell volume. The two subscripts gives again the base node and its neighbouring in the opposite direction, respectively; hence the values $f_{P,E}$ and $f_{E,P}$ in Equation (15) correspond to the auxiliary points wx and ex, in Figure 5(b).

As a first approximation these values can be obtained assuming that the dependent variable is constant over the entire cell-face, namely:

$$f_{P,E} = F_{P,w} \quad \text{and} \quad f_{E,P} = F_{E,e} \tag{16}$$

An alternatively and more accurate technique is to obtain the needed values using the previously calculated gradients along the corresponding cell-faces (e.g. Equation (9c)) and the relations:

$$f_{P,E} = F_{P,w} + s_P \left(\frac{\partial \Phi}{\partial y}\right)_w \quad \text{and} \quad f_{E,P} = F_{E,e} - s_E \left(\frac{\partial \Phi}{\partial y}\right)_{ee} \tag{17}$$

Finally, in an even more elaborate approach the needed gradients can be calculated separately for each half of a cell-face, including in the summation of Equation (9c) only the neighbour cells having common interface with the corresponding part of the face. Thus, in the example of Figure 5(b) the previous relations become:

$$f_{P,E} = F_{P,w} + s_P \left(\frac{\partial \Phi}{\partial y}\right)_{w,n} \quad \text{and} \quad f_{E,P} = F_{E,e} - s_E \left(\frac{\partial \Phi}{\partial y}\right)_{ee,s} \tag{18}$$

All the above three relations return to the simple form (14) in case of a regular, non-refined grid. Moreover, they are compatible with a possible application of a boundary condition at a cell-face.

The above methodology can be applied to obtain a more compact form (without remote points) in many other second or higher-order schemes (e.g. QUICK). The bounded second-order upwind scheme BSOU [22] is used in the present study to evaluate the developed methodology.

2.5. The discretized equations

Following the previously described discretization methodology, the resulting algebraic equations take the general form:

$$\alpha_P \Phi_P = \sum_{i=1}^{n} \alpha_i \Phi_i + \sum_{i=1}^{n} b_i + \sum_{i=1}^{n} c_i + S_\Phi, \quad \alpha_P = \sum_{i=1}^{n} \alpha_i \tag{19}$$

where the additional terms b_i contain the remote-nodes data used by the BSOU and introduced explicitly [22], and the terms c_i comes from the implementation of the auxiliary points

(Equation (6)). The link coefficients α_i and the terms b_i and c_i are expressed as follows:

$$\alpha_i = \max(0, -\mathbf{q}_{\mathrm{P},i} \cdot \mathbf{d}_{\mathrm{P},i}) + \mu A_{\mathrm{P},i}/\Delta s_{\mathrm{P},i} \tag{20}$$

$$b_i = \min(0, \mathbf{q}_{\mathrm{P},i} \cdot \mathbf{d}_{\mathrm{P},i})\gamma_i(f_{i,\mathrm{P}} - \phi_{i,\mathrm{P}}) + \max(0, \mathbf{q}_{\mathrm{P},i} \cdot \mathbf{d}_{\mathrm{P},i})\gamma_i(f_{\mathrm{P},i} - \phi_{\mathrm{P},i}) \tag{21}$$

$$c_i = \alpha_i[g_{\mathrm{P},i}(F_{\mathrm{P},j} - \Phi_{\mathrm{P}}) - g_{i,\mathrm{P}}(F_{i,k} - \Phi_i)], \quad j\text{–}k = \text{e–w or w–e or n–s or s–n} \tag{22}$$

where $\mathbf{d}_{\mathrm{P},i}$ is the unit outward vector normal to the P cell-faces, μ is the fluid viscosity and $\Delta s_{\mathrm{P},i}$ the distance between the auxiliary points of $\phi_{\mathrm{P},i}$ and $\phi_{i,\mathrm{P}}$. The coefficient γ in Equation (21) is a blending factor of the BSOU scheme that preserves its boundedness property [22]. Also, the auxiliary values ϕ and f appearing in the same equation are obtained from their definition Equations (4) and (16), (17) or (18) respectively.

3. SOLUTION ALGORITHM

A preconditioned bi-conjugate gradient (Bi-CG) method is used to solve the system of discretized equations (19) for a dependent variable Φ at all the solvable grid nodes. The incomplete lower/upper (ILU) decomposition with no fill-in was used for preconditioning. Due to its explicit nature this solver does not require knowledge of the matrix band structure and can be easily applied to any adaptive or unstructured grid.

In the examined test cases of steady, incompressible, laminar flows the coupled system of the discretized Navier-Stokes equations is solved iteratively using the SIMPLE algorithm [23], modified for non-staggered grids [24]. With this algorithm the pressure field and the fluxes through the cell-faces (contained in the link coefficients of Equation (19)) are progressively corrected in order to satisfy the continuity equation. The latter is also extracted from the general form (3), setting $\Phi = 1$.

For the purposes of the present study the computational stencils can employ any number of points, depending on the specific grid topology. However, for practical applications this number can be reduced to at most nine, following certain grid construction rules to control the grid spacing changes (e.g. Reference [12]).

A proper numbering of the control volumes could enable the discretization algorithm to split into two parts, one for the regular cells and a second for the cells having common interface with at least one cell of different refinement. By this manipulation the cell-face values and gradients can be stored and used only for the second class of cells, which usually are only a small portion of the total number. Since no memory storage problems are encountered in the present 2D calculations, this treatment is planned to implement in a 3D-version of the algorithm.

The extension to 3D problems does not present difficulties, because the final form of the discretized equations (19)–(22) remains unchanged and the same is valid for most of the rest modelling expressions, as the cell-face calculation Equations (7) and (8) or the derivatives Equations (9). A little more complex would be the interpolation formulae of Equations (4), (17) and (18) for the auxiliary points definition. Also, the solution algorithm does not need any modification to apply in 3D cases.

4. MODEL APPLICATION AND RESULTS

4.1. Arbitrary unstructured grids in the lid-driven square cavity

The classic 2D problem, in which the flow in a square cavity is driven by the lid moving with uniform velocity, has been widely used to evaluate discretization schemes and computational meshing techniques. Here it was used at first to check the applicability of the methodology to arbitrary grids, as well as to investigate the performance of the alternative sub-models expressed by Equations (7), (8) and (16)–(18).

A special algorithm was developed for the generation of arbitrary Cartesian meshes in the square cavity. The algorithm fills progressively and in a random way the domain with control volumes of dimensions that are randomly selected between a minimum and a maximum value. Keeping the minimum cell dimension constant and equal to 1/40 of the square size, a maximum permissible aspect ratio of 1:1 produces the uniform 40×40 grid, whereas higher aspect ratios result in more irregular grids, as shown in Figures 6(a) and 7(a) obtained using the maximum ratios 2:1 and 5:1, respectively. No-slip boundary conditions are applied to all boundaries.

A solution for $Re = 1000$ was first taken using regular uniform grids and the BSOU scheme. The results of a 200×200 nodes such grid were almost identical with the ones obtained by a coarser, 160×160 grid, as well as with other exact solutions in the literature [25]. Thus the former are considered adequately accurate to be used for comparison purposes.

The accuracy of the numerical results of an arbitrary grid is quantified by calculating the mean percentage error from the following relation:

$$\text{Error}(\%) = \frac{100}{A_{\text{lid}}} \sum_{i=1}^{N} \frac{(|U_i - u_{\text{exact}}| + |V_i - v_{\text{exact}}|)h_{i,x}h_{i,y}}{2U_{\text{lid}}} \tag{23}$$

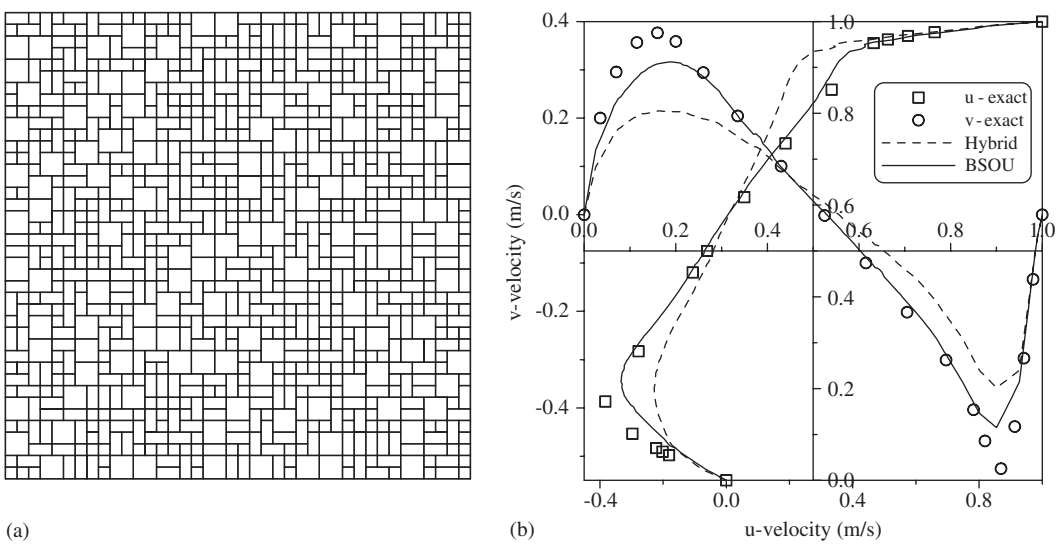


Figure 6. Indicative results for the square driven cavity, $Re = 1000$: mesh structure (a); u and v velocity profiles along the cavity vertical and horizontal mid-lines, respectively (b).

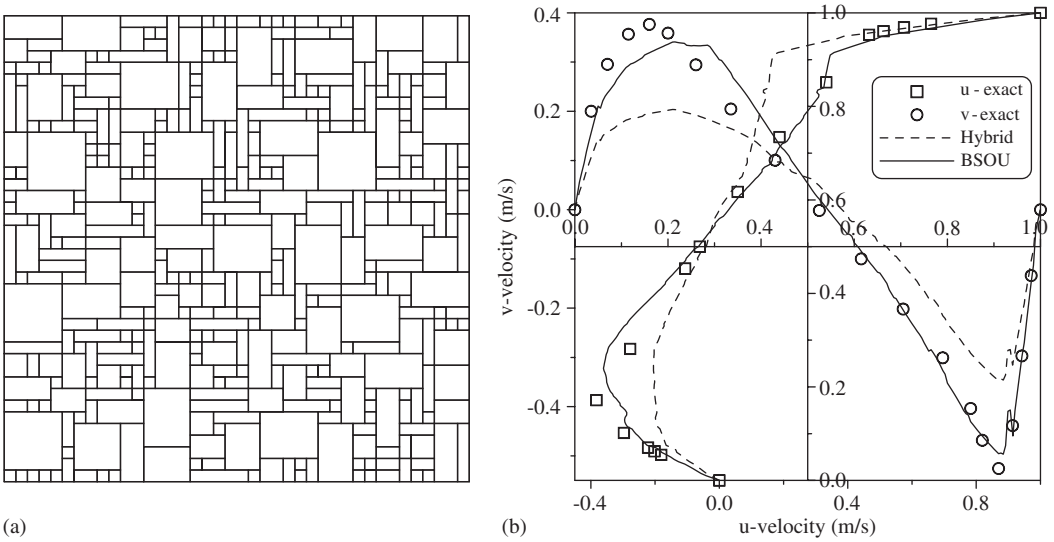


Figure 7. Indicative results for the square driven cavity, $Re = 1000$: mesh structure (a); u and v velocity profiles along the cavity vertical and horizontal mid-lines, respectively (b).

where N is the number of control volumes of the arbitrary grid, U and V are the resulting velocity components at the centre of each such volume i having dimensions $h_{i,x}$ and $h_{i,y}$, u_{exact} and v_{exact} are the corresponding velocities produced at the same location from the accurate solution, U_{lid} is the driven velocity and A_{lid} the square cavity area.

Due to the random grid generation method, arbitrary grids of the same maximum aspect ratios may produce quite different results, although they have almost equal number of cells. Hence it is possible for a coarser grid to give similar or even better results than a finer grid, as in the example of Figures 7(b) and 6(b), respectively. For this reason a number of 100 arbitrary grids were constructed for each of the seven spacing classes examined (maximum permissible aspect ratios from 2:1 to 8:1), and their results are used to obtain the statistical behaviour of the various models.

In order to estimate the value of a dependent variable required at certain locations inside an arbitrary grid, as for example along the horizontal or the vertical centreline of the square cavity to draw the curves of Figures 6(b) and 7(b), a 4-point general interpolation method was developed, as shown in Figure 8. According to the method the Cartesian co-ordinate system centred at a point P divides the 2D domain into four quarters. The algorithm finds the nearest to P point in each quarter and forms a quadrilateral 1234 that contains the point P. Then the following interpolating relation is applied:

$$\phi_{\text{P}} = \frac{\Phi_1 x_{\text{b}} y_{\text{c}} + \Phi_2 x_{\text{d}} y_{\text{c}} + \Phi_3 x_{\text{d}} y_{\text{a}} + \Phi_4 x_{\text{b}} y_{\text{a}}}{(x_{\text{b}} + x_{\text{d}})(y_{\text{a}} + y_{\text{c}})} \tag{24}$$

which is an extension of the conventional interpolation in a 4-point orthogonal system. The co-ordinates x and y of the points a, b, c, d are taken as absolute values.

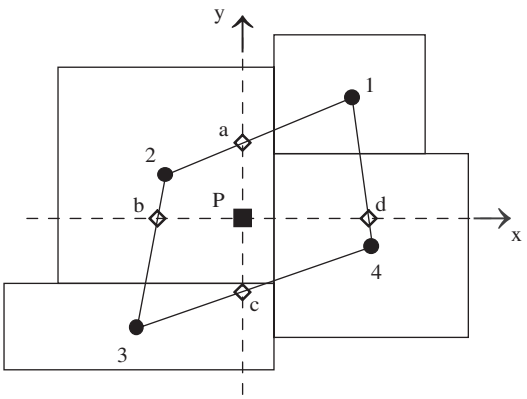


Figure 8. Schematic of the 4-point interpolation method.

The percentage error averaged over 100 arbitrary grids is plotted in Figure 9(a) for various discretization schemes and sub-models, as function of the spacing of an 'equivalent' uniform grid having about the same number of control volumes. Moreover, Figure 9(b) displays the corresponding error standard deviation that can be considered as a reliability index: lower error deviation over different grids of equal density implies a more predictable and consistent behaviour of a model.

The results of the BSOU scheme exhibit as expected much lower error than the Hybrid scheme ones, for all the examined grids. Only in cases of very irregular and coarse meshes the performance of both schemes become similar (Figure 9(a)). For the BSOU scheme, increasing the accuracy of the remote point value calculation, using Equation (18) instead of (17) or (16), both the average error and its standard deviation are reduced (Figures 9(a) and 9(b)). On the other hand, from the same figures it appears that the interpolation technique of Equation (8) is more advantageous over the area-weighted formula (7) for the calculation of cell-face values in both discretization schemes. Hence the combination of Equations (8), and (18) with the BSOU scheme achieves always the lowest error and standard deviation and, in addition, it needs in general less iterations of the algorithm to converge. The latter are less than half with the Hybrid scheme, which also shows less error standard deviation, especially for the coarser grids (Figure 9(b)). However, such very coarse and irregular meshes would not be used in practical applications.

Concerning the computer time requirements, the use of Equations (17) or (18) instead of (16) increases the CPU-time per grid node and per iteration by about 3% or 7.5%, respectively, whereas the implementation of the iterative formula (8) needs 7–9% more computing time than Equation (7). Generally speaking, the most accurate combination (BSOU—Equations (8) and (18)) requires about 20% more CPU-time per node and per iteration than the simplest one (Hybrid—Equation (7)). However a more elaborate model converges faster to a solution with a prescribed error. For example, to reduce the average error below 4% in Figure 9 the Hybrid scheme needs 17 CPU-seconds, whereas the next four combinations in the legend with the BSOU scheme require 9.7, 7.6, 7.1 and 6.2 CPU-seconds, respectively, and this gain increases as the error decreases.

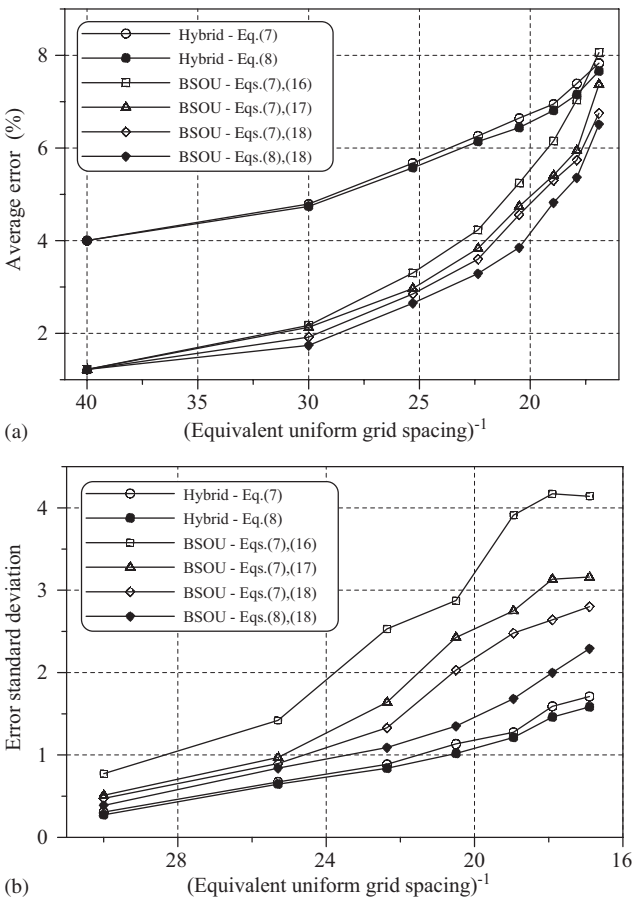


Figure 9. Statistical performance of the algorithm on unstructured grids with various aspect ratios: average percentage error of the results (a), and average error standard deviation (b).

The iterative solution algorithm showed a quite stable behaviour in all the examined grids and models. Only in few grids with large spacing (aspect ratio above 6:1) the algorithm failed to converge even with the more stable Hybrid scheme. Similar was the model performance and the results obtained for some different Reynolds numbers tested (400 and 3000).

4.2. Adaptive grids in a step profile transport

The simple but very stringent problem of the transport of a step change in a uniform velocity field is another classical test for assessing discretization schemes (e.g. Reference [22]) or gridding methods (e.g. Reference [9]). Here it was used to evaluate the present methodology that combines both higher-order discretization and mesh refining. The transport equation (1) for a scalar quantity Φ is solved; the diffusivity Γ_Φ equals the fluid viscosity, and $S_\Phi = 0$. The square domain of Figure 10(a) is divided in two parts and the boundary value of Φ equals unity on the upper-left part and zero on the other boundaries. The velocity vector was skewed

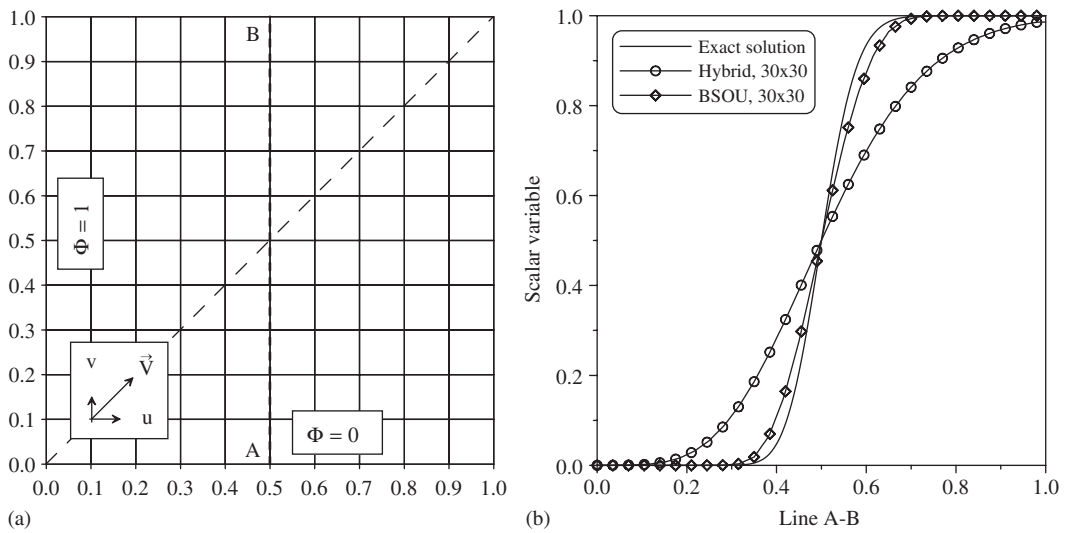


Figure 10. Step profile transport for flow angle 45° and $Re = 500$: geometry and boundary conditions (a); solutions on the vertical centreline AB using a regular grid (b).

to an angle of 45°, at which the false diffusion maximizes [9], and the numerical results were obtained for $Re = 500$. The comparison of the Hybrid and BSOU schemes as well as of the various gridding techniques used is based on the numerical results obtained along the line AB (Figure 10(a)).

Such results are indicatively plotted in Figure 10(b) for a 32×32 nodes regular equispaced grid, which is the first and simplest gridding method tested. The superiority of the BSOU scheme is evident in this figure against the Hybrid scheme that exhibits considerable false diffusion error.

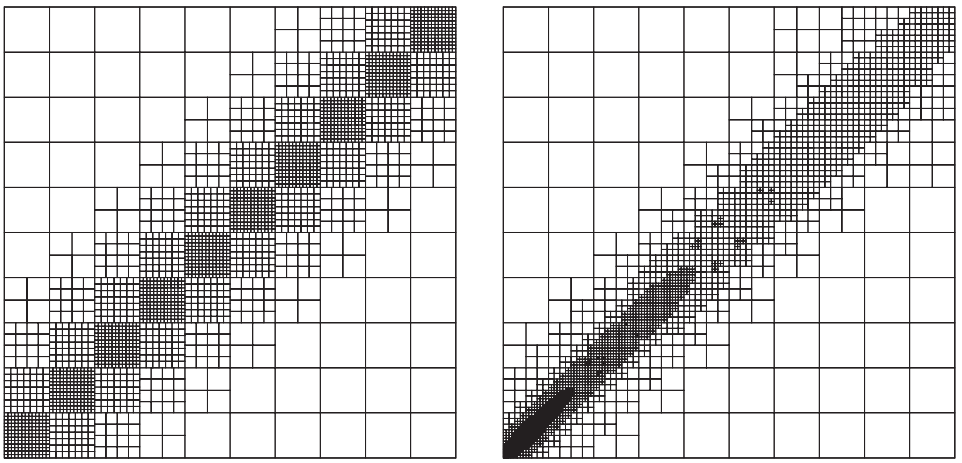
A simple criterion for grid refinement is adopted, taking into account the maximum gradient of the scalar variable in each control volume:

$$C_r = \max[(\partial\Phi/\partial x)_P, (\partial\Phi/\partial y)_P] \tag{25}$$

and according to that, the refinement ratio of a cell i is given by the following expression:

$$R_i = K \frac{C_{r,i}}{(1/N)\sum_{i=1}^{N} C_{r,i}} \tag{26}$$

where N is the total number of grid cells, and K an adjustable number. The above criterion was then applied in two ways: with the first method the final grid is constructed in one step and before the solution, assuming that $\Phi = 1$ everywhere in the upper triangle of the domain, $\Phi = 0$ in the lower triangle and $\Phi = 0.5$ at the diagonal line. In the second, solution adaptive technique, a multilevel refinement is performed in several steps based each time on the results of the previous solution and using always a maximum refinement ratio of 2. In both methods the coefficient K in Equation (26) is properly regulated to produce the desired total number of nodes in the final grid. Figure 11 presents indicative grid arrangements generated by the above pre-refined and solution adaptive methods. Refinement ratios different than 2 are used



(a) (b)

Figure 11. Examples of refined grids constructed for the step profile problem: pre-refined, 4066 nodes (a); solution adaptive, 3073 nodes (b).

in many of the first method grids, as can be seen in Figure 11(a). Also, a grid-smoothing algorithm was applied in both methods to reduce the number of neighbouring cells and thus the computer storage requirements. In all cases the initial grid is an equispaced 10×10 mesh.

The average error is computed by a norm similar to (23):

$$\text{Error}(\%) = \frac{100}{A} \sum_{i=1}^{N} |\Phi_i - \Phi_{\text{exact}}| A_i \tag{27}$$

where A_i is the area of grid cell i and A is the total computational area. Average and maximum errors of the various numerical solutions obtained for this problem are concentrated in Figures 12(a) and 12(b), respectively. It is obvious that for all three gridding techniques the BSOU scheme produces much more accurate results (almost one order of magnitude less average error) than the Hybrid scheme. The latter shows enhanced accuracy only when the maximum Peclet number approaches the value of 2 (below which it reverts to second-order central differences). On the other hand, it can be observed that for both discretization schemes the use of pre-refined or solution-adaptive grids reduces considerably and almost equally the average and the maximum error. The log–log slope of Figure 12(a) curves, which is an indication of the order of accuracy [12], increases from about -0.9 in case of a regular grid with the Hybrid scheme to about -1.5 with the BSOU, and reaches -2.5 when the latter is combined with a refined or adaptive grid. Moreover, Figure 12 reveals that at least for this particular problem the use of higher-order discretization in a regular grid is clearly better than a low-order scheme with local grid refinement. If however higher-order discretization is applied to a refined or adaptive grid following the present methodology, the accuracy can be remarkably improved. For example, the results obtained with grids of about 1024 control volumes (32×32 regular) exhibit an average error of 9% for the combination Hybrid scheme-regular grid, 5% for the Hybrid-refined, 1.4% for the BSOU-regular, and only 0.36% for the BSOU-refined model (Figure 12(a)) whereas the corresponding maximum errors are 23%,

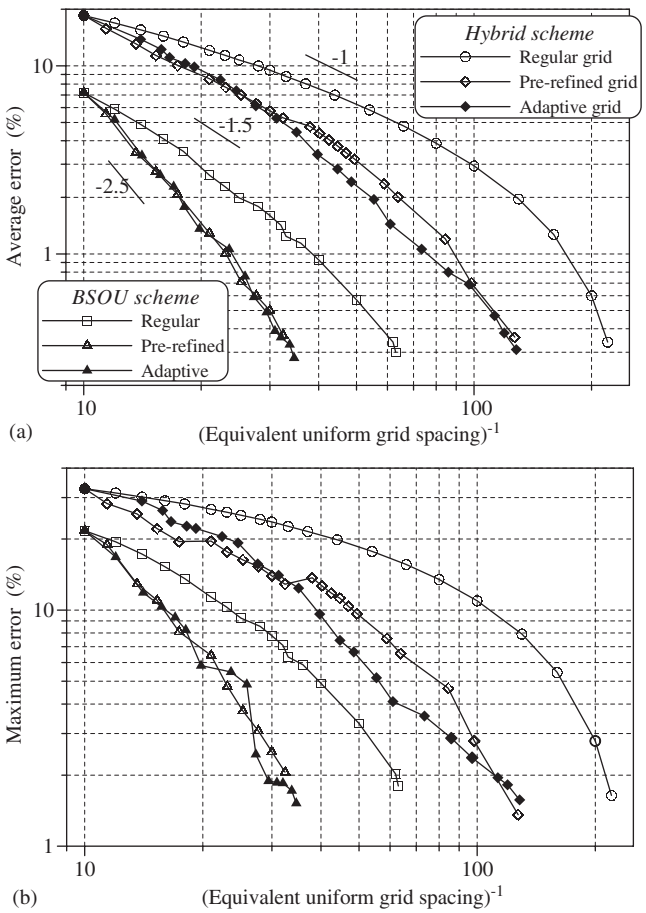


Figure 12. Comparative view of the step transport results accuracy: average percentage error (a); maximum percentage error (b).

13%, 7.1%, and 1.85% (Figure 12(b)). Figures 10(b) and 13 display the resulting values of Φ along the line AB (Figure 10(a)) that correspond to the above example. Figure 13 contains also the numerical results of Reference [9] taken with a 1474 nodes embedding grid and the Hybrid scheme, which are of similar accuracy with the present ones with the same scheme. It is apparent that the new methodology (BSOU-refined) provides an almost grid-independent solution for a relatively small number of control volumes.

Finally, the use of refinement ratios different than 2 does not affect neither the stability nor the accuracy of the algorithm, as can be seen from the pattern of the 'pre-refined grid' curves in Figure 12, that is similar to the 'adaptive grid' curves obtained with a refinement ratio of 2.

4.3. Adaptive grids in the lid-driven square cavity

The lid-driven cavity case is used again in order to verify the previously showed advantages of the new methodology in a more realistic, fluid flow problem. An initial solution was first

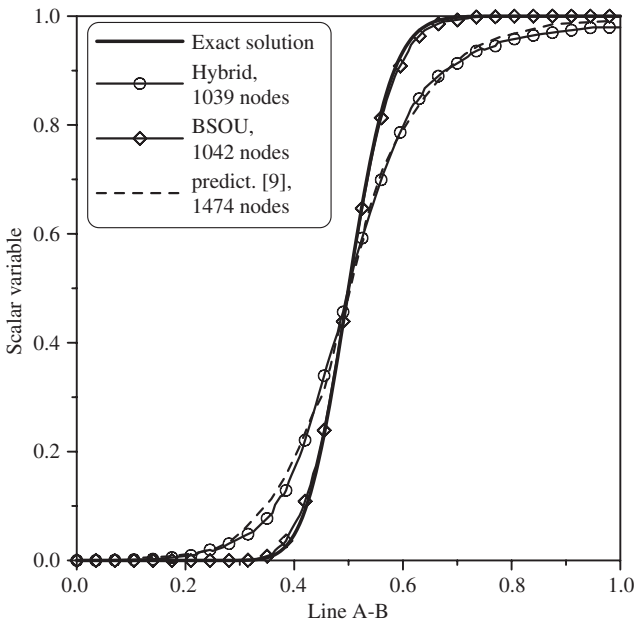


Figure 13. Step profile transport results on the vertical centerline using a solution-adaptive grid.

obtained for $Re = 1000$ on a 20×20 uniform grid, and based on these results several different adaptive grids were constructed in a single-step process. The gradient indicator is adopted again as grid refinement criterion (similar to Equation (25)):

$$C_r = \max[(\partial u/\partial x)_P, (\partial u/\partial y)_P, (\partial v/\partial x)_P, (\partial v/\partial y)_P] \tag{28}$$

The refinement ratio of each cell is calculated from Equation (26) and its maximum value is allowed to vary from 2 to 5 in order to check the accuracy of the results for grids with various numbers of nodes. An indicative such grid with a maximum refinement ratio of 5 is illustrated in Figure 14. The grid-smoothing algorithm was not activated, in order to examine the performance of the method on cells with large interface ratios.

The average and the maximum error curves of the results produced by the various tested grids are concentrated for comparison in Figure 15. The corresponding curves for a regular grid are included in the same figure. Once again the Hybrid scheme shows much reduced accuracy comparing to the BSOU and moreover, its results are worse than those of a regular equivalent mesh. This is because the truncation error term becomes first-order at the interface of cells with different refinement [12]. On the contrary, the gain in accuracy with the BSOU-adaptive model is considerable comparing to a regular grid.

No convergence problems were observed in any of the previous solutions, however the number of iterations for convergence was varied. Therefore the real gain in computations can be estimated when the accuracy of the obtained results is correlated with the CPU-time requirements. From this point of view, Figure 16 shows that the BSOU-adaptive grid can reduce the computer time requirements more than 50% of the corresponding regular grid.

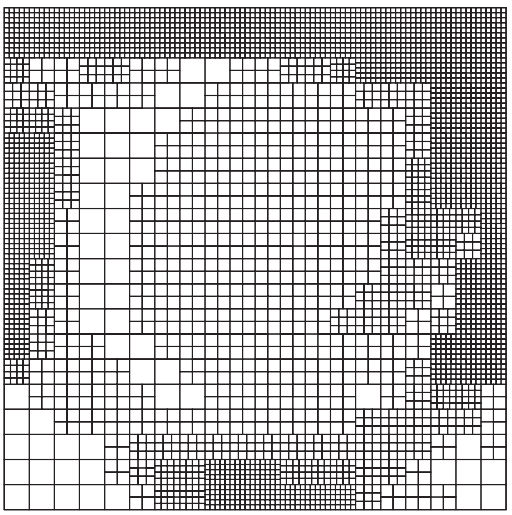


Figure 14. Example of an adaptive grid constructed in the square cavity.

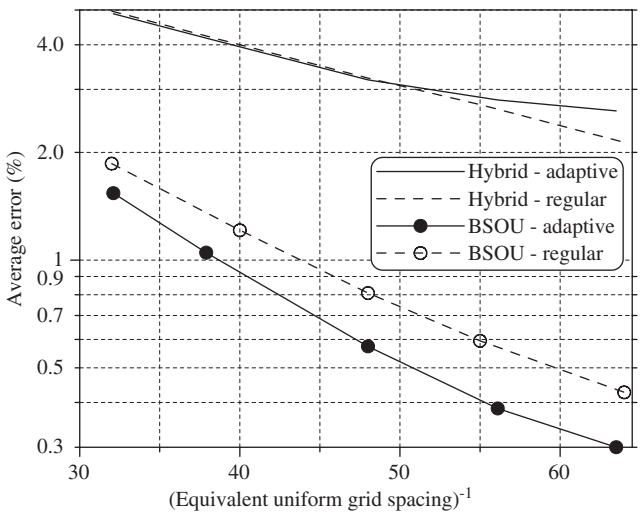


Figure 15. Comparative view of the square cavity results accuracy.

Obviously this gain would be even greater if a better refinement criterion were used and a solution-adaptive optimum grid was constructed.

4.4. Flow over a backward-facing step

The backward-facing step flow is another commonly used case for validation of CFD methods and techniques. Detailed measurements in a 2D channel are presented in Reference [26]. A sketch of the flow configuration is included in Figure 17, where the measured reattachment length is compared with the predictions of [26] and of the present model. The experiments

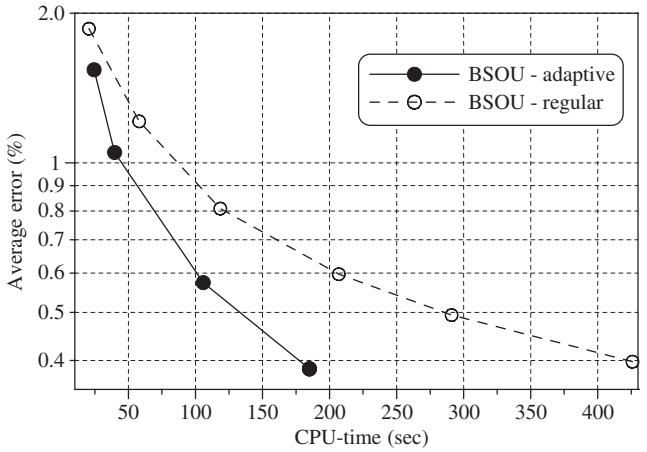


Figure 16. Accuracy versus CPU-time requirements for the square cavity problem.

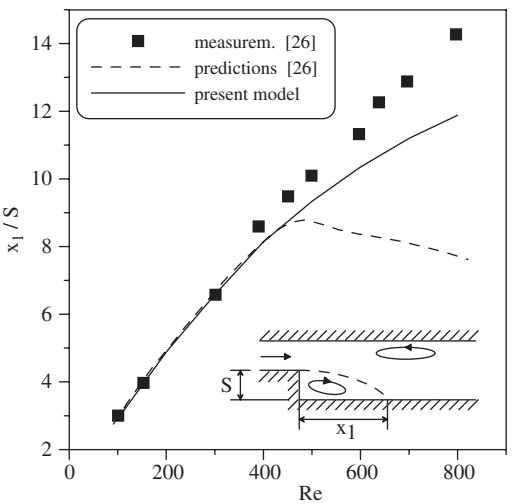


Figure 17. Variation of the reattachment length with the Reynolds number in the backward-facing step problem.

showed that for $Re > 400$ the flow downstream of the step does not remain two-dimensional due to the appearance of additional separation regions. However, the great discrepancy of the predictions of Reference [26] shown in Figure 17 is mainly caused by the increasing with the Re false diffusion error, since a relatively coarse mesh and a first-order differencing scheme were used. The grid-independent, 'exact' results obtained by the present algorithm using a fine enough mesh (25 000 to 40 000 nodes) and the BSOU scheme agree well with the measurements up to about $Re = 400$, while exhibit a more reasonable behaviour for higher Re (Figure 17).

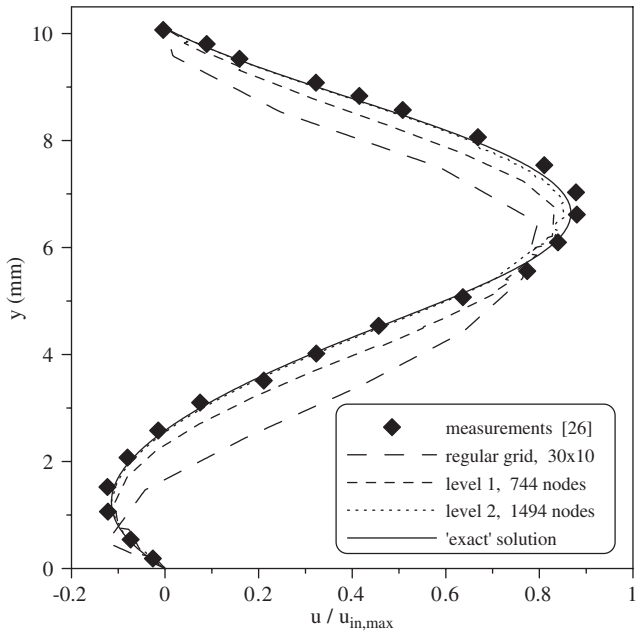


Figure 18. Comparison of axial velocity profiles at $x/S = 4.18$ for $Re = 389$.

A solution-adaptive strategy is adopted for a multilevel grid refinement, using in each level a refinement ratio of 2. The gradient criterion for refinement (Equation (28)) is used here as well, but separately along the x and the y direction, allowing thus for a grid cell to divide in either two or four parts. All calculations start from a uniform regular grid having resolution $S/2$ and $S/5$ in the x and y direction, respectively, where S is the step height. No-slip conditions are applied to all the channel walls and a fully developed parabolic velocity profile is imposed at the inflow.

Figure 18 shows an indicative comparison between predicted and measured velocity profiles at $x/S = 4.18$, for $Re = 389$. The coarse initial grid produces significant error, but two adaptation levels are enough to obtain results of adequate accuracy. The average error is computed again from Equation (23), replacing the term $2U_{\text{lid}}$ with the bulk inflow velocity, and falls below 1.5% after these two calculation levels. Using a first-order scheme, as the Hybrid one here, the above accuracy is achieved only after five adaptation levels and a final grid having about four times more cells (see also Reference [4]).

For higher Reynolds numbers the flow field becomes more complicated, containing additional separation zones, as shown in Figure 19(a) for $Re = 800$. Some computational grids constructed at different adaptation levels are indicatively drawn in Figures 19(b), 19(c), and the average error is plotted in Figure 20 as function of the number of grid cells. Although in this case the refined grid regions cover almost all the computational domain, the superiority of combining grid refinement with higher-order differencing is evident again. The log–log slope of this curve is about -2.8 in comparison with -2.4 of the regular grid—BSOU curve. On the other hand, although starting from the same solution (taken with the BSOU scheme), the present adaptation approach seems rather inefficient when the first-order Hybrid scheme is used

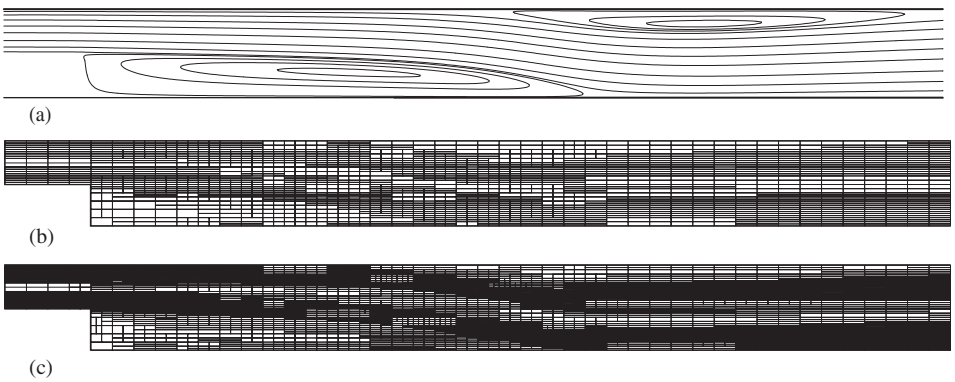


Figure 19. Backward-facing step results for $Re = 800$: computed streamlines (a); adaptive grid at second level, 2250 nodes (b); and at third level, 4380 nodes (c).

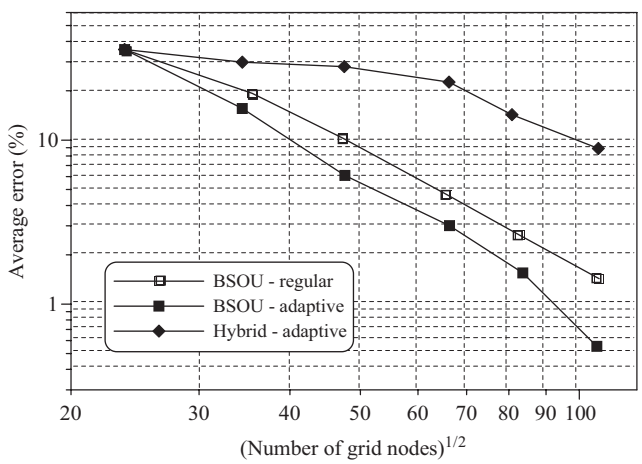


Figure 20. Comparative view of backward-facing step results accuracy for $Re = 800$.

in the next levels. The same performance is also observed in the corresponding predictions of the reattachment length. Finally, it should be noted here again that the gain in accuracy with the new simulation methodology could be even higher if an improved grid adaptation technique was used (e.g. better refinement criterion, grid smoothing and merging techniques).

5. CONCLUSIONS

The implementation of higher-order discretization in unstructured or arbitrarily refined Cartesian grids was achieved by a new methodology analysed in this paper. Two alternative methods were tested to generate the auxiliary cell-face variables in such grids. These variables were

introduced in various interpolation techniques and multiple-scale stencils in order to improve the accuracy of the discretized form of the governing equations, as well as to utilize data from remote cells. Although computer memory consuming, the storage of the cell-face data exhibits important benefits concerning the simplicity, robustness and generality of the numerical algorithm. The latter was found to be very stable and efficient when applied in some selected classical 2D problems.

From the numerical studies performed in this work it was revealed that a locally refined grid with low-order discretization is inferior in accuracy to a regular mesh with a higher-order scheme. However, combining the advantages of grid adaptation and higher-order discretization, the present methodology can produce quite better results than each of the above techniques alone for the same computational effort.

Although the bounded second-order upwind (BSOU) scheme is used throughout this study, several other second or higher-order differencing schemes can be treated in a similar way and easily incorporated in the algorithm. Also, the method is applicable to turbulent flows as well, and some preliminary tests with the k-ε turbulence model showed a stable and reliable performance. Finally, an extension to three dimensions is quite straightforward and this work is currently underway.

REFERENCES

1. Soni BK. Grid generation: Past, present, and future. *Applied Numerical Mathematics* 2000; **32**:361–369.
2. Kallinderis Y. A 3-D finite-volume method for the Navier-Stokes equations with adaptive hybrid grids. *Applied Numerical Mathematics* 1996; **20**:387–406.
3. Muzaferija S, Gosman D. Finite-volume CFD procedure and adaptive error control strategy for grids of arbitrary topology. *Journal of Computational Physics* 1997; **138**:766–787.
4. Wang ZJ. A quadtree-based adaptive Cartesian/quad grid flow solver for Navier-Stokes equations. *Computers and Fluids* 1998; **27**:529–549.
5. Yung C, Keith T, Witt K. Numerical simulation of axisymmetric turbulent flow in combustors and diffusers. *International Journal for Numerical Methods in Fluids* 1989; **9**:167–183.
6. Anagnostopoulos J, Bergeles G. Modeling of the flow and the interface surface in a continuous casting mold model. *Metallurgical and Materials Transactions B* 1999; **30**:1095–1105.
7. Atta EA, Vadyak A. A grid overlapping scheme for flowfield computations about multicomponent configurations. *AIAA Journal* 1983; **21**:1271–1277.
8. *CRC Handbook of Grid Generation*. Thompson JF, Soni BK, Weatherill NP (eds). CRC Press: Boca Raton, FL, 1999.
9. Coelho P, Pereira JCF, Carvalho MG, Calculation of laminar recirculating flows using a local non-staggered grid refinement system. *International Journal for Numerical Methods in Fluids* 1991; **12**:535–557.
10. Coelho PJ, Argain J. A local grid refinement technique based upon Richardson extrapolation. *Applied Mathematical Modelling* 1997; **21**:427–436.
11. Papadakis G, Bergeles G. A local grid refinement method for three-dimensional turbulent recirculating flows. *International Journal for Numerical Methods in Fluids* 1999; **31**:1157–1172.
12. Bennett BV, Smooke MD. Local rectangular refinement with application to nonreacting and reacting fluid flow problems. *Journal of Computational Physics* 1999; **151**:684–727.
13. Coelho PJ, Pereira JCF. Calculation of a confined axisymmetric laminar diffusion flame using a local grid refinement technique. *Combustion Science and Technology* 1993; **92**:243–253.
14. De Lange HC, De Goey LPH. Numerical flow modelling in a locally refined grid. *International Journal for Numerical Methods in Engineering* 1994; **37**:497–515.
15. Thompson MC, Ferziger JH. An adaptive multigrid technique for the incompressible Navier-Stokes equations. *Journal of Computational Physics* 1989; **82**:94–121.
16. Edwards MG. Elimination of adaptive grid interface errors in the discrete cell centered pressure equation. *Journal of Computational Physics* 1996; **126**:356–372.
17. Pember RB, Bell JB, Colella P, Crutchfield WY, Welcome ML. An adaptive Cartesian grid method for unsteady compressible flow in irregular regions. *Journal of Computational Physics* 1995; **120**:278–304.
18. Fuchs L. A local mesh-refinement technique for incompressible flows. *Computers and Fluids* 1986; **14**:69–81.

19. Berger MJ, Colella P. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics* 1989; **82**:64–84.
20. Calhoun D, LeVeque RJ. A Cartesian grid finite-volume method for the advection-diffusion equation in irregular geometries. *Journal of Computational Physics* 2000; **157**:43–180.
21. Udaykumar HS, Mittal R, Rampunggoon P, Khanna A. A sharp interface Cartesian grid method for simulating flows with complex moving boundaries. *Journal of Computational Physics* 2001; **174**:345–380.
22. Papadakis G, Bergeles G. A locally modified second order upwind scheme for convection terms discretization. *International Journal of Numerical Methods for Heat and Fluid Flow* 1995; **5**:49–62.
23. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. Hemisphere, Washington, DC, 1980.
24. Rhie CM, Chow WL. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal* 1983; **21**:1525–1532.
25. Ghia U, Ghia KN, Shin CT. High-Re solution for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; **48**:387–411.
26. Armaly BF, Durst F, Pereira JCF, Schönung B. Experimental and theoretical investigation of backward-facing step flow. *Journal of Fluid Mechanics* 1983; **127**:473–496.

