ACADEMIC PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 184 (2003) 1–36

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A representation of curved boundaries for the solution of the Navier–Stokes equations on a staggered three-dimensional Cartesian grid

M.P. Kirkpatrick*, S.W. Armfield, J.H. Kent

School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, Sydney 2006, Australia

Received 26 September 2001; received in revised form 5 September 2002; accepted 26 September 2002

Abstract

A method is presented for representing curved boundaries for the solution of the Navier–Stokes equations on a non-uniform, staggered, three-dimensional Cartesian grid. The approach involves truncating the Cartesian cells at the boundary surface to create new cells which conform to the shape of the surface. We discuss in some detail the problems unique to the development of a cut cell method on a staggered grid. Methods for calculating the fluxes through the boundary cell faces, for representing pressure forces and for calculating the wall shear stress are derived and it is verified that the new scheme retains second-order accuracy in space. In addition, a novel ''cell-linking'' method is developed which overcomes problems associated with the creation of small cells while avoiding the complexities involved with other cell-merging approaches. Techniques are presented for generating the geometric information required for the scheme based on the representation of the boundaries as quadric surfaces. The new method is tested for flow through a channel placed oblique to the grid and flow past a cylinder at $Re = 40$ and is shown to give significant improvement over a staircase boundary formulation. Finally, it is used to calculate unsteady flow past a hemispheric protuberance on a plate at a Reynolds number of 800. Good agreement is obtained with experimental results for this flow.

Keywords: Navier–Stokes; Boundary condition; Complex geometry; Cartesian grid; Cartesian mesh; Embedded boundary; Cut cell; Partial cell; Shaved cell; Computational fluid dynamics

1. Introduction

The numerical solution of the equations of fluid dynamics is greatly simplified if the discretisation is performed in a Cartesian coordinate system. Many flow simulations, however, involve complex geometries with curved and planar boundaries oblique to the grid. In a Cartesian coordinate system such boundaries are generally approximated as a series of staircase steps. Unless a very fine grid is used, the predicted velocity and pressure fields adjacent to the boundary will be inaccurate.

* Corresponding author. Tel.: +1-650-604-3932; fax: +1-650-604-0841.
E-mail address: michael.kirkpatrick@aeromech.usyd.edu.au (M.P. Kirkpatrick).

0021-9991/02/$ - see front matter
PII: S0021-9991(02)00013-X

A number of other types of grid are currently used for solving the equations in complex geometries, including overlapping orthogonal grids (Chimera grids) [1,2]; flexible mesh embedding (FAME) [3]; curvilinear boundary-fitted grids [4–7] and unstructured grids [8,9]. While these approaches simplify the implementation of boundary conditions, each introduces new difficulties such as extra terms in the equations, extra interpolations, larger computational molecules, and problems associated with the transfer of information across grid interfaces. This added complexity makes code development more difficult and increases computation time. Discussions of these techniques can be found in [10,11].

An alternative approach which has attracted increasing interest in recent years uses a Cartesian grid for all cells except those which are intersected by the boundary. These boundary cells are truncated so that they conform to the shape of the boundary surface. In this way, the advantages of a Cartesian grid are retained for the standard, non-boundary cells and a more complex treatment is only required for the boundary cells. Cells completely outside the flow domain are removed from the computations. This approach is referred to in the literature as the ''embedded boundary method'', the ''Cartesian grid method'' or the ''cut cell method''.

Considerable progress has already been made in developing algorithms for the solution of different types of equations based on this approach. In 1993, De Zeeuw and Powell [12] presented an adaptively refined Cartesian mesh solver for the Euler equations. In 1994, both Pember et al. [13] and Quirk [14] presented adaptive Cartesian mesh approaches for the solution of the Euler equations. In 1995, Coirier and Powell [15] investigated the accuracy of Cartesian mesh approaches for the Euler equations and in 1996 they presented a solution adaptive approach for both viscous and inviscid flows in two-dimensions. Yang and co-workers have focussed on Cartesian mesh methods for the Euler equations in compressible flow. In 1997 they presented a method for compressible flows for static and moving body problems [16,17], which they extended to three-dimensions in 2000 [18]. A similar approach was used by Causon et al. [19] in 2000 for the solution of the shallow-water equations. The group at Lawrence Berkeley National Laboratory have also made significant contributions to the field. In 1997, Almgren et al. [20] presented a Cartesian grid projection method for the incompressible Euler equations in complex geometries. In 1998, Johansen and Colella [21] presented a second-order accurate method for solving Poisson's equation on irregular two-dimensional domains. This approach was extended in 2001 by McCorquodale et al. [22] to the solution of the time-dependent heat equation.

To date, there has been relatively little work focussing on cut cell methods for the full Navier–Stokes equations. In 1999, Colella et al. [23] applied the method of Johansen and Colella cited above to free surface flows in three dimensions. They used the projection method of Bell et al. [24] to compute finite difference approximations and a volume of fluid method to represent the free surface. Udaykumar et al. [25–28] have published a number of papers dating from 1996 to the present in which they discuss their ''ELAFINT'' method for solving two-dimensional incompressible fluid flow problems in the presence of both irregularly shaped solid boundaries and moving/free-phase boundaries. The solver uses a finite volume formulation on a uniform, non-staggered grid with all gradients approximated using central differences. A related solver developed by Ye et al. [29] uses a similar formulation but includes an improved interpolation scheme at the boundaries and a fractional-step method for time advancement. While Ye et al. show that their interpolation scheme is itself second-order accurate, in their test for the accuracy of the overall scheme they use simulations of Wannier flow in which there is no contribution from the advection terms. A survey of the literature yielded only two papers presenting cut cell methods for the Navier–Stokes equations on a staggered grid: the work of Tau [30] presented in 1994, and Tucker and Pan [31] in 2000. In both cases, only two-dimensional problems are addressed, and the accuracy of the discretisation at the boundary appears to be first-order.

In this paper we present a second-order accurate Cartesian cut cell method for the Navier–Stokes equations on a three-dimensional, non-uniform, staggered grid. The method uses central-differences for the viscous and pressure gradient terms and either central-differences or the QUICK scheme of Leonard [32]

for the advection terms. Solid boundaries are defined as quadric surfaces in a Cartesian coordinate system. Fluxes are calculated using the area of the truncated cell faces and interpolated values of variables at the face centroids. Body forces are calculated using the free volume of the cell and shear forces at the wall are calculated using the correct node height and wall surface area. The treatment of the boundary cells is similar to that used for cells in a non-orthogonal grid and requires similar corrections to the fluxes in order to account for the non-orthogonal geometry and ensure that the discretisation retains second-order accuracy. The corrections in the present formulation, however, are simplified by exploiting the presence of the boundary and are designed to fit easily into an existing Cartesian discretisation scheme.

The use of a staggered grid for the solution of the Navier–Stokes equations has a number advantages which have made it popular for the solution of incompressible flow problems. In particular, the use of a staggered grid overcomes numerical problems associated with pressure–velocity coupling which occur when a colocated grid is used. The representation of curved boundaries on a staggered Cartesian grid, however, entails complexities not present when a non-staggered or colocated grid is used.

On a non-staggered grid, not only are the velocity and pressure nodes colocated, but the position and geometry of the associated cells is also identical. With a staggered grid, the pressure cell and the cells associated with each of the three velocity components are at a different location and will generally have a different shape when they are cut by an embedded boundary. A cut cell scheme for a staggered grid must deal with this extra complexity in a consistent manner.

The pressure correction procedure by which mass conservation is enforced is also different for staggered and colocated arrangements, leading to a number of complications in the staggered case which do not occur when a colocated grid is used. Firstly, on a staggered grid the placement of the velocity node is dictated by its role in the pressure correction equation—it must be placed at the centroid of the cut face of the pressure cell if the advantages of using a staggered grid are to be maintained. This is discussed in more detail in Section 2.3. Secondly, it is possible that a velocity node in a boundary cell may not have two pressure nodes associated with it, making it impossible to enforce conservation of mass at that velocity node. This problem is dealt with using a cell-linking process as discussed in Section 2.3.4.

A final complication in the case of a staggered grid involves the calculation of the advective fluxes, as different interpolations are required for the velocity components parallel and perpendicular to the flux direction.

For the purpose of this paper, boundaries are defined using quadric surfaces. Quadric surfaces were chosen because they can be represented mathematically in a concise manner. In Section 2.4 we present methods for specifying boundaries as quadric surfaces and for calculating the geometric information required for the implementation. The authors are currently working on an extension of the scheme in which boundaries are defined using the surface triangulation method of Aftosmis et al. [33]. While this will enable our scheme to be used for arbitrary geometries, a detailed description is beyond the scope of the work presented here.

A common problem with cut cell methods is the creation of very small cells. This leads to problems with stiffness of the equations and, in the case of time-dependent simulations, problems with numerical stability. Researchers have dealt with this in a number of ways, including using a volume-of-fluid approach such as that presented by Almgren et al. [20] or cell-merging as presented by Ye et al. [29]. A disadvantage of cell-merging is that it generally entails a considerable increase in complexity as fluxes between diagonally adjacent cells must also be calculated and the computational molecule for merged boundary cells becomes different to that used for the standard cells. There are also significant problems associated with the formulation of a systematic merging algorithm in three-dimensions. In this paper we present a novel cell merging approach which circumvents these problems. Rather than merging two cells to form a single cell, the two cells are linked as a "master/slave" pair in which the two nodes are coincident while each cell remains a distinct entity. As a result, the fluxes, wall shear stress, volumetric and surface information are calculated in exactly the same way for the master and slave cells as they are for the standard boundary cells.

In fact, the same routines are used for calculating the fluxes for master/slave boundary cells, standard boundary cells, and non-boundary cells and the computational molecule for all boundary cells remains the same as that used for the non-boundary cells. The only differences in the treatment of these three classes of cells occur in the pre-processing stage, during which the nodes are relocated and the interpolation factors, volumes and areas are calculated. This feature of our approach greatly simplifies the implementation.

In the following, we describe our formulation in detail. We describe the interpolation corrections required for the boundary cells, methods of generating the required geometric information, and the cell-linking procedure. The spatial accuracy of the scheme is tested by calculating the flow in a driven cavity containing a circular obstruction and is shown to be second-order. A comparison of the cut cell and staircase boundary approaches is carried out for flow in a channel skewed to the grid, and for flow past a circular cylinder, where it is shown that the cut cell approach is considerably more accurate. Finally, results are obtained for three-dimensional, unsteady flow over a hemisphere and it is shown that our method is capable of accurately simulating this complex flow.

2. Method

The cut cell method was implemented in the PUFFIN code (Particles IN Unsteady Fluid Flow) [34] which uses a control volume formulation on a staggered, non-uniform, Cartesian grid to solve the Navier–Stokes equations. The Navier–Stokes equations for an incompressible fluid are

$$\frac{\partial \varrho}{\partial t} + \frac{\partial(\varrho u_j)}{\partial x_j} = 0, \tag{1}$$

$$\frac{\partial \varrho u_i}{\partial t} + \frac{\partial(\varrho u_i u_j)}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] - \varrho g_i, \tag{2}$$

where u_i are the Cartesian components of the velocity vector $\boldsymbol{u} = (u_1, u_2, u_3)$, x_i are the Cartesian coordinates of the position vector $\boldsymbol{x} = (x_1, x_2, x_3)$, P is the pressure, μ is the dynamic viscosity, ϱ is the density, and g_i is the acceleration due to gravity. Cartesian tensor notation with the convention of summation over repeated indices is used here and throughout this paper where general results are presented. Where specific cases are considered, however, it is clearer to dispense with the indices and results are presented in terms of the vectors $\boldsymbol{u} = (u, v, w)$ and $\boldsymbol{x} = (x, y, z)$. Also in the interests of clarity, it is assumed that the viscosity and density are constant, and the force due to gravity, ϱg_i is dropped. The arguments followed in the formulation remain the same if buoyancy and variable viscosity and density are included.

2.1. Advancement of equations in time

For unsteady calculations, the momentum equations (2) are discretised in time using a second-order hybrid scheme. Advection terms are treated explicitly using second-order Adams–Bashforth while diffusion terms are treated implicitly using second-order Adams–Moulton. The equations are advanced in time using a fractional step method. The pressure correction method [35,24] was found to be the fastest of the methods tested by Armfield and Street [36] and is the method used here. Gresho [37] has shown analytically that this method is second-order accurate in time.

The second-order Adams–Bashforth/Adams–Moulton scheme for the momentum equations is written as

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\delta t} + \left[\frac{3}{2}H(\boldsymbol{u}^n) - \frac{1}{2}H(\boldsymbol{u}^{n-1})\right] = -Gp^{n-1/2} + \left[\frac{1}{2}L(\boldsymbol{u}^*) + \frac{1}{2}L(\boldsymbol{u}^n)\right], \tag{3}$$

where H is the discrete advection operator, G is the discrete gradient, L is the discrete diffusion operator, and n is the time level. Integration of this equation yields an approximate solution for the velocity field, $\boldsymbol{u}^*$. Mass conservation is enforced through a pressure correction step in which the approximate velocity field is projected onto a subspace of divergence free velocity fields. The projection is achieved by solving a Poisson equation for the pressure correction, p', in which the source term is the divergence of velocity in each cell

$$Lp' = \frac{1}{\delta t} D\boldsymbol{u}^*. \tag{4}$$

The pressure correction is then used to correct the velocity field,

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \delta t G p', \tag{5}$$

and the pressure field,

$$p^{n+1/2} = p^{n-1/2} + p'. \tag{6}$$

Verification of the second-order time accuracy of the PUFFIN code, and a more detailed discussion of time-stepping schemes and the fractional step method used here can be found in [38]. The scheme is stable for Courant numbers ($C = \delta t u_i/\delta x_i$) up to $\approx$1.0, and a value of $C = 0.8$ was used in the unsteady simulations presented in Section 3.4. To calculate C for a cut cell, the dimension δx_i is the maximum cell width in the direction of the velocity component δu_i.

For steady-state calculations, a method of false time stepping is used. The momentum equations are advanced using the first-order implicit Euler scheme, with a time step given by $\delta t^* = \delta t U/L = 0.5$. Here δt is non-dimensionalised in terms of a representative velocity, U, and a representative length scale of the geometry, L.

2.2. Spatial discretisation

The equations are discretised in space using a finite volume formulation on a non-uniform, staggered, Cartesian grid. The second-order central difference scheme is used for diffusion terms, pressure gradient terms and the pressure correction equation. It is important to use the same discretisation for the pressure gradient in the momentum equations and the pressure correction in the pressure correction equation. This minimises the projection error and ensures convergence if an iterative scheme is used. The third-order accurate QUICK scheme of Leonard [32] is used for the advection terms.

The resulting discretised transport equation for a general variable ϕ using information at three time levels, $n+1$, n, and $n-1$, can be written

$$\begin{aligned} A_{\mathrm{P}}^{n+1}\phi_{\mathrm{P}}^{n+1} = {} & \sum_{nb}(A_{nb}^{n+1}\phi_{nb}^{n+1}) + S_{\mathrm{imp}}\phi_{\mathrm{P}}^{n+1} + S_{\mathrm{exp}}p^{n+1} + \left[\sum_{nb}(A_{nb}^{n}\phi_{nb}^{n}) - A_{\mathrm{P}}^{n}\phi_{\mathrm{P}}^{n} + S_{\mathrm{imp}}\phi_{\mathrm{P}}^{n} + S_{\mathrm{exp}}^{n}\right] \\ & + \left[\sum_{nb}(A_{nb}^{n-1}\phi_{nb}^{n-1}) - A_{\mathrm{P}}^{n-1}\phi_{\mathrm{P}}^{n-1} + S_{\mathrm{imp}}\phi_{\mathrm{P}}^{n-1} + S_{\mathrm{exp}}^{n-1}\right], \end{aligned} \tag{7}$$

where nb is a generic subscript for a neighbour cell, and S_{imp} and S_{exp} contain any further implicit and explicit sources respectively. Neighbouring cells are labelled using the following convention: East (E), North (N), and Up (U) correspond to the positive x, y, and z directions respectively (or x_1, x_2, and x_3 in index notation) and West (W), South (S), and Down (D) to the negative x, y, and z directions. The subscript P refers to the cell for which the flux is being calculated. Because a hybrid time-stepping scheme is used, the contributions from advection and diffusion terms to the coefficients will be different for the $n+1$, n, and $n-1$ time levels.

2.3. Boundary cells

The arrangement of grid cells and nodes close to a boundary is shown in Fig. 1. The cells intersected by the boundary, which were originally rectangular prisms, are truncated to fit the boundary. The area of the truncated cell faces is used to calculate fluxes and the truncated cell volume is used to calculate body forces. A method for calculating the face areas and volumes for a cell truncated by a quadric surface is presented in Section 2.4.

It is necessary to relocate the velocity nodes associated with the cut boundary cells. The velocity nodes are placed at the centre of the cut face of the pressure cell, which allows the pressure correction equation to be solved and mass conservation to be enforced for the boundary cells in the same manner as for the standard cells. The only difference is that the truncated area of the cell faces is used for the boundary cells. An alternative approach, whereby the velocity nodes are placed at the centroid of the new velocity cell, requires interpolation of the velocity to the face of the pressure cell for the calculation of mass fluxes in the pressure correction equation, and then back again. These extra interpolations reduce the accuracy of the mass conservation step and cause stability problems.

For incompressible flow, pressure only appears in the momentum equations as a gradient. As a result, there is no reason to move the pressure nodes and they are left in their original position, even though this may mean that they are physically outside the boundaries of the associated pressure cell.

The rearrangement of nodes and the changes in the shape of the cells means that, if the order of accuracy of the spatial discretisation is to be maintained, some modifications must be made to the way the fluxes are calculated. The approach used here has been to derive modifications which fit easily into the existing scheme for the Cartesian grid.

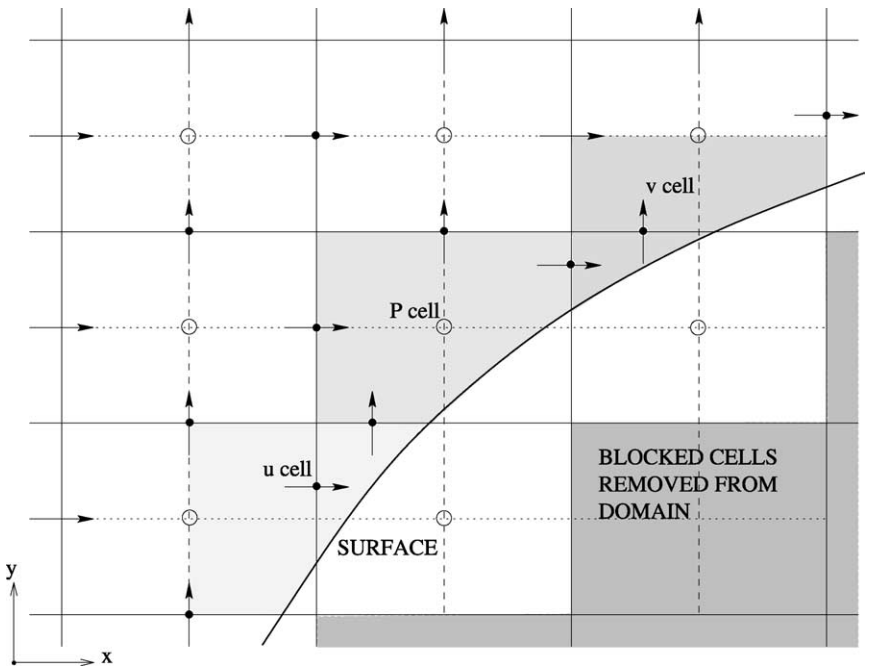


Fig. 1. Staggered grid with an oblique boundary in two dimensions. Cells are truncated by the boundary. Circles indicate pressure nodes. Horizontal arrows indicate the location of nodes for the u velocity component and vertical arrows the v velocity component. Examples of a u, v, and P cell at the boundary are highlighted.

2.3.1. Advective fluxes

The advective flux across a cell face is given by

$$F_{\text{adv}} = \left(\varrho u_{\perp} \Delta A u_j\right)_{\text{f}}, \tag{8}$$

where ϱ is the density, $u_{\perp}$ is the velocity normal to the face (the advecting velocity), u_j is the advected velocity component, and A is the area of the face. The subscript f indicates that these variables are evaluated at the cell face.

Advection of a velocity component in a direction parallel to the direction of the component is considered first. This situation is shown in Fig. 2. To reduce the complexity of the notation, in the remainder of this section, only the flux of the $u \equiv u_1$ component is considered. All results can be applied to other cases.

For standard cells, the velocity at the cell face is approximated using the third-order QUICK scheme [32]. The velocity at the centre of the east cell face is given by the interpolation formula

$$u_{\text{e}} = [(1-\theta)u_{\text{P}} + \theta u_{\text{E}}] - \frac{1}{8}\text{CRV} \times \Delta x_{\text{E}}^2, \tag{9}$$

where the weighting factor for the linear interpolation is

$$\theta = \frac{\Delta x_{\text{e}}}{\Delta x_{\text{E}}}, \tag{10}$$

and the upwind biased curvature term is

$$\text{CRV} = \frac{u_{\text{P}} - 2u_{\text{E}} + u_{\text{EE}}}{\Delta x_{\text{E}}^2}, \quad u < 0, \tag{11}$$

or

$$\text{CRV} = \frac{u_{\text{E}} - 2u_{\text{P}} + u_{\text{W}}}{\Delta x_{\text{E}}^2}, \quad u > 0. \tag{12}$$

A double subscript such as EE refers to the cell east of the eastern neighbour. The first term in Eq. (9) is the value of u at the cell face calculated using linear interpolation. The second term is an upwind-biased curvature term which makes the overall interpolation quadratic. If this term is not included the scheme reverts to the standard second-order central difference scheme.

The linear interpolation term accounts for the non-uniform grid through the weighting factor, θ, while the curvature terms have no grid weighting included. Castro and Jones [39] have shown that the uniform grid formula for QUICK gives negligible errors for grid expansion ratios ($r_x = \Delta x_{i+1}/\Delta x_i$) between 0.8 and 1.25, so this approach is suitable for standard cells.

Thus the advection flux of u across the east face for standard cells takes the form

$$F_{\text{adv}_{\text{e}}} = (\varrho u \Delta A)_{\text{e}} u_{\text{e}} = C_{\text{e}}[((1-\theta)u_{\text{P}} + \theta u_{\text{E}}) - S_{\text{QUICK}}], \tag{13}$$

where $C_{\text{e}} = (\varrho u A)_{\text{e}}$ is the advective flux contribution to the coefficients in the discretised conservation equation (Eq. (7)) and S_{QUICK} is the curvature term described above. S_{QUICK} is always treated explicitly, however, u_{P} and u_{E} may be treated implicitly or explicitly depending on the time-stepping scheme. It is therefore important that any corrections for the boundary cells do not change the basic form of this formula.

At the boundaries, the truncation of the boundary cells means there is a large variation in cell size between the truncated boundary cell and adjacent cells. Consequently, the requirement that $0.8 < r_x < 1.25$ is not satisfied and the curvature term must be calculated more accurately. A formula which takes into account large changes in both node and cell face spacing is derived by integrating

$$\frac{\partial^2 u}{\partial x^2} = \mathrm{CRV} \tag{14}$$

for constant CRV between the node P and the cell face and then using the values of u at the relevant upwind nodes to find the constants of integration. The resulting formula is

$$u_e = (1-\theta)u_P + \theta u_E - \frac{1}{2}\mathrm{CRV} \times \Delta x_E^2 \theta(1-\theta), \tag{15}$$

where

$$\mathrm{CRV} = \frac{\Delta x_{EE} u_P - (\Delta x_E + \Delta x_{EE})u_E + \Delta x_E u_{EE}}{\Delta x_E \Delta x_{EE}(\Delta x_E + \Delta x_{EE})/2}, \quad u_e < 0, \tag{16}$$

or

$$\mathrm{CRV} = \frac{\Delta x_E u_W - (\Delta x_W + \Delta x_E)u_P + \Delta x_W u_E}{\Delta x_W \Delta x_E(\Delta x_W + \Delta x_E)/2}, \quad u_e > 0. \tag{17}$$

Using this formula, the advective flux takes the same form as Eq. (13), the only change being in the calculation of S_{QUICK}.

The second modification required concerns the position of the interpolated velocity on the cell face. As can be seen in Fig. 2, the velocity at the face of a boundary cell estimated by the interpolations above, u_e, is not necessarily located at the centre of the face. To correct this in a way that fits easily into the existing scheme, the fact that the velocity at the boundary is zero (the no-slip boundary condition) is used. Interpolation between point e and the boundary along the cell face gives the velocity at the centre of the face as

$$u_{ec} = \alpha_c u_e, \tag{18}$$

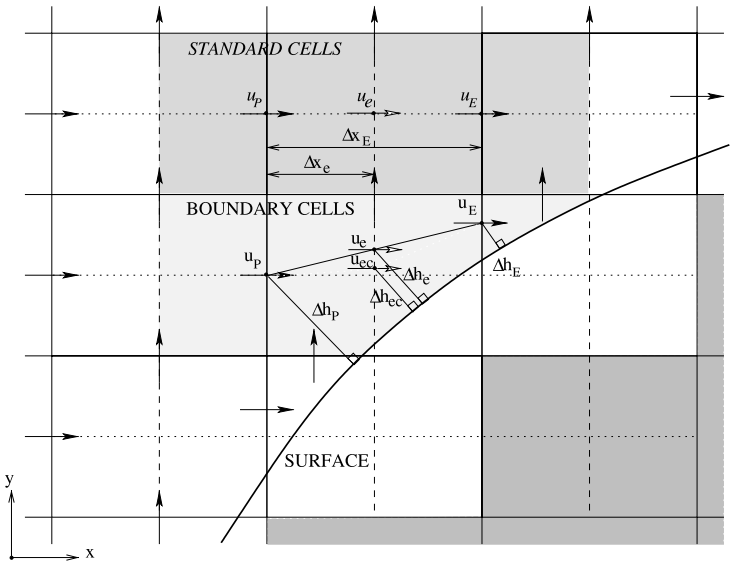


Fig. 2. Advection of the u component of velocity through the east face for a standard cell and a boundary cell. The perpendicular distances, Δh, of various points from the surface are used to calculate the interpolation correction used in the flux calculations.

where the correction factor α_c is calculated geometrically, assuming that the local curvature of the surface is small compared to the grid cell size,

$$\alpha_c \approx \frac{\Delta h_{ec}}{\Delta h_e} \approx \frac{\Delta h_{ec}}{(1-\theta)\Delta h_P + \theta \Delta h_E}. \tag{19}$$

Perpendicular distances, Δh, of various key points from the boundary are used to calculate the correction factor. Δh_e is approximated using Δh_P and Δh_E as these are already calculated during the course of creating the new truncated cells. A method for calculating these distances is presented in Section 2.4.

Using this approach, the formula for the advective flux becomes

$$F_{\mathrm{adv}_e} = (\varrho \alpha_c u \Delta A)_e \alpha_c u_e = C_e[((1-\theta)u_P + \theta u_E) - S_{\mathrm{QUICK}}], \tag{20}$$

where $C_e = \alpha_c^2 (\varrho u A)_e$. This is identical to formula (13) except that C_e has been multiplied by the correction factor squared, α_c^2.

Advection of a velocity component in a direction perpendicular to the direction of the component is shown in Fig. 3. In this case advecting and advected velocity components are no longer the same, so the correction factor used for the advected velocity in the previous case is applied only once. The formula for the advective flux of u across the north face by component v becomes

$$F_{\mathrm{adv}_n} = (\varrho v \Delta A)_n \alpha_c u_n = C_n[((1-\theta)u_P + \theta u_N) - S_{\mathrm{QUICK}}], \tag{21}$$

where $C_n = \alpha_c (\varrho v A)_n$.

The advecting velocity v must also be considered. On a staggered grid, even for standard cells, the advecting velocity must be interpolated to the centre of the cell face using a formula such as

$$v_e = \theta_e v_{ne} + \theta_w v_{nw}, \tag{22}$$

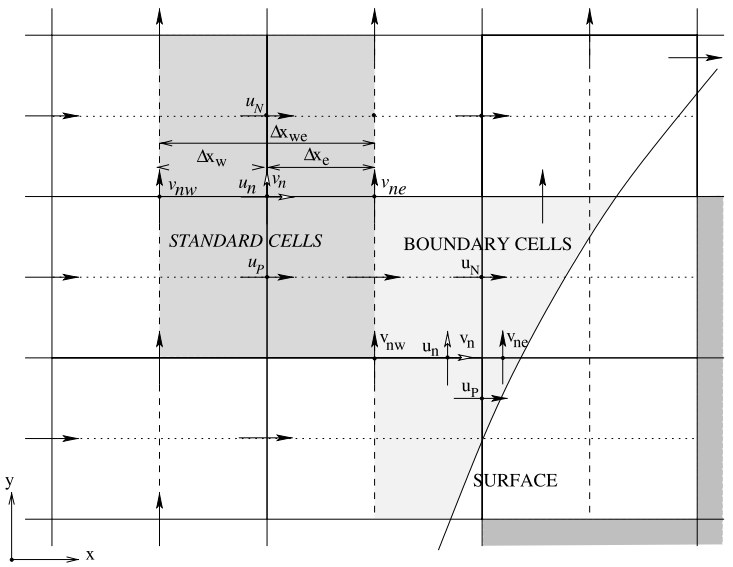


Fig. 3. Advection of the u component of velocity through the north face for a standard cell and a boundary cell. Weighting factors for the interpolation of the advecting velocity are calculated using the new position of the nodes.

where

$$\theta_e = \frac{\Delta x_w}{\Delta x_{we}} \quad \text{and} \quad \theta_w = \frac{\Delta x_e}{\Delta x_{we}}. \tag{23}$$

The only modification required for the boundary cells is to recalculate θ_e and θ_w based on the new position of the nodes as shown in Fig. 3. In three dimensions this interpolated velocity may still be offset slightly from the centre of the face.

In summary, advection between boundary cells is handled by multiplying the advective flux contribution by a correction factor and using weighting factors in the interpolations which have been revised based on the new position of the nodes.

2.3.2. Diffusive fluxes

The diffusive flux of u across a cell face is given by

$$F_{\text{diff}} = \left[\mu \Delta A \left(\frac{\partial u}{\partial \boldsymbol{n}}\right)\right]_f, \tag{24}$$

where μ is the viscosity, $\boldsymbol{n}$ is the direction normal to the face, and A is the area of the face. As was the case for advection, all variables are evaluated at the centre of the cell face.

Diffusion of a velocity component, u, for standard and boundary cells is shown in Fig. 4. For standard cells, the central difference approximation to the gradient gives the flux across the east cell face as

$$F_{\text{diff}_e} = \frac{\mu \Delta A}{\Delta x_E}(u_E - u_P) = D_e(u_E - u_P), \tag{25}$$

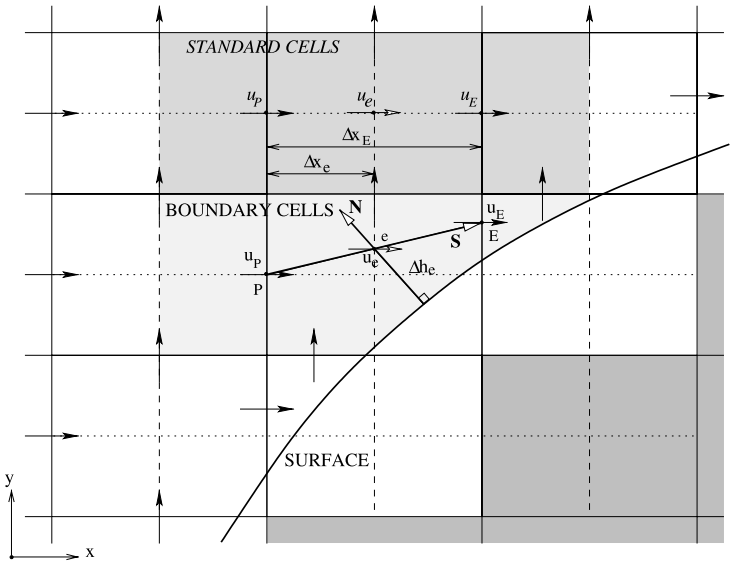


Fig. 4. Diffusion of the u component of velocity through the east face for a standard cell and a boundary cell. The perpendicular distance of point e from the surface is shown as Δh_e. $\boldsymbol{N}$ is the unit normal vector from the surface through the point e and $\boldsymbol{S}$ is the vector from P to E.

where $D_e = \mu A/\Delta x_E$ is the diffusive flux contribution to the coefficients in the discretised conservation equation, Eq. (7). Again, since u_P and u_E may be treated implicitly or explicitly it is important that any corrections for the boundary cells do not change the basic structure of this formula.

As can be seen in Fig. 4, diffusion between boundary cells will need some modification. The line between the nodes P and E is no longer perpendicular to the cell face, so the simple form given in Eq. (25) is no longer valid. A correction that will fit into the existing scheme is derived as follows. A displacement vector, $\boldsymbol{S}$ from P to E is defined. Normalising $\boldsymbol{S}$ gives a unit vector defining the direction between the two nodes

$$\boldsymbol{s} = \frac{\boldsymbol{S}}{|\boldsymbol{S}|}. \tag{26}$$

The directional derivative along $\boldsymbol{s}$ is defined as

$$\frac{\partial u}{\partial \boldsymbol{s}} = \boldsymbol{s}\nabla u = s_x\frac{\partial u}{\partial x} + s_y\frac{\partial u}{\partial y} + s_z\frac{\partial u}{\partial z}, \tag{27}$$

where s_x, s_y, and s_z are the components of $\boldsymbol{s}$ in the x, y, and z directions, respectively. Approximating $\partial u/\partial \boldsymbol{s}$ using central differences gives

$$\frac{u_E - u_P}{\boldsymbol{S}} \approx s_x\frac{\partial u}{\partial x} + s_y\frac{\partial u}{\partial y} + s_z\frac{\partial u}{\partial z}, \tag{28}$$

allowing the gradient perpendicular to the east face to be written as

$$\frac{\partial u}{\partial x} \approx \frac{1}{s_x}\left[\frac{u_E - u_P}{|\boldsymbol{S}|} - \left(s_y\frac{\partial u}{\partial y} + s_z\frac{\partial u}{\partial z}\right)\right] \approx \frac{u_E - u_P}{|S_x|} - \frac{1}{s_x}\left(s_y\frac{\partial u}{\partial y} + s_z\frac{\partial u}{\partial z}\right). \tag{29}$$

Approximations are obtained for $\partial u/\partial y$ and $\partial u/\partial z$ by using the no-slip boundary condition on velocity. In Fig. 4 the point e is the intersection of $\boldsymbol{S}$ and the east cell face, Δh is the perpendicular distance of e from the surface and $\boldsymbol{N}$ is the unit normal vector at the surface which passes through e. A method for calculating the perpendicular distance Δh and the vector $\boldsymbol{N}$ is outlined in Section 2.4. The gradients at the point e are approximated using a one-sided difference between u_e and the boundary value of zero. From the constructions shown in Fig. 4

$$\frac{\partial u}{\partial y} \approx \frac{N_y(u_e - u_b)}{\Delta h} \quad \text{and} \quad \frac{\partial u}{\partial z} \approx \frac{N_z(u_e - u_b)}{\Delta h}. \tag{30}$$

Using the same linear interpolation as was used for the advective fluxes to estimate u_e, an approximation is obtained for the gradient perpendicular to the north face

$$\frac{\partial u}{\partial x} \approx \frac{u_E - u_P}{S_x} - \left[\frac{(1-\theta)u_P + \theta u_E}{s_x\Delta h}(s_yN_y + s_zN_z)\right] \approx \frac{u_E - u_P}{S_x} - \left[\frac{(1-\theta)u_P + \theta u_E}{S_x\Delta h}(S_yN_y + S_zN_z)\right]. \tag{31}$$

The first term is equivalent to the formula for standard cells given in Eq. (25) with the new displacement of the nodes taken into account, while the second term constitutes a correction term to account for the non-orthogonality of the cell geometry. The simplest way to implement the correction term is to add it to the explicit source term as

$$S_{exp} = S_{exp} - \mu A\frac{(1-\theta)u_P + \theta u_E}{S_x\Delta h}(S_yN_y + S_zN_z). \tag{32}$$

2.3.3. Wall shear stress

A major shortcoming of using a staircase boundary relates to the calculation of the force on the fluid due to the wall shear stress. Typically the perpendicular distance from the node to the staircase rather than to the original surface is used. In the method presented here, the perpendicular distance between the node and the quadric surface is used in conjunction with the area of the surface within the cell to calculate the shear force on the fluid.

Fig. 5 shows a shear stress, $\boldsymbol{\tau}$, acting on the fluid with velocity, $\boldsymbol{u} = (u, v, w)$. Assuming that $\boldsymbol{u}$ is tangential to the surface, for a no-slip boundary the shear stress and velocity vectors are parallel, so that

$$\boldsymbol{\tau} = -\mu \frac{\partial \boldsymbol{u}}{\partial h}. \tag{33}$$

Breaking the velocity vector into its components $\boldsymbol{u} = u\boldsymbol{i} + v\boldsymbol{j} + w\boldsymbol{k}$, the shear stress can be approximated as

$$\boldsymbol{\tau} \approx -\mu\left(\frac{u}{\Delta h}\boldsymbol{i} + \frac{v}{\Delta h}\boldsymbol{j} + \frac{w}{\Delta h}\boldsymbol{k}\right) \tag{34}$$

and hence the shear force due to the wall is

$$\boldsymbol{F} \approx -\mu A_{\text{surf}}\left(\frac{u}{\Delta h}\boldsymbol{i} + \frac{v}{\Delta h}\boldsymbol{j} + \frac{w}{\Delta h}\boldsymbol{k}\right), \tag{35}$$

where A_{surf} is the area of surface within the cell. A method for calculating A_{surf} is outlined in Section 2.4. From Eq. (35) it is seen that the shear force can also be broken into components which depend only on the associated velocity component

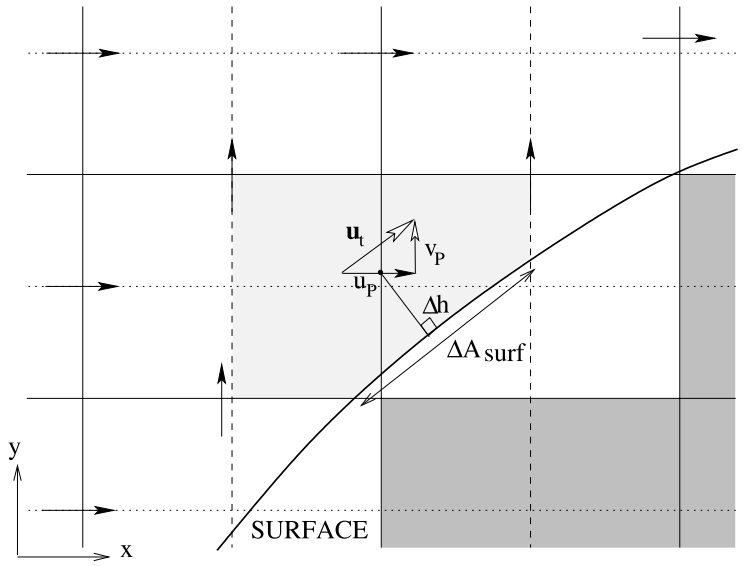


Fig. 5. Wall shear stress for u component of velocity. The perpendicular distance of the velocity node from the surface is shown as Δh. The associated total velocity vector $\boldsymbol{u}_{\text{t}}$ is assumed to be parallel to the shear stress $\boldsymbol{\tau}_w$. The area of surface within the cell is shown as ΔA_{surf}.

$$F_x \approx -\mu \Delta A_{\text{surf}} \frac{u}{\Delta h},$$
$$F_y \approx -\mu \Delta A_{\text{surf}} \frac{v}{\Delta h}, \tag{36}$$
$$F_z \approx -\mu \Delta A_{\text{surf}} \frac{w}{\Delta h}.$$

The shear force can therefore be applied implicitly as is generally done with staircase boundaries. The difference in the treatment here is that the area of the quadric surface actually in contact with the cell and the true distance between the node and the surface are used.

If a wall function is used, the total velocity at a node, $\boldsymbol{u} = (u_1, u_2, u_3)$, is usually required in order to non-dimensionalise the height and velocity in terms of wall units. Here, $u_i/\Delta h$ can be estimated by projecting the local strain rate tensor, S_{ij}, onto the surface by contracting it with the unit normal vector at the surface, N_j,

$$\frac{u_i}{\Delta h} \approx S_{ij} \cdot N_j. \tag{37}$$

2.3.4. Problem cases—cell linking

A common problem with cut cell methods is the creation of very small cells. This has a number of important consequences. Firstly, since the Courant number is inversely proportional to the cell dimension in the direction of a particular velocity component, the presence of small velocity cells implies a proportionally small time step in order to maintain numerical stability. Secondly, a large variation in cell size causes the resulting system of linear equations to become stiff and difficult to solve. This is particularly problematic in the case of the pressure correction equation since this equation is elliptic. A third problem specifically related to small velocity cells on a staggered grid is that, under some circumstances, it is possible to have a velocity cell that does not have two pressure cells associated with it. An example of this is shown in Fig. 6. Where this occurs mass conservation cannot be enforced for that velocity cell and the pressure gradient term in the momentum equation cannot be calculated.

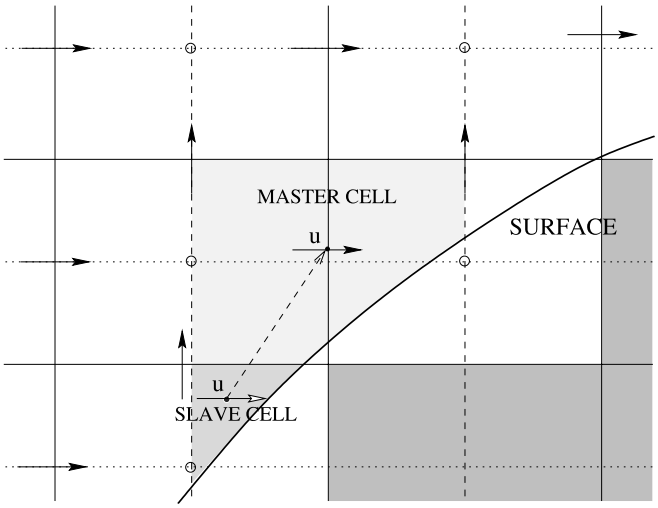


Fig. 6. A master and slave cell are shown for the u component of velocity. The slave cell velocity has only one pressure node associated with it. It is moved to the same position as the master cell node.

These three problems may be overcome by merging small "problem" cells with a larger neighbour. As discussed in the introduction, however, cell-merging generally brings its own problems such as the need to calculate fluxes with additional diagonally adjacent neighbours. There are also significant problems associated with the formulation of a systematic merging algorithm in three-dimensions. To address these issues we have developed a novel cell-merging or more accurately, cell-linking approach. Rather than actually merging two cells to form a single cell, the two cells are linked as a "master/slave" pair in which the two nodes are made coincident while each cell remains a distinct entity. Because the slave node and slave cell continue to exist as entities separate from the the master node and master cell, the fluxes and wall shear stresses, as well as volumetric and surface information, are calculated in exactly the same way for the master and slave cells as they are for the standard boundary cells. In particular, the same interpolation corrections described in Sections 2.3.1 and 2.3.2 can be used for the fluxes between slave cells and other cells. In the case of slave cells, these corrections are particularly important as the node is moved a considerable distance from its original position.

The criteria used to determine when cell-linking is performed are designed to resolve the three problems associated with small cells listed above. The first criterion is that any velocity cell which has only one associated pressure cell becomes a slave cell and is linked with a master cell. The coincidence of the master and slave nodes then means that the pressure gradient and velocity correction calculated for the master node can also be used for the slave node. In fact this criterion also resolves the Courant number problem since, as can be seen in Fig. 6, a velocity cell will only have two associated pressure cells while the maximum cell width in the direction of the velocity component is greater than half the full cell width. Linking cells based on the single-pressure-cell criterion limits the minimum cell width in the direction of a velocity component to half the standard cell width and so resolves the problem associated with the Courant number criterion.

This does not overcome the problem of stiffness in the pressure correction equation, however, as it is still possible to create pressure cells with very small volumes. This is overcome by requiring that the area of the larger face in each pair of cell faces (E/W, N/S, and U/D) be no less than 1% of the original cell face area. Pressure cells which do not meet this criterion are merged by removing the node from the calculation and treating the associated velocity cells as slave cells. In this way, the stiffness problem is removed, while both momentum and mass are still conserved throughout the entire computational domain.

Linking the master and slave velocity nodes is achieved in the following manner. The two nodes are not made exactly coincident but rather, the slave node is placed a small distance ($10^{-8}\times$ cell size) from the master node. The diffusion flux between the two nodes then automatically becomes extremely high and forces the two velocities to take the same value. There are other ways of linking the two velocity nodes, but again this approach ensures that the slave and master cells can be treated in exactly the same manner as standard boundary and non-boundary cells.

The master cell for a given slave cell is chosen to be the cell whose node lies closest to the boundary normal. In cases where two nodes are equally close, such as when a boundary is at 45° to a regular grid, the choice is arbitrary, and cells are given priority in order of their direction x, y, z relative the slave cell.

The linking procedure outlined above ensures that all fluxes and forces in the linked cells are accounted for while resolving problems associated with the creation of small cells. The above procedure is equally applicable to two- and three-dimensional formulations.

2.4. Geometry

2.4.1. Boundaries as quadric surfaces

A general quadric surface in a Cartesian coordinate system is defined as

$$a_{11}x_1^2 + a_{22}x_2^2 + a_{33}x_3^2 + 2a_{12}x_1x_2 + 2a_{13}x_1x_3 + 2a_{23}x_2x_3 + 2a_1x_1 + 2a_2x_2 + 2a_3x_3 + a = 0 \tag{38}$$

or in matrix form as

$$\boldsymbol{x}\boldsymbol{A}\boldsymbol{x}^{\mathrm{T}} + 2\boldsymbol{a}\boldsymbol{x}^{\mathrm{T}} + a = \boldsymbol{0}, \tag{39}$$

where

$$\boldsymbol{x} = (x_1, \quad x_2, \quad x_3), \quad \boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \quad \boldsymbol{a} = (a_1, \quad a_2, \quad a_3). \tag{40}$$

Here, $\boldsymbol{A}$ is a symmetric matrix, $a_{ij} = a_{ji}$. Through appropriate coordinate transformations, any quadric can be rewritten in one of two normal forms,

$$\lambda_1 x_1^2 + \lambda_2 x_2^2 + \lambda_3 x_3^2 + d = 0 \tag{41}$$

or

$$\lambda_1 x_1^2 + \lambda_2 x_2^2 + m x_3 = 0 \tag{42}$$

depending on whether or not the surface has a point of symmetry. We will concentrate on the first case. The signs of the characteristic values (or eigenvalues) determine the nature of the surface. Quadric surfaces described by Eq. (41) are ellipsoids, single sheet hyperboloids or double sheet hyperboloids. Cylinders and planes constitute degenerate forms of these surfaces resulting from one or two of the characteristic values being zero. A list of quadric surfaces with their characteristic values can be found in [40].

For the purposes of defining boundary conditions using quadric surfaces it is useful to start with the surface in normal form so the basic shape of the surface can be defined. This shape can then be rotated and translated to the desired position. In matrix notation the normal form is

$$\boldsymbol{x}_{\Sigma_0}\boldsymbol{C}\boldsymbol{x}_{\Sigma_0}^{\mathrm{T}} + d = 0, \tag{43}$$

where $\boldsymbol{C}$ is the matrix of characteristic values

$$\boldsymbol{C} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{44}$$

and Σ_0 is the coordinate system with axes formed by the principal axes of the surface. This coordinate system is identical to the coordinate system of the computational domain.

Rotating the surface is equivalent to rotating the coordinate axes and requiring that the original equation be satisfied in the new coordinate system. Thus rotation of the surface by an angle θ_1 around the x_1-axis is achieved by transforming to a new coordinate system Σ_{1a} through multiplication by a rotation matrix of direction cosines $\boldsymbol{R}_1$,

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{\Sigma_{1a}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & \sin\theta_1 \\ 0 & -\sin\theta_1 & \cos\theta_1 \end{bmatrix}}_{\boldsymbol{R}_1} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{\Sigma_0} \tag{45}$$

and then requiring that Eq. (43) be satisfied in Σ_{1a}, that is

$$\boldsymbol{x}_{\Sigma_{1a}}\boldsymbol{C}\boldsymbol{x}_{\Sigma_{1a}}^{\mathrm{T}} + d = 0. \tag{46}$$

Returning to coordinates Σ_0,

$$\left(\boldsymbol{R}_1\boldsymbol{x}_{\Sigma_0}^{\mathrm{T}}\right)^{\mathrm{T}}\boldsymbol{C}\left(\boldsymbol{R}_1\boldsymbol{x}_{\Sigma_0}^{\mathrm{T}}\right)+d=0, \tag{47}$$

which can be simplified using the identity $(AB)^{\mathrm{T}}=B^{\mathrm{T}}A^{\mathrm{T}}$ to give

$$\boldsymbol{x}_{\Sigma_0}\boldsymbol{R}_1^{\mathrm{T}}\boldsymbol{C}\boldsymbol{R}_1\boldsymbol{x}_{\Sigma_0}^{\mathrm{T}}+d=0. \tag{48}$$

A sequence of rotations about the x_1, x_2, and x_3 axes can be written as a single transformation $\boldsymbol{R}=\boldsymbol{R}_3\cdot\boldsymbol{R}_2\cdot\boldsymbol{R}_1$ from coordinate system Σ_0 to Σ_1. (The order of rotations is important.) The rotated surface is then given by the equation

$$\boldsymbol{x}_{\Sigma_0}\boldsymbol{R}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{R}\boldsymbol{x}_{\Sigma_0}^{\mathrm{T}}+d=0. \tag{49}$$

The second step is to move the surface to the desired position in the domain. Translation of the coordinate system Σ_1 to Σ_2 by a vector $\boldsymbol{T}$ is given by

$$\boldsymbol{x}_{\Sigma_2}=\boldsymbol{x}_{\Sigma_1}-\boldsymbol{T}_{\Sigma_1} \tag{50}$$

$$=\left(\boldsymbol{x}_{\Sigma_0}-\boldsymbol{T}_{\Sigma_0}\right)\boldsymbol{R}^{\mathrm{T}}. \tag{51}$$

Substitution into Eq. (49) gives the final form of the equation for the quadric surface rotated and translated in the coordinate system Σ_0

$$\left(\boldsymbol{x}_{\Sigma_0}-\boldsymbol{T}_{\Sigma_0}\right)\boldsymbol{A}\left(\boldsymbol{x}_{\Sigma_0}-\boldsymbol{T}_{\Sigma_0}\right)^{\mathrm{T}}+d=0, \tag{52}$$

where $\boldsymbol{A}=\boldsymbol{R}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{R}$.

In summary, the data required to specify a general quadric surface in space comprise: the characteristic values, λ_i; the constant, d; the angle of rotation about each axis and order of rotation; and the translation vector.

2.4.2. Calculation of geometric information

The calculations of correction factors in Section 2.2 require the unit normal vector at the surface that passes through a given point $\boldsymbol{x}_{\mathrm{P}}$ within the domain, and also the perpendicular distance of that point from the surface.

The normal vector $\boldsymbol{N}$ at a point $\boldsymbol{x}_0$ on a surface $f(\boldsymbol{x})=0$ is given by

$$\boldsymbol{N}(\boldsymbol{x}_0)=\left(\frac{\partial f(\boldsymbol{x})}{\partial x_1}\right)_{\boldsymbol{x}_0}+\left(\frac{\partial f(\boldsymbol{x})}{\partial x_2}\right)_{\boldsymbol{x}_0}+\left(\frac{\partial f(\boldsymbol{x})}{\partial x_3}\right)_{\boldsymbol{x}_0}. \tag{53}$$

For a quadric surface defined in the form outlined in Section 2.4.1, $f(\boldsymbol{x})$ is given by

$$f(\boldsymbol{x})=(\boldsymbol{x}-\boldsymbol{T})\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{T})^{\mathrm{T}}+d. \tag{54}$$

Here it is simpler to use $\boldsymbol{y}$ where $\boldsymbol{y}=\boldsymbol{x}-\boldsymbol{T}$ giving

$$f(\boldsymbol{y})=\boldsymbol{y}\boldsymbol{A}\boldsymbol{y}^{\mathrm{T}}+d=0, \tag{55}$$

since

$$\left(\frac{\partial f(\boldsymbol{x})}{\partial x_i}\right)_{\boldsymbol{x}_0}=\left(\frac{\partial f(\boldsymbol{y})}{\partial y_i}\right)_{y_0=(\boldsymbol{x}_0-\boldsymbol{T})}. \tag{56}$$

Partial differentiation of Eq. (55) gives

$$\frac{\partial f(\boldsymbol{y})}{\partial y_i} = \boldsymbol{y}\boldsymbol{A}\left(\frac{\partial \boldsymbol{y}}{\partial y_i}\right)^{\mathrm{T}} + \left(\frac{\partial \boldsymbol{y}}{\partial y_i}\right)\boldsymbol{A}\boldsymbol{y}^{\mathrm{T}}, \tag{57}$$

which is more clearly represented in tensor notation

$$\frac{\partial f}{\partial y_i} = y_j a_{jk}\frac{\partial y_k}{\partial y_i} + \frac{\partial y_j}{\partial y_i}a_{jk}y_k \tag{58}$$

$$= y_j a_{ji} + a_{ik}y_k \tag{59}$$

$$= y_j(a_{ji} + a_{ij}). \tag{60}$$

Thus the normal vector at the surface is given by

$$N_i(\boldsymbol{x}_0) = (y_j(a_{ji} + a_{ij}))_{\boldsymbol{y}_0} \tag{61}$$

or

$$\boldsymbol{N}(\boldsymbol{x}_0) = (\boldsymbol{y}(\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}}))_{\boldsymbol{y}_0} \tag{62}$$

$$= ((\boldsymbol{x} - \boldsymbol{T})(\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}}))_{\boldsymbol{x}_0}. \tag{63}$$

Finding a line normal to the surface that passes through a given point $\boldsymbol{x}_{\mathrm{P}}$ off the surface requires first finding the intersection of the line with the surface, $\boldsymbol{x}_0$. This is accomplished as follows. The equation for this line can be written parametrically as

$$\boldsymbol{x} = \boldsymbol{x}_{\mathrm{P}} + t\boldsymbol{N}(\boldsymbol{x}_0), \tag{64}$$

where t is a parameter and $\boldsymbol{N}(\boldsymbol{x}_0)$ is the normal vector at the surface. Because the surface is defined implicitly (i.e., in the form $f(\boldsymbol{x}) = 0$), partial differentiation of f at a point in the neighbourhood of $\boldsymbol{x}_0$ will give an estimate of the normal vector at $\boldsymbol{x}_0$. Thus a first approximation to the normal can be found by differentiating $f(\boldsymbol{x})$ at $\boldsymbol{x}_{\mathrm{P}}$,

$$\boldsymbol{N}(\boldsymbol{x}_0) \approx \left(\frac{\partial f(\boldsymbol{x})}{\partial x_1}\right)_{\boldsymbol{x}_{\mathrm{P}}} + \left(\frac{\partial f(\boldsymbol{x})}{\partial x_2}\right)_{\boldsymbol{x}_{\mathrm{P}}} + \left(\frac{\partial f(\boldsymbol{x})}{\partial x_3}\right)_{\boldsymbol{x}_{\mathrm{P}}}. \tag{65}$$

This gives a first estimate of the equation of the line

$$\boldsymbol{x} = \boldsymbol{x}_{\mathrm{P}} + t\boldsymbol{N}^1(\boldsymbol{x}_0). \tag{66}$$

The point of intersection of the line with the surface is found by solving the associated system of equations. Because the equation of the surface is non-linear it is necessary to use an iterative method such as the Newton–Kantorovich method (see, for example, [40]) for this task.

The intersection point provides a first estimate of the point on the surface, $\boldsymbol{x}_0^1$. This can be used to calculate a more accurate estimate of $\boldsymbol{N}$ and the process is repeated. This method has proved to be very efficient and convergence to a high degree of accuracy ($10^{-8}\times$ the cell size) is generally achieved in a few iterations. Once the surface point $\boldsymbol{x}_0$ is known, it is a simple matter to calculate the unit normal vector at this point and the distance between $\boldsymbol{x}_{\mathrm{P}}$ and $\boldsymbol{x}_0$ is the perpendicular distance required.

Face areas and free volumes of truncated cells are calculated by numerical integration. The area of the quadric surface contained within a cell is obtained by firstly projecting differential areas $\mathrm{d}\boldsymbol{A}$ from a cell face onto the surface to give $\mathrm{d}\boldsymbol{A}_{\mathrm{surf}}$ and then integrating. Fig. 7 shows how this is performed if a cell face in the xy-plane is chosen. Using this method, the area of surface within a cell is given by

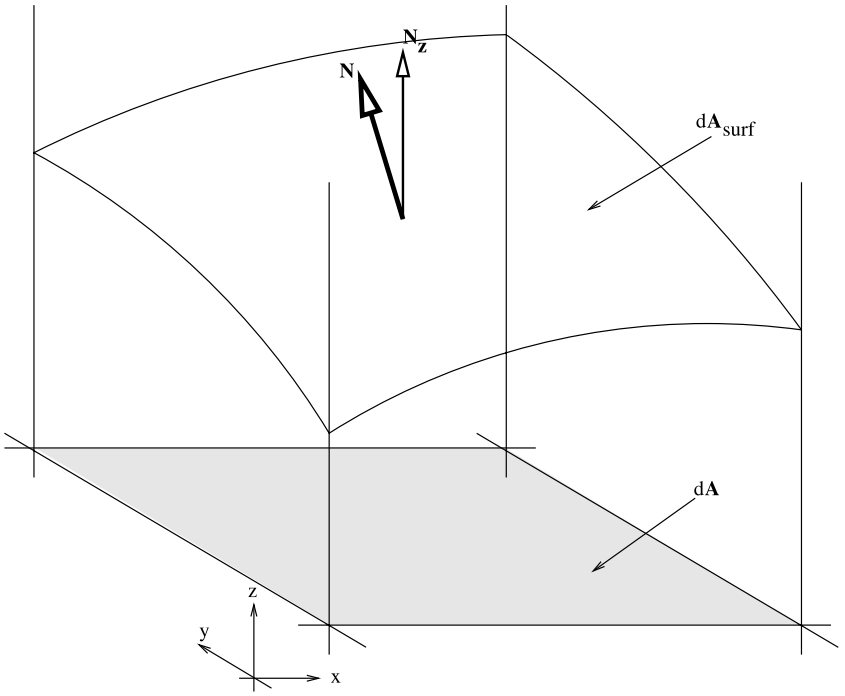


Fig. 7. Projection of a differential area $\mathrm{d}\boldsymbol{A}$ in the xy-plane onto the boundary surface to give $\mathrm{d}\boldsymbol{A}_{\mathrm{surf}}$. $\boldsymbol{N}$ is the direction of $\mathrm{d}\boldsymbol{A}_{\mathrm{surf}}$ and $\boldsymbol{N}_z$ is the projection of $\boldsymbol{N}$ onto the z-axis.

$$A_{\mathrm{s}} = \int\int_{\Omega} \frac{\mathrm{d}A}{|\boldsymbol{N}_z|}. \tag{67}$$

Here, $\boldsymbol{N}$ is the direction of $\mathrm{d}\boldsymbol{A}_{\mathrm{surf}}$, $\boldsymbol{N}_z$ is the projection of $\boldsymbol{N}$ onto the z-axis, and Ω is the projection onto the cell face of the surface contained within the cell. To maximise the accuracy of the procedure the cell face chosen is that for which Ω is greatest.

2.5. Implementation

The discrete equations obtained using the method described above are solved using a Bi-conjugate Gradient solver with an MSI preconditioner. Convergence of the solver is measured using the L_2 norm of the residual. The convergence criterion for the solution of the momentum equations is that the residual be less than 10^{-12}. Typically a single sweep of the preconditioner and solver are required to obtain convergence.

For unsteady calculations, which are carried out using the second-order hybrid Adams scheme described in Section 2.1 with a Courant number of $C = 0.8$, the conservation error for momentum is typically of the order 10^{-3} for each time step. For steady-state calculations, using the method of false time stepping with the first-order implicit Euler scheme (see also Section 2.1), the solution is considered converged once the momentum conservation error is less than 10^{-6}. An exception to this is the spatial convergence test discussed in Section 3.1. In this case, the solution error must be significantly smaller than the discretisation error, and it is necessary to obtain the solutions to an accuracy of 10^{-8}.

To ensure adequate conservation of mass at each time step a number of iterations of the pressure/velocity correction step are generally required. Within each iteration the Poisson pressure correction equation is solved until either the residual is reduced to 10% of its original value or the solver has performed seven sweeps. The solution is then used to correct the pressure and velocity fields and the divergence of the corrected velocity field is calculated. The process is repeated until the divergence error is less than a pre-set value. For unsteady calculations, the divergence error criterion is $\mathrm{Div}_{\mathrm{err}} < 10^{-8}$. For steady-state calculations, to speed up the solution process, this criterion is relaxed to $\mathrm{Div}_{\mathrm{err}} < 10^{-4}$. As the solution converges, the divergence error decreases automatically and the final solution has a divergence error of less than 10^{-10}.

3. Results

3.1. Order of accuracy

The spatial order of accuracy of the scheme was tested for flow in a square driven cavity of height H containing a circular cylinder of diameter D where $D = 1/2H$. To avoid singularities in the pressure field at the corners, the lid speed was given a parabolic profile such that its value is zero at the corners and maximum at the centre of the cavity. The Reynolds number based on the mean lid velocity, U_{lid}, was $Re = 20$. The geometry and flow field are shown in Fig. 8(a).

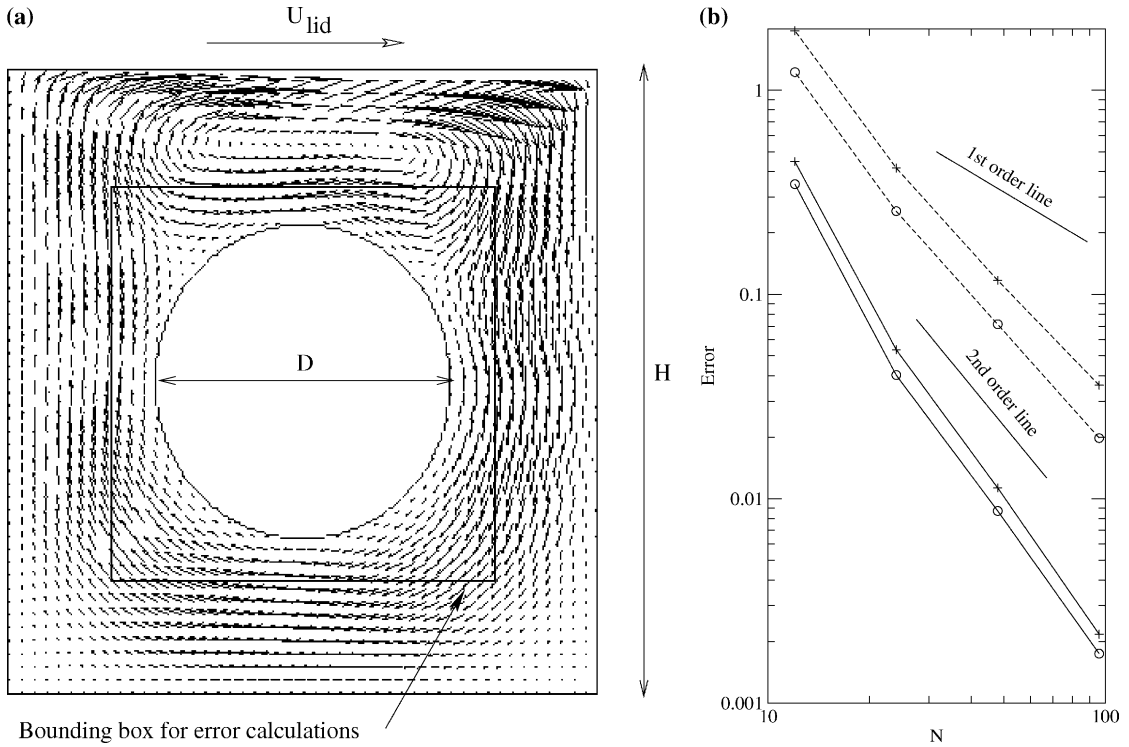


Fig. 8. Driven cavity with lid velocity U_{lid} and height H containing a cylinder of diameter D: (a) shows the resulting velocity field; (b) shows the variation of L_∞ and L_2 norm errors with the number of cells in each coordinate direction N for u (+) and v (○). Dashed and solid lines represent L_∞ and L_2 errors, respectively.

The flow was solved on five uniform grids listed in Table 1. A benchmark solution was calculated on the finest grid, Grid E, and errors in the solutions on the four coarser grids were then quantified as the L_2 and L_∞ norms of the difference of the test and the benchmark solution. Differences were obtained by interpolating the benchmark solution onto the test grid using fourth-order tension splines. To ensure that these norms represent errors in the boundary cell discretisation, errors were calculated only in a square box of size $2H/3$ surrounding the cylinder as shown in Fig. 8.

The errors for velocity components u and v plotted against the number of grid cells in each coordinate direction are shown in Fig. 8(b). Convergence in the asymptotic section of the L_2 norm curves for both u and v is second-order, showing that the scheme maintains global second-order accuracy in the region close to the boundary. In fact convergence is slightly better than second-order due to the third-order accuracy of the QUICK scheme used for the advection terms. The L_∞ norm curves also exhibit second-order convergence for both velocity components. In this case the slope of the curve for u is marginally lower than second-order. This may be due to irregularities in the mesh degrading the local error slightly at the boundary.

3.2. Channel flow

Results were obtained for laminar flow through a channel placed skewed to the grid at a Reynolds number $Re = 25$ based on the channel half-width H and the mean velocity U_m. The channel was placed at five angles relative to the grid: $\beta = 0°$, $\beta = 10°$, $\beta = 20°$, $\beta = 35°$, and $\beta = 45°$.

For the purposes of comparison, the calculations were performed using a staircase boundary and a cut cell boundary condition on two grids of square cells. The grids for the case $\beta = 20°$ are shown in Fig. 9. Grid A has cells of width 2H/7 corresponding to a maximum cell Reynolds number of around 10, and Grid B has cells of width 2H/15, corresponding to a maximum cell Reynolds number of around 5. A parabolic velocity profile is applied at the inlet boundary with the velocity direction aligned parallel to the axis of the channel. The outlet boundary condition for velocity is a zero normal gradient. The length of the channel is approximately ten times the channel width. Doubling the length of the channel has negligible effect on the results. The numerical results are compared with analytic results below.

Figs. 10 and 11 show comparisons of the numerical and analytic velocity profiles for the two components u and v on the two grids for the case $\beta = 20°$. The results are obtained on 20 $y =$ constant planes with 7 and 14 data points per plane for Grids A and B, respectively. On Grid A, the coarser of the two grids, the staircase method performs very poorly with large deviations from the analytic solution, particularly for the u component. By comparison, the cut cell boundary results for Grid A are excellent, with very little deviation observed. On the finer grid, Grid B, the staircase method performs considerably better than it did on Grid A, however some deviation from the analytic profile is still apparent. On Grid B the cut cell results overlay the analytic profile. It is clear that both the staircase and cut cell methods are converging to the analytic profile, however the cut cell boundary performs significantly better than the staircase boundary.

The errors in the velocity fields and the strain rate field for each angle are presented in Fig. 12 as the L_2 norm of the difference between the numerical and analytic solutions. For $\beta = 0°$ both schemes perform

Table 1
Grids used for spatial convergence test with circular cylinder in a driven cavity

Grid	Number of cells
A	12×12
B	24×24
C	48×48
D	96×96
E	384×384

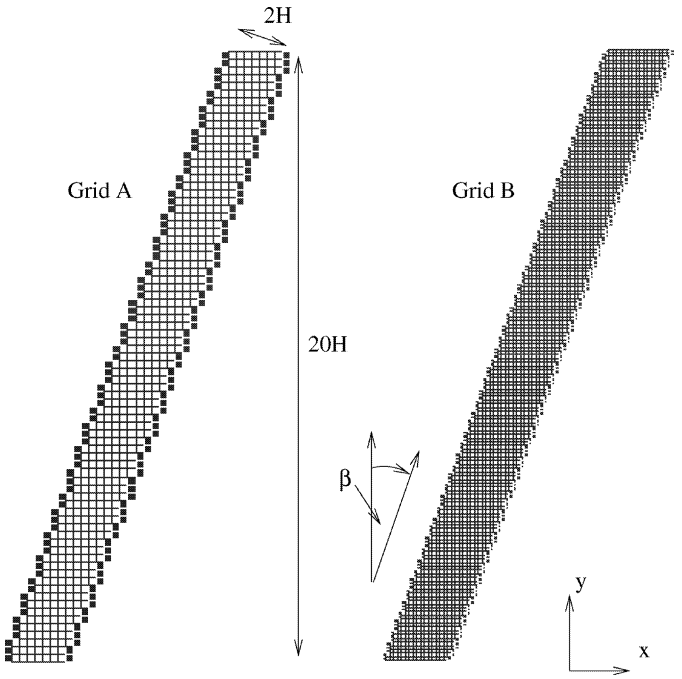


Fig. 9. Grids used for channel flow test. The channel has a half-width of H. The case shown is for a channel skewed at an angle $\beta = 20°$ to the grid.

equally well, as expected. For intermediate angles, $\beta = 10°$, $\beta = 20°$, and $\beta = 35°$, the errors with the cut cell boundary are typically one-quarter those of the staircase boundary. At $\beta = 45°$, a series of square steps provides a good approximation to the smooth boundary and the staircase boundary performs better than it did for the small angles. Even for this case, however, the cut cell boundary performs better, as a result of using the correct distance between the boundary-adjacent velocity node and the boundary.

3.3. Flow past a circular cylinder

The case of laminar flow past a circular cylinder provides more complex flow conditions as well as a curved surface for testing the cut cell method. The nature of the solution for cylinder flow changes as the Reynolds number increases, unlike the solution for laminar flow through a channel which is Reynolds number independent. At a Reynolds number of $Re \sim 6$ the flow separates from the cylinder and forms a pair of standing vortices in the wake of the cylinder. At Reynolds numbers greater than $Re \sim 40$–50 [41] the flow becomes unsteady. Results are presented for a Reynolds number, $Re = 40$, for which the flow has separated but has not yet become unsteady. All parameters are written in non-dimensional form using the following scalings:

$$l^* = l/D; \quad u^* = u/U_\infty; \quad v^* = v/U_\infty; \quad t^* = tU_\infty/D; \quad P^* = \frac{P}{\rho U_\infty^2}; \quad Re = \frac{U_\infty D}{\nu}, \tag{68}$$

where D is the cylinder diameter, ν is the kinematic viscosity, and U_∞ is the free-stream velocity.

While the physical domain for flow past a cylinder is infinite in all directions, it is necessary to use a finite computational domain. A schematic representation of the computational domain is shown in Fig. 13. Since

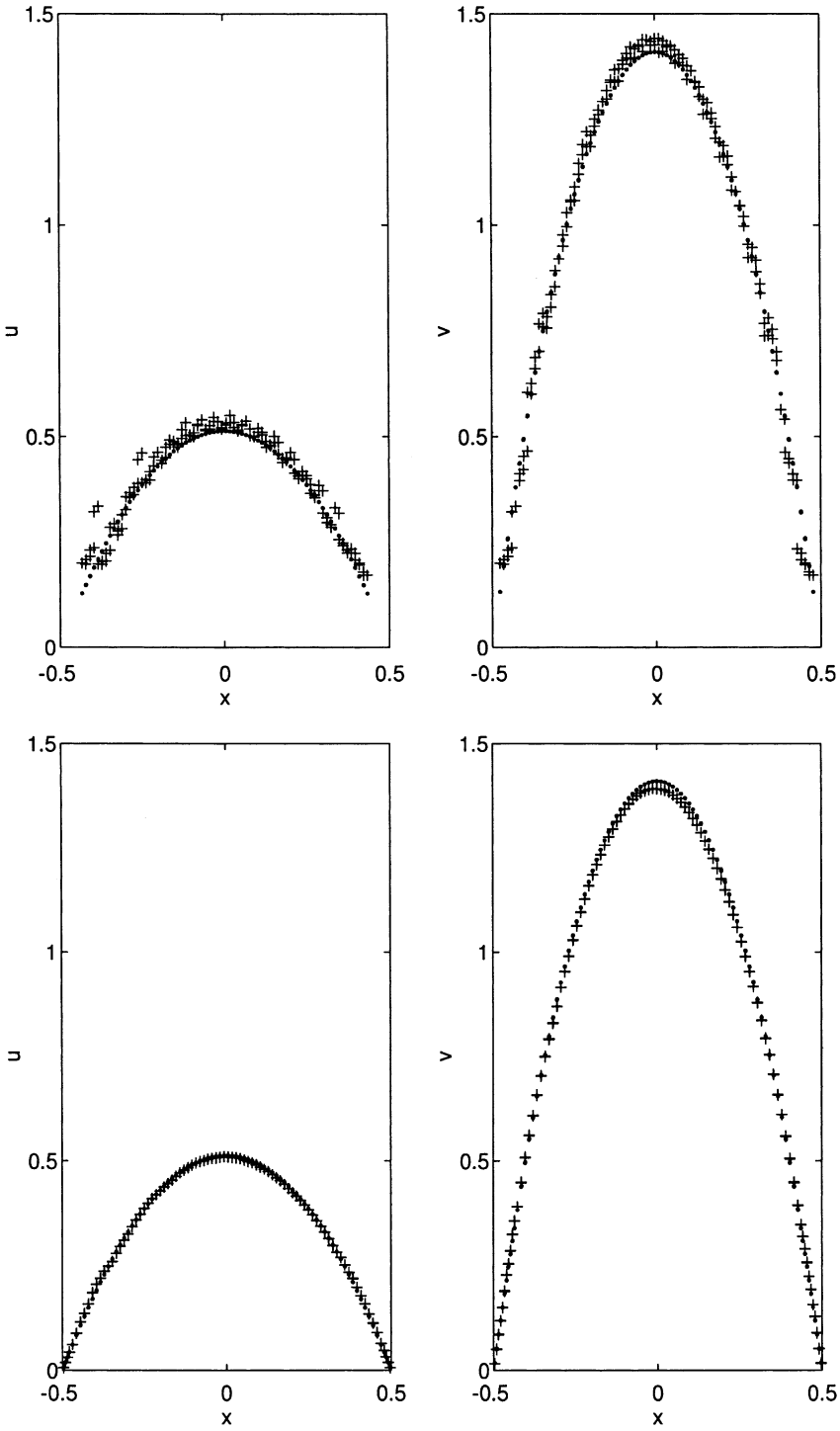


Fig. 10. Velocity profiles across channel skewed at angle $\beta = 20°$ to grid on Grid A: staircase boundary above and cut cell boundary below, with crosses representing the numerical results and filled circles the analytic parabolic profile.

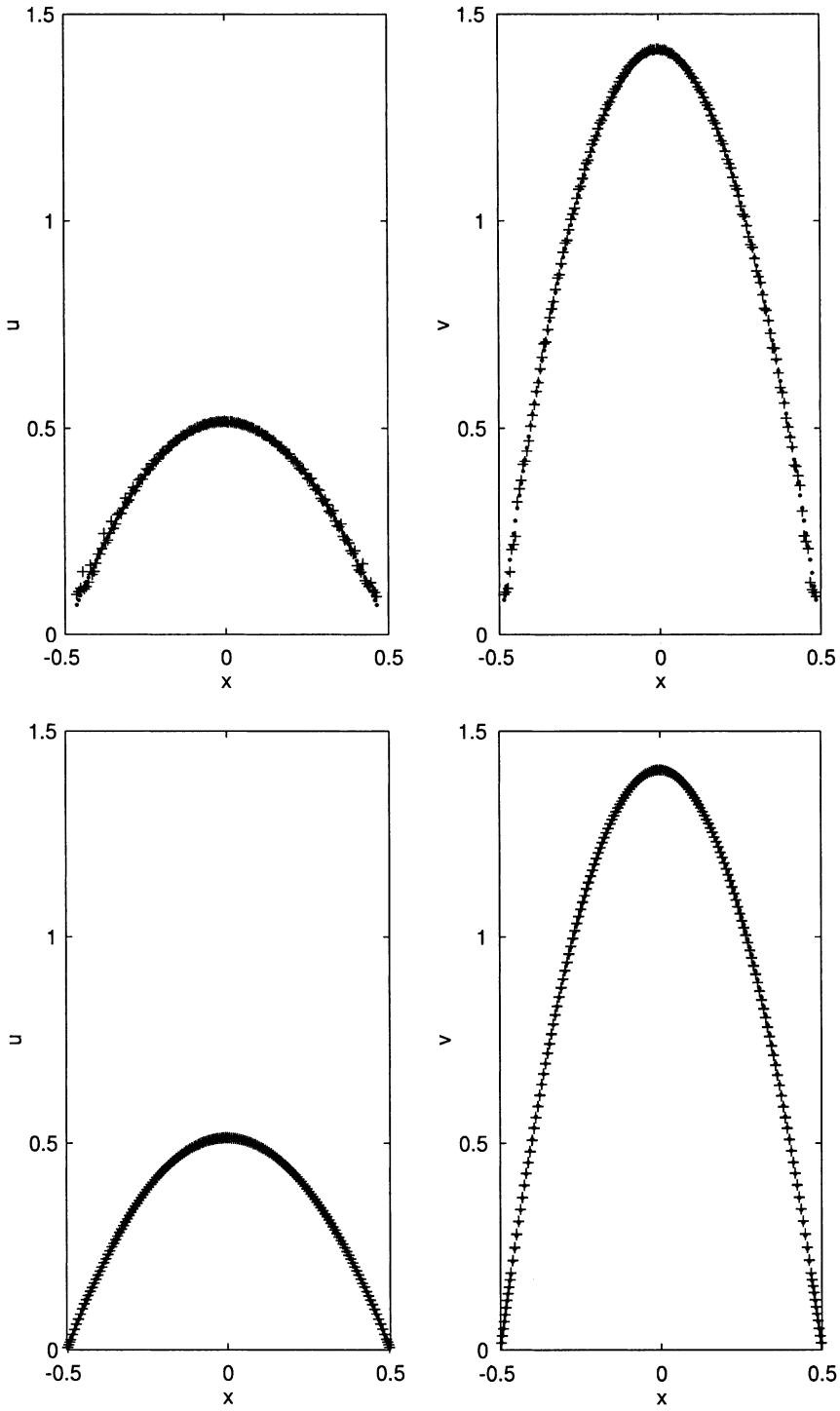


Fig. 11. Velocity profiles across channel skewed at angle $\beta = 20°$ to grid on Grid B: staircase boundary above and cut cell boundary below. Symbols as in Fig. 10.

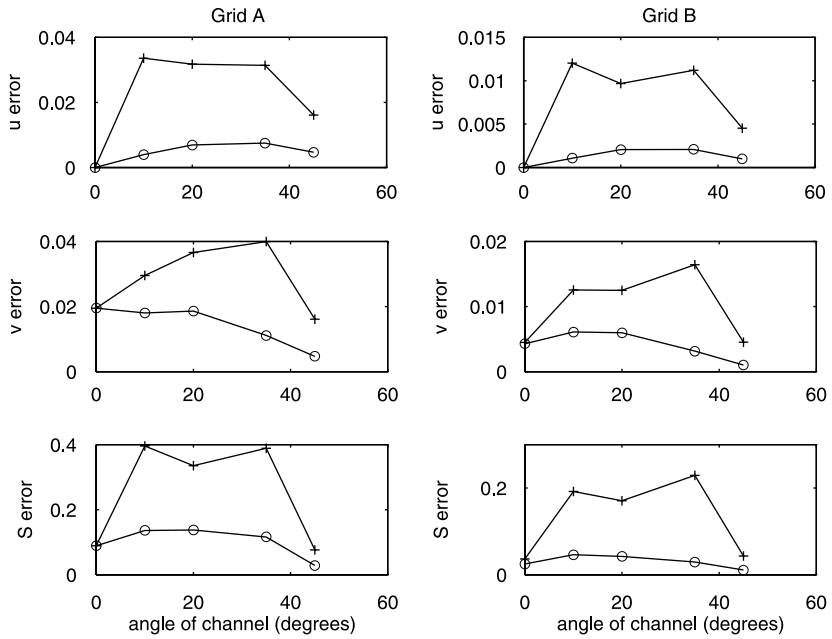


Fig. 12. Variation of velocity and strain rate (S) errors with angle of channel for cut cell boundary (○) and staircase boundary (+).

the flow is steady, a symmetry plane is placed through the centre of the cylinder. The domain size is 60×20 in the x and y directions, respectively, with the cylinder placed at $(20, 0)$ and the flow parallel to the x direction. The boundary conditions for velocity are: an inlet with uniform velocity profile upstream; a zero normal gradient boundary downstream; an impermeable, free-slip boundary at the sides; and an impermeable, no-slip boundary at the cylinder surface. For purposes of comparison, four grids are used for the

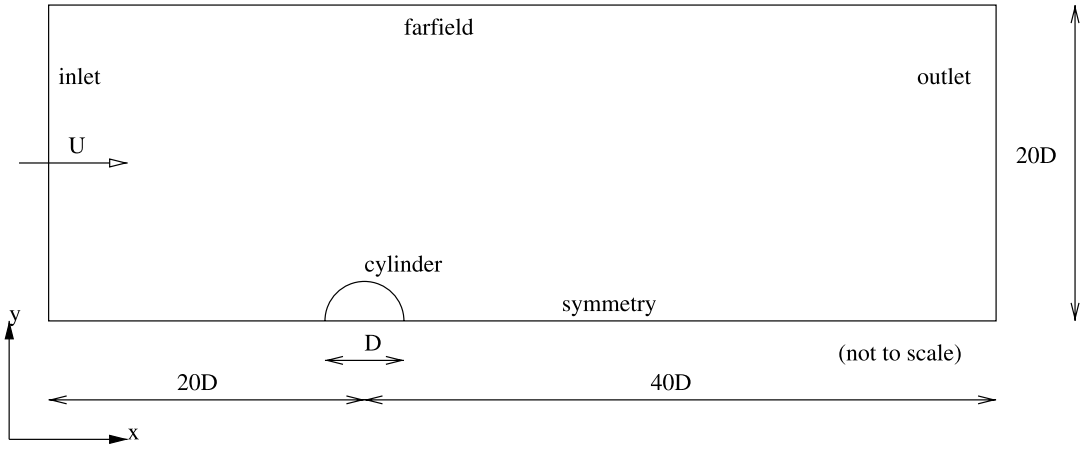


Fig. 13. Domain used for calculation of flow past a cylinder at $Re = 40$. D is the cylinder diameter and U is the velocity of the fluid at the inlet boundary.

simulations. These grids are listed in Table 2. In all cases the expansion ratio of the grid is less than 1.1 in both directions.

The parameters chosen for comparison are: vortex length L^*, drag coefficient C_D, and separation angle θ_s. The vortex length is given by $L^* = L/D$, where L is the length of the vortex measured from the rear of the cylinder. Vortex length is commonly used to validate numerical approaches, however, as pointed out by Koumoutsakos and Leonard [42], streamline diagnostics such as vortex length are a poor indicator of accuracy as streamlines are a smooth functional of the flow. The vorticity field is two derivatives less smooth than streamlines, so parameters which depend on the vorticity field such as drag coefficients and separation angle provide a more rigorous test.

The drag coefficient is defined as

$$C_D = \frac{\text{drag force}}{1/2\varrho U_\infty^2 D}. \tag{69}$$

The pressure component is calculated by integrating pressure over the surface of the cylinder

$$C_{D_p} = \int_0^{2\pi} P_s^* \cos\theta \, d\theta. \tag{70}$$

The frictional component is calculated by integrating the x component of the shear stress over the surface. For the cut cell boundary the shear stress at the surface is calculated in the application of the no-slip boundary condition (see Section 2.3.3). In the case of the staircase boundary, the shear stress at the staircase rather than the actual surface is used as the boundary condition. In this case the frictional force is found by integrating vorticity over the cylinder surface

$$C_{D_f} = \frac{2}{Re} \int_0^{2\pi} \omega_s^* \sin\theta \, d\theta. \tag{71}$$

The interpolations of the discrete pressure and vorticity fields to the cylinder surface required for the implementation of Eqs. (70) and (71) are performed using fourth-order tension splines. The separation angle θ_s is measured from the rear of the cylinder and is calculated by finding the point at which the vorticity at the surface changes sign. This method is exact for steady state flows [43].

Table 3 shows a comparison of the results obtained for the cut cell and staircase boundaries on the finest grid (Grid D) with the results of other authors. The experimental results of Coutanceau and Bouard [44] and the numerical results of Collins and Dennis [45] and Fornberg [46] have been widely used as benchmarks for testing numerical schemes. Both Fornberg, and Collins and Dennis used a stream-function/vorticity method in polar coordinates. The results of Zdravkovich [47] are a compilation of results collected from the literature. Both the cut cell and the staircase boundary solutions are in very good agreement with the benchmark results for all parameters. The greatest discrepancy is in the vortex length prediction. This parameter is very sensitive to the far field boundary conditions and it is probable that this accounts for the larger error.

Fig. 14 shows the variation of C_d and θ_s with grid resolution using the two techniques. Both methods are apparently converging to the same results, with a difference in the parameters of approximately 1–2% on the

Table 2
Grids used for calculation of flow past a cylinder at $R = 40$

Grid	Total number of cells	Number of cells along cylinder radius
A	126×58	10
B	202×78	20
C	271×108	40
D	297×144	80

Table 3
Comparison of results for staircase and cut cell boundaries obtained on Grid D together with other numerical and experimental results

	C_d	θ_s	L^*
Collins and Dennis [45]	1.56	53.6°	2.15
Coutanceau and Bouard [44]	—	53.4°	2.19
Fornberg [46]	1.4980	55°	2.24
Zdravkovich [47]	1.536	—	—
Cut cell	1.535	53.55°	2.259
Staircase	1.509	52.52°	2.260

finest grid. Furthermore, the cut cell method gives much better accuracy for a given grid resolution, particularly in the calculation of the drag force.

Figs. 15 and 16 show the pressure and vorticity distributions on the surface of the cylinder, calculated with Grids B and D, compared with the experimental results of Grove et al. [48] and numerical results of Fornberg [46]. The pressure distribution is presented in terms of the pressure coefficient,

$$C_P = \left(p - p_0 + \frac{1}{2}\rho V_\infty^2\right) \Big/ \frac{1}{2}\rho V_\infty^2.$$

Also shown are the velocity components u and v at a non-dimensional distance of $d^* = 0.025$ from the surface.

The distributions for the cut cell boundary on both grids are in very good agreement with the results of Grove et al. and Fornberg. The small grid scale oscillations in the results obtained with Grid B are no longer visible in the Grid D results. Distributions predicted using the staircase boundary on both grids are very poor in comparison. On Grid B there are large deviations from the benchmark results and considerable grid scale oscillations for all variables. The results for the staircase boundary on Grid D are better, however the oscillations in the vorticity profile are not removed with increased grid resolution.

These results reinforce the previous assertion that streamline diagnostics are poor indicators of the accuracy of a boundary condition. The staircase boundary gives poorest results for angles between 20° and 90° where the boundary layer is thinnest and the fluid velocity close to the surface is highest. In the wake region, however, the fluid velocity close to the surface is low and the poor representation of momentum transfer with the boundary has little influence. As a result, the large scale properties of the flow downstream of the cylinder, such as vortex length, are less dependent on the accuracy of the boundary representation than are local properties such as drag coefficients.

3.4. Flow past a hemispheric protuberance on a plate

In the final case the method is extended to three-dimensional, unsteady flow with the cut cell method used to define a solid hemispheric protuberance on a flat plate. This flow was investigated experimentally by Acarlar and Smith [49] (subsequently referred to as AS) who give a detailed description of the flow. A brief summary of their discussion is given here. Fig. 17 shows schematically the structure of the vorticity field generated by the interaction between a laminar boundary layer formed on the plate and the hemisphere. The vortex sheets which form in the boundary layer are deformed by the hemisphere. At the front of the hemisphere, the vortex sheets roll up to form a standing vortex. In the wake, the separation of the boundary layer and the spiralling motion generated by the interaction of the boundary layer with the hemisphere concentrate the vorticity in the region downstream of the protuberance. Fluctuations in the base pressure (the pressure behind the hemisphere) are associated with a periodic build-up and release

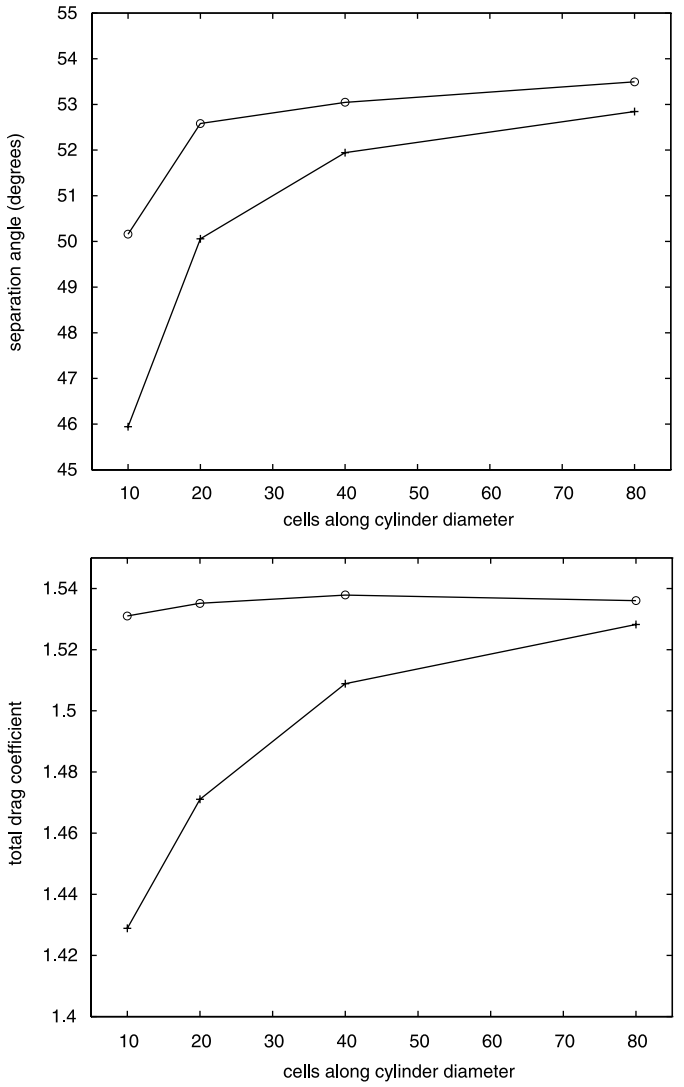


Fig. 14. Variation of separation angle and total drag coefficient with grid resolution for flow past a cylinder with cut cell boundary (○) and staircase boundary (+).

of concentrated vorticity and the generation of hairpin vortices in the wake. In Fig. 17 AS identify three regions in the hairpin vortex: the legs, head, and tip. The legs of the vortex run parallel to the flow direction while the head and tip form a vertical loop which is stretched radially as it is advected downstream.

The results obtained for this flow are presented below in non-dimensional form using the following scalings:

$$x^* = x/R; \quad y^* = y/R; \quad z^* = z/R; \quad \delta^* = \delta/R; \quad u^* = u/U; \quad v^* = v/U; \quad t^* = tU/R;$$
$$T_s^* = T_s U/R; \quad Re = \frac{UR}{\nu}. \tag{72}$$

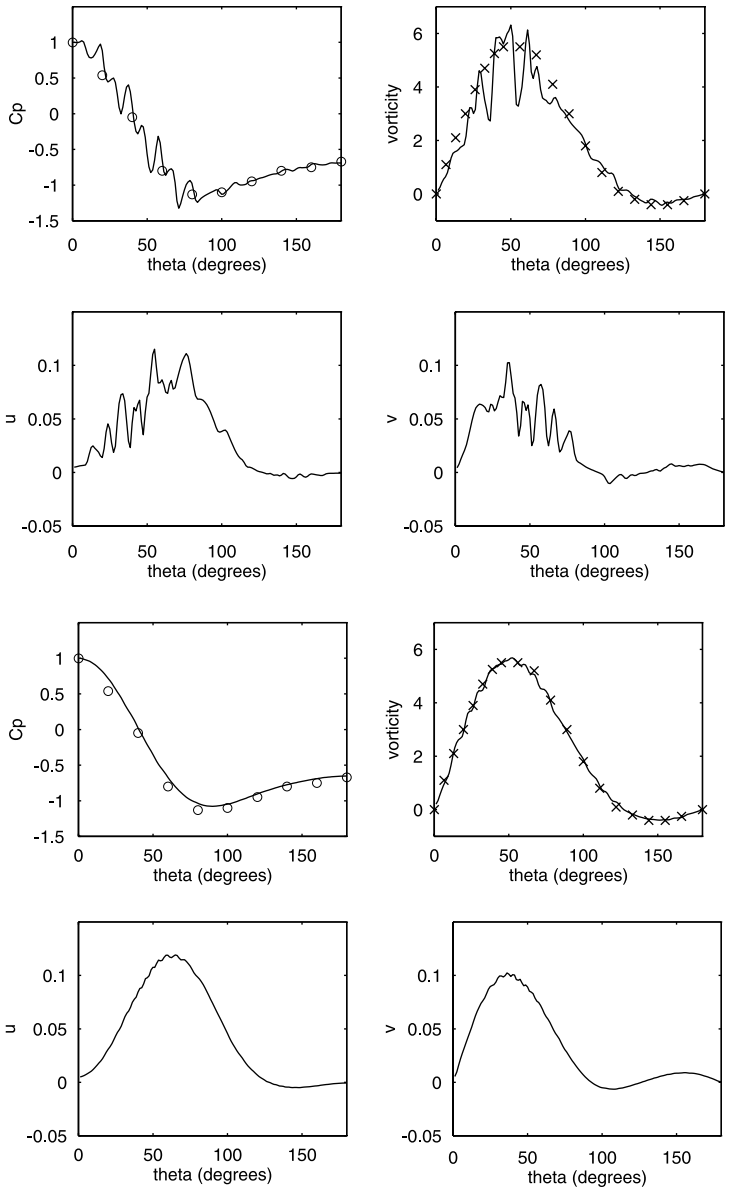
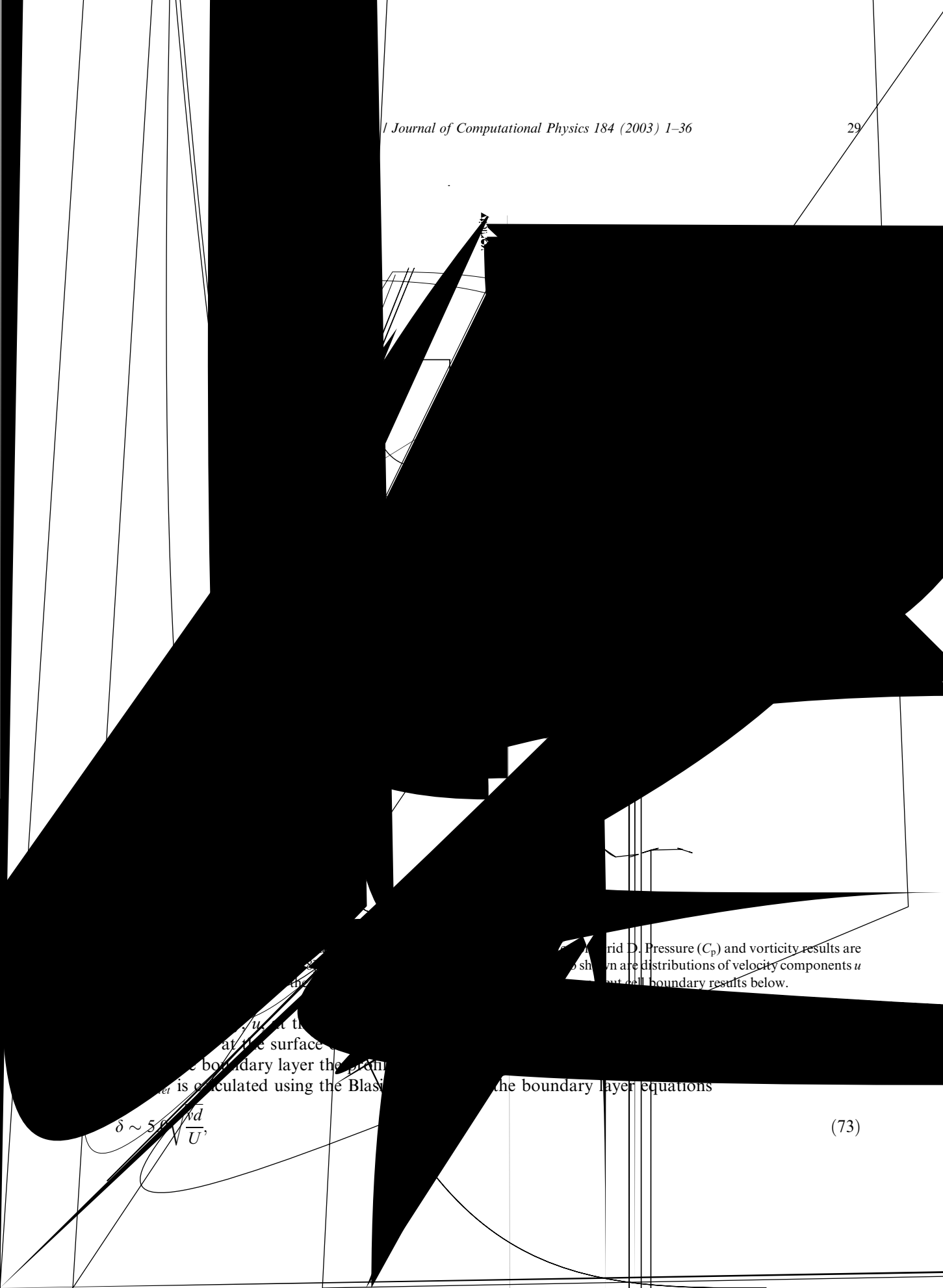


Fig. 15. Surface distributions of various parameters for flow past a cylinder computed on Grid B. Pressure (C_p) and vorticity results are compared to the results of Grove et al. [48] (○) and Fornberg [46] (×), respectively. Also shown are distributions of velocity components u and v at a distance $d^* = 0.025$ above the surface. Staircase boundary results are above and cut cell boundary results below.

Here R is the radius of the hemisphere, x, y, z are coordinates with origin at the centre of the hemisphere, δ is the thickness of the boundary layer, t is the time, T_s is the period of vortex shedding, v is the kinematic viscosity, and U is the free-stream velocity.

The flow was calculated for a Reynolds number $Re = 800$ with a boundary layer thickness at the hemisphere of $\delta_{\text{hemisphere}} = 1.2$. The computational domain used for the calculation is shown in Fig. 18. The



rid D. Pressure (C_p) and vorticity results are
n are distributions of velocity components u
ut cell boundary results below.

u t
at the surface
boundary layer the
is calculated using the Blas
he boundary layer equations

$$\delta \sim 5.0\sqrt{\frac{\nu d}{U}}, \tag{73}$$

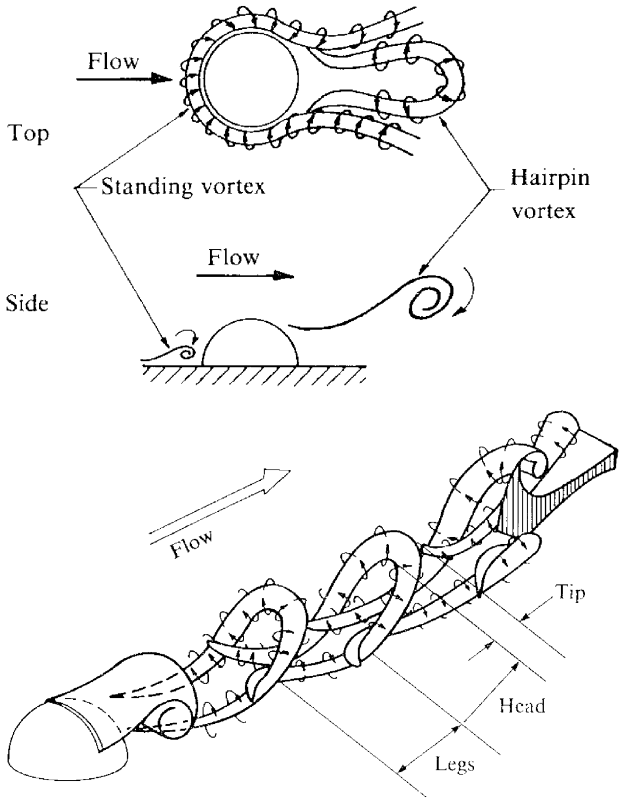


Fig. 17. Schematic drawing showing the structure of the vorticity field for flow over a hemispheric protuberance on a plate. (Copyright © Cambridge University Press, 1987. Reprinted with the permission of Cambridge University Press).

where d is the distance from the upstream edge of the plate. Equation (73) is first solved for $\delta_{\text{hemisphere}} = 1.2$ to find d. $d - 6R$ is then substituted back into the same equation to find δ_{inlet}. Symmetry boundaries are applied at the top and sides of the domain. Impermeable, no-slip boundaries are applied at the surfaces of the plate and the hemisphere, and a zero-normal-gradient boundary condition is applied to the velocity at the downstream boundary. The boundary conditions for the pressure correction are only required at the downstream boundary. Here the pressure correction is used to force the total mass flux across the boundary such that global mass conservation is satisfied. This boundary condition speeds convergence while retaining second-order time accuracy as discussed in [38].

The equations were integrated using the second-order Adams–Bashforth/Adams–Moulton scheme with the time step limited to give a maximum Courant number of 0.8. The equations were integrated to a time, $t^* = 72$. Inspection of time series results for the velocity and pressure fields indicated that the flow-field can be considered to be fully-developed at this time.

The results of AS show an approximately linear relationship between the Strouhal number ($St = fR/U$) and Reynolds number up to $Re \sim 1200$ with a range of 0–0.4. The vortex shedding period for the numerical simulation was calculated by measuring the period of oscillations in the pressure field at the location $(5.0, 0, 0.5)$. This method gives a vortex shedding period of $T_s^* \sim 4$ and a Strouhal number of approximately 0.25, which is in good agreement with the corresponding experimental value $St = 0.26$ for this flow.

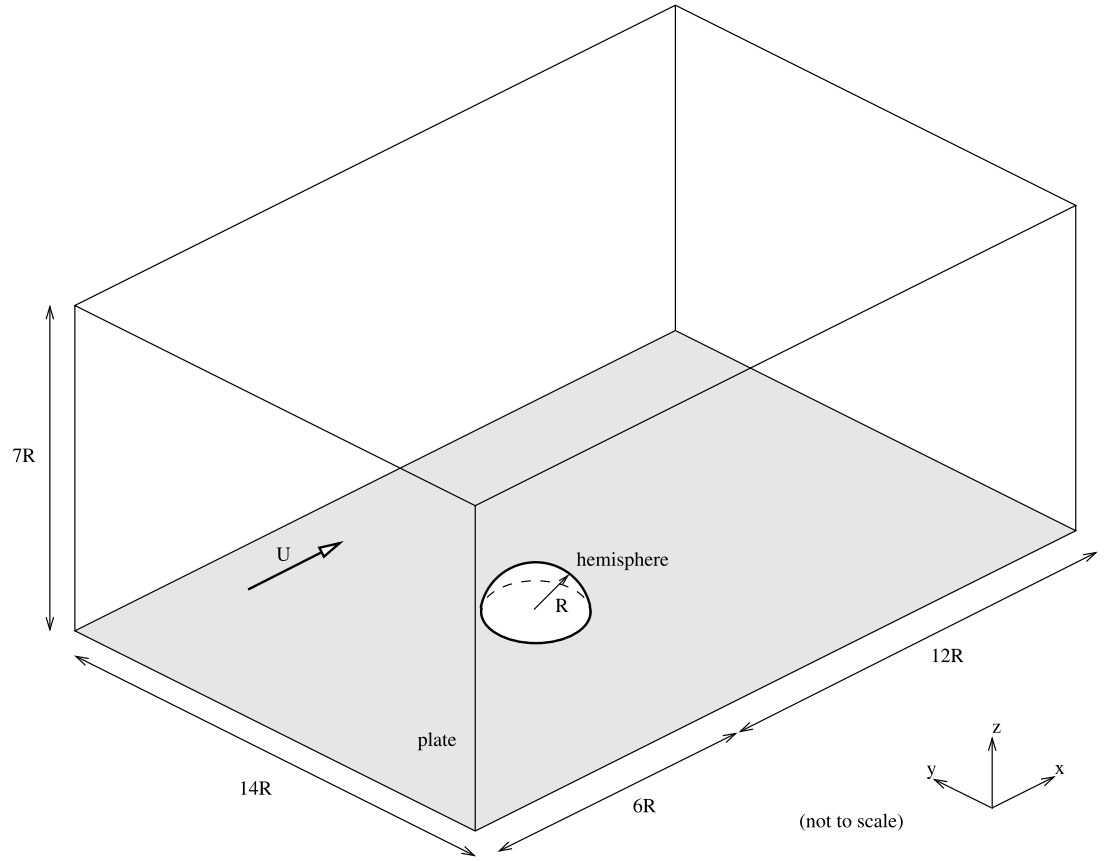


Fig. 18. Domain used for calculation of flow over a hemispheric protuberance on a plate. R is the radius of the cylinder and U is the free-stream velocity.

The fully developed flow-field results calculated over one vortex shedding period are shown from different perspectives in Figs. 19–21 with a time interval between the realisations of $\Delta t^* = 1.3 = T_s^*/3$. Fig. 19 shows a three-dimensional view of the vorticity field at $t^* = 72$, 73.3, and 74.6. Fig. 20 shows vorticity results on a streamwise cross-section through the centre of the hemisphere at $t^* = 72$, 73.3, 74.6, and 75.9 (giving a full cycle). Fig. 21 shows the vorticity and velocity fields on vertical cross-sections perpendicular to the flow direction at $t^* = 72$ for $x^* = 2.5$, $x^* = 5.0$, and $x^* = 7.5$. Also shown is an experimental visualisation produced by AS using hydrogen bubbles generated on a horizontal wire placed across the flow at $x^* = 5.0$.

There is good agreement with the results of AS. A standing vortex is seen around the base of the hemisphere and a vortex sheet formed by the separated boundary layer trails downstream from the separation line on the hemisphere to $x \sim 3.0$. This corresponds to the near-wake region defined by AS. The growth and progress of two hairpin vortices can also be seen. One hairpin vortex is emerging from the separated vortex sheet. A small residue of the sheet trails the vortex as it is advected by the main flow. A second hairpin vortex, further downstream, shows the continued development of the characteristic toroidal structure of the head and tip of the vortex. Comparison of the numerical and experimental flow fields at $x^* = 5.0$ (see Fig. 21) shows good agreement between the two sets of results.

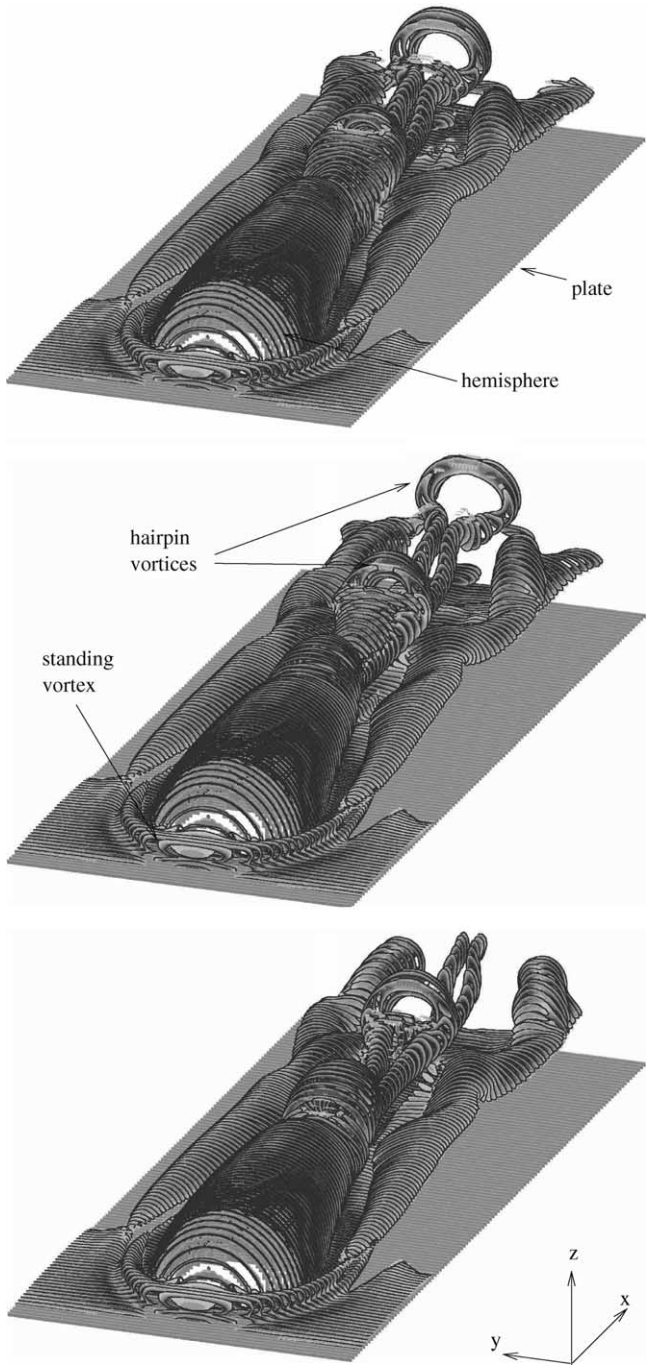


Fig. 19. Three-dimensional view of vorticity field for flow over a hemispheric protuberance on a plate. Solutions are $\Delta t^* = 1.3$ apart.

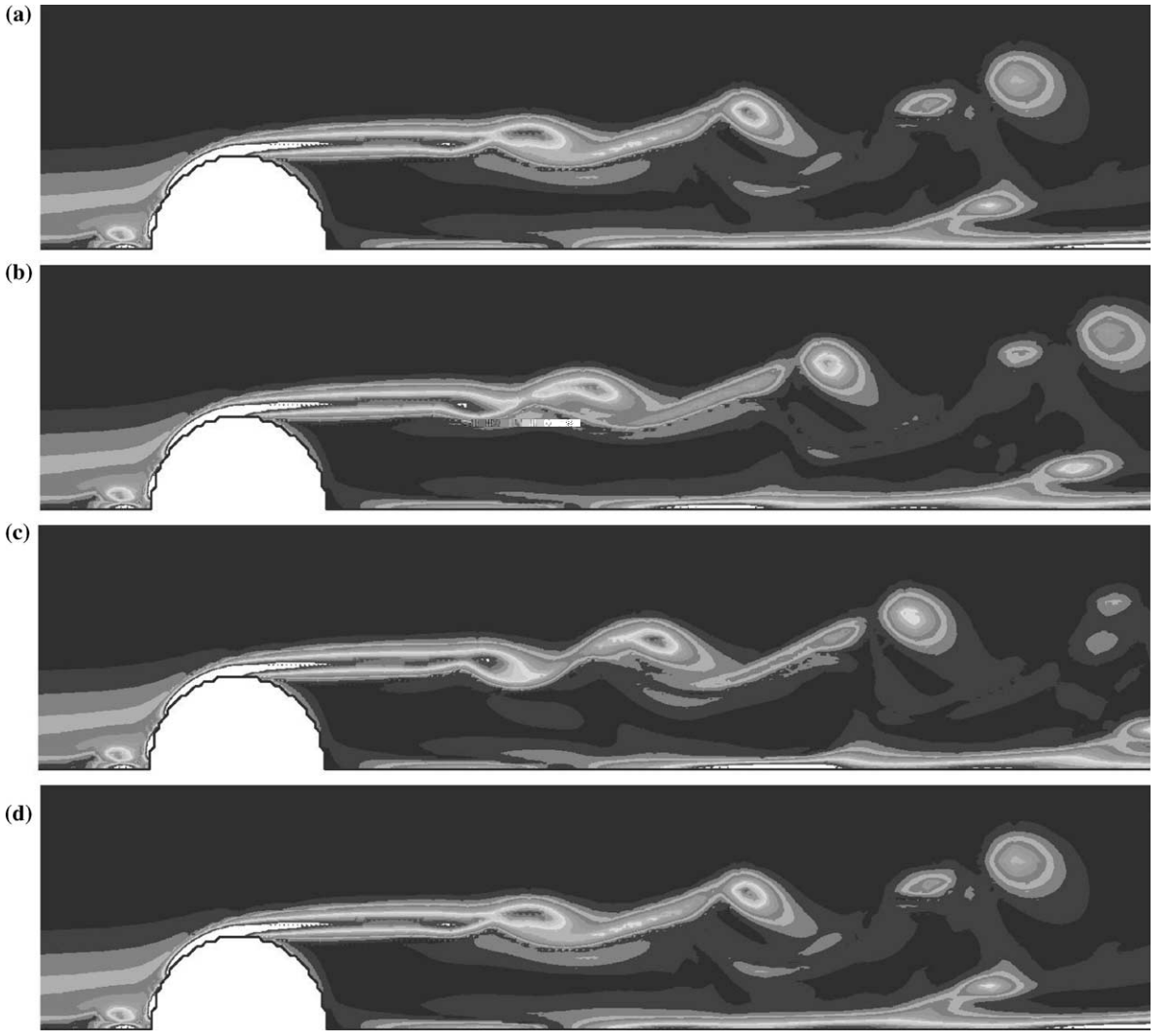


Fig. 20. Cross-section in xz-plane showing vorticity field for flow over a hemispheric protuberance on a plate over a complete cycle. Solutions are $\Delta t^* = 1.3$ apart.

4. Conclusions

A method has been presented for representing curved boundaries for the solution of the Navier–Stokes equations on a Cartesian grid. The approach involves truncating the Cartesian cells at the boundary surface to create new cells which conform to the shape of the surface. Methods for calculating the fluxes through the boundary cell faces, for representing pressure forces and for calculating the wall shear stress were derived and it was verified that the new scheme retains second-order accuracy in space.

Techniques were presented for generating the geometric information required for the scheme based on the representation of the boundaries as quadric surfaces. A series of quadric surfaces may be combined to create geometries common in engineering and scientific applications.

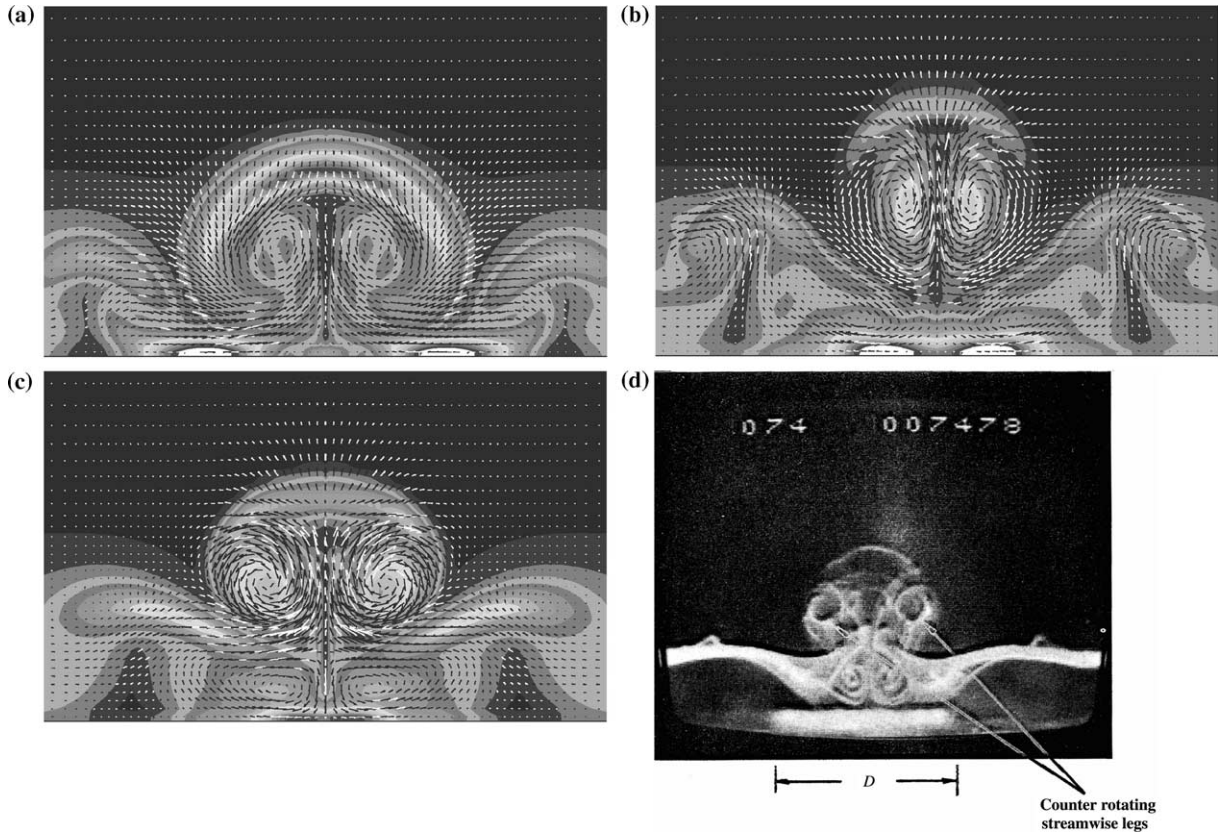


Fig. 21. Cross-section in *yz*-plane showing vorticity field for flow over a hemispheric protuberance on a plate at various distances from hemisphere: (a) $X^* = 2.5$; (b) $X^* = 5.0$; (c) $X^* = 7.5$; (d) experimental results of Acarlar and Smith [49]—visualisation of vorticity field at $X^* = 5.0$ using hydrogen bubbles. (Copyright © Cambridge University Press, 1987. Reprinted with the permission of Cambridge University Press).

The new method was compared to the staircase boundary commonly used to represent oblique or curved surfaces on a Cartesian grid. The two boundary formulations were compared to analytic results for flow through a channel placed oblique to the grid, and to accepted experimental and numerical results for flow past a cylinder at $Re = 40$. The new method typically gave a fourfold reduction for errors in the velocity field and an order of magnitude reduction for errors in the vorticity and strain-rate fields. The latter is of particular importance if turbulence models are used, as these models typically use the resolved strain-rate or vorticity fields to calculate turbulence parameters.

Finally, the new method was applied to an unsteady, three-dimensional problem—flow past a hemispheric protuberance on a plate at a Reynolds number of 800. The method performed well, with no stability problems, and good agreement was obtained with experimental results for this complex flow.

Acknowledgements

The first author gratefully acknowledges the support of the Australian Research Council.

References

[1] M. Hinatsu, J.H. Ferziger, Numerical computation of unsteady incompressible flow in complex geometry using a composite multigrid technique, Int. J. Numer. Meth. Fluids 13 (1991) 971–997.

[2] C.Y. Perng, R.L. Street, A coupled multigrid-domain-splitting technique for simulating incompressible flows in geometrically complex domains, Int. J. Numer. Meth. Fluids 13 (1991) 269–286.

[3] C.M. Albone, Embedded meshes of controllable quality synthesized from elementary geometric features, AIAA paper 92-0662, 1992.

[4] J.F. Thompson, F.C. Thames, C.W. Mastin, Automatic numerical grid generation of body-fitted curvilinear coordinate systems for fields containing any number of arbitrary two-dimensional bodies, J. Comp. Phys. 15 (1974) 299–319.

[5] D. Rayner, Multi-grid flow solutions in complex two-dimensional geometries, Int. J. Numer. Meth. Fluids 13 (1991) 507–518.

[6] G. Chesshire, W.D. Henshaw, Composite overlapping meshes for the solution of partial differential equations, J. Comp. Phys. 90 (1980) 1–64.

[7] N.P. Weatherill, C.R. Forsey, Grid generation and flow calculations for complex aircraft geometries using a multi-block scheme, AIAA paper 84-1665, 1984.

[8] W. Kordulla, M. Vinokur, Efficient computation of volume in flow predictions, AIAA J. (1983) 917–918.

[9] B.L. Muir, B.R. Baliga, Solution of three dimensional convection–diffusion problems using tetrahedral elements and flow orientated upwind interpolation functions, Numer. Heat Transfer 9 (1986) 143–162.

[10] J.H. Ferziger, M. Perić, Computational Methods for Fluid Dynamics, Springer, Berlin, 1996.

[11] J.D. Anderson Jr., Computational Fluid Dynamics—The Basics with Applications, McGraw-Hill, New York, 1995.

[er G. inFluid Newcthep23etries,Int. Fluid Dyn5I6ApplicaW.6flvfl'6s0xIy4i'I8t9o[4fieIlrz''xIIF0T4xyI6s,0z1h6iSMT6'619I4MT}7c'y'61racy1

[30] E.Y. Tau, A 2nd-order projection method for the incompressible Navier–Stokes equations in arbitrary domains, J. Comp. Phys. 115 (1) (1994) 147–152.
[31] P.G. Tucker, Z. Pan, A Cartesian cut cell method for incompressible viscous flow, Appl. Math. Modell. 24 (2000) 591–606.
[32] B.P. Leonard, A stable and accurate convective modeling procedure based on quadratic upstream interpolation, Comput. Meth. Appl. Mech. Eng. (1979).
[33] M.J. Aftosmis, M.J. Berger, J.E. Melton, Robust and efficient Cartesian mesh generation for component-based geometry, AIAA J. 36 (6) (1998) 952–960.
[34] M.P. Kirkpatrick, A large eddy simulation code for industrial and environmental flows, PhD thesis, The University of Sydney, 2002.
[35] J. Van Kan, A second order accurate pressure correction scheme for viscous incompressible flow, SIAM J. Sci. Stat. Comput. 7 (1986) 870–891.
[36] S.W. Armfield, R. Street, The fractional-step method for the Navier–Stokes equations on staggered grids: the accuracy of three variations, J. Comp. Phys. 153 (1999) 660–665.
[37] P. Gresho, On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via finite element method that also introduces a nearly consistent mass matrix. Part 1: Theory, Int. J. Numer. Meth. Fluids (1990) 587–620.
[38] M.P. Kirkpatrick, S.W. Armfield, J.H. Kent, T. Dixon, Simulation of vortex-shedding flows using high-order fractional step methods, J. ANZIAM 42 (E) (2000) 856–876.
[39] I.P. Castro, J.M. Jones, Studies in numerical computations of recirculating flows, Int. J. Numer. Meth. Fluids (1987).
[40] I.N. Bronshtein, K.A. Semendyayev, Handbook of Mathematics, Springer, Berlin, 1985.
[41] C. Norberg, An experimental investigation of the flow around a circular cylinder: influence of aspect ratio, J. Fluid Mech. 258 (1994) 287–316.
[42] P. Koumoutsakos, A. Leonard, High-resolution simulations of the flow around an impulsively started cylinder using vortex methods, J. Fluid Mech. 296 (1995) 1–38.
[43] W. Sears, D.P. Telionis, Boundary layer separation in unsteady flow, SIAM J. Appl. Math. 28 (1975) 215–235.
[44] M. Coutanceau, R. Bouard, Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow, J. Fluid Mech. 79 (1977) 231–256.
[45] W.M. Collins, S.C.R. Dennis, Flow past an impulsively started circular cylinder, J. Fluid Mech. 60 (1973) 105–127.
[46] B. Fornberg, A numerical study of steady viscous flow past a circular cylinder, J. Fluid Mech. 98 (1980) 819–855.
[47] M.M. Zdravkovich, Flow Around Circular Cylinders, Oxford University Press, Oxford, 1997.
[48] A.S. Grove, F.H. Shair, E.E. Petersen, A. Acrivos, An experimental investigation of the steady separated flow past a circular cylinder, J. Fluid Mech. 19 (1964) 60–80.
[49] M.S. Acarlar, C.R. Smith, A study of hairpin vortices in a laminar boundary layer. Part 1. Hairpin vortices generated by a hemisphere protuberance, J. Fluid Mech. 175 (1987) 1–41.