INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2000; **34**: 463–478

Unstructured pressure-correction solver based on a consistent discretization of the Poisson equation

Benjamin de Foy* and William Dawes

Department of Engineering, University of Cambridge, Trumpington Street, Cambridge, U.K.

SUMMARY

A finite volume incompressible flow solver is presented for three-dimensional unsteady flows based on an unstructured tetrahedral mesh, with collocation of the flow variables at the cell vertices. The solver is based on the pressure-correction method, with an explicit prediction step of the momentum equations followed by a Poisson equation for the correction step to enforce continuity. A consistent discretization of the Poisson equation was found to be essential in obtaining a solution. The correction step was solved with the biconjugate gradient stabilized (Bi-CGSTAB) algorithm coupled with incomplete lower–upper (ILU) preconditioning. Artificial dissipation is used to prevent the formation of instabilities. Flow solutions are presented for a stalling airfoil, vortex shedding past a bridge deck and flow in model alveoli.

KEY WORDS: flow solver; incompressible; tetrahedron; unsteady; unstructured

1. INTRODUCTION

Unstructured meshes have become very popular in computational fluid dynamics (CFD) and many compressible flow solvers exist for these based on different methods. For incompressible flow, there exist a number of finite element codes using unstructured meshes. See, for example, Guermond and Quartapelle [1], Ravachol [2], Ramamurti and Löhner [3] and Rida *et al.* [4]. In comparison, the body of literature is less developed for finite volume methods as can be seen from the summary of published solvers shown in Table I. Six of the schemes are based on the pressure-correction method of Harlow and Welch [5] marker-and-cell (MAC) or on a similar principle—the SIMPLE schemes of Patankar and Spalding [6] or the pressure-correction (PC) scheme of Hirt and Cook [7]. The artificial compressibility (AC) method of Chorin [8] is used for the remaining two schemes. The table also shows whether the solvers are steady or unsteady, two- or three-dimensional and the type of variable storage used. From this it can be seen that both of the unsteady flow solvers make use of cell-centred storage, which is, on

* Correspondence to: Atmospheric Studies Group, Earth Tech Incorporated, 196 Baker Avenue, Concord, MA 01742-2167, U.S.A.

Received 5 November 1998
Revised 10 December 1999

average, five times as expensive in memory requirements as vertex storage for unstructured meshes; whereas the schemes making use of vertex storage are for steady flow only. This paper seeks to extend the current table by developing a three-dimensional unsteady solver using vertex storage.

Before describing the flow solver, the reason that the pressure-correction solver of Watterson [9] is steady will be explored. This will lead into a discussion of the issues related to collocated vertex storage for the pressure Poisson equation (PPE), which are of paramount importance in the formulation of the present solver.

In Watterson [9], the Poisson equation for pressure is solved approximately with two Jacobi iterations. Numerical experiments show that these iterations have a smoothing effect on the initial guess but do not solve the equation. Indeed, results in Reference [10] suggest that the solver works equally well without any iterations of the Poisson solver. The initial guess is given by

$$p'_k = \frac{-\nabla\cdot\mathbf{u}}{\dfrac{S_\rho}{\Delta t} + \Delta t C_k} \tag{1}$$

where S_ρ accounts for compressibility effects. C_k is a geometric coefficient representing the influence of the pressure correction at node k on the Laplacian at that node. Substituting for this term with the use of approximations for a regular mesh, and assuming incompressibility, the equation for the initial guess reduces to

$$\frac{p'_k}{\Delta t} \approx -\left(\frac{h_k}{2\Delta t}\right)^2 \nabla\cdot\mathbf{u} \tag{2}$$

where h_k is a representative radius for the control volume. This is equivalent to the artificial compressibility method with an artificial speed of sound β given by

Table I. Summary of unstructured incompressible schemes.

Author	Year	Scheme	Steady/ unsteady	Two-/three-dimensional	Discretization/variable storage
Despotis and Tsangaris [28]	1995	MAC	S	2D	Semi-staggered
Thomadakis and Leschziner [29]	1995	SIMPLE	S	2D	Semi-staggered
Murthy Mathur [30]	1997	SIMPLE	S	2D	Collocated cell-centred
Davidson [31]	1996	SIMPLEC	S	2D	Collocated cell-centred
Anderson *et al.* [32]	1996	AC	S	3D	Collocated vertex
Watterson [9]	1994	PC	S	3D	Collocated vertex
Weiss and Smith [33]	1995	AC	U	2D(3D)	Collocated cell-centred
Jiang and Przekwas [34]	1994	SIMPLEC	U	2D(3D)	Collocated cell-centred

$$\beta = \frac{h_k}{2\Delta t} \tag{3}$$

which shows that the pressure-correction step is actually equivalent to an artificial compressibility step with an automatic limit on the propagation distance of pressure waves through the domain. The equivalence between pressure-correction and artificial compressibility is further described in Peyret and Taylor [11], where it is shown that artificial compressibility can be thought of as a method for solving the PPE.

The other pressure correction flow solvers in Table I also only solve the PPE approximately, such that the initial guess remains the dominant factor in the approximate solution. This suggests that they too function more as artificial compressibility solvers than as pressure correction ones, where the initial guess and any relaxation factors applied determine the functioning of the scheme.

The reason that this is so, and that the pressure-correction equation is usually only solved approximately, is that there is an inherent conflict between three factors: obtaining an accurate Laplacian operator, ensuring the compatibility relation and preventing odd–even decoupling of the flow solution. Different schemes addressing these issues will now be reviewed for structured meshes.

The staggered MAC arrangement of variables is used on structured meshes as it automatically guarantees the compatibility relation and prevents odd–even decoupling. This is very expensive and cumbersome on unstructured meshes such that collocation of variables is desirable. Rhie and Chow [12] were the first to describe a solver using collocation of variables on curvilinear grids. This was based on a special interpolation procedure at the cell faces so at to satisfy the compatibility relation and ensure stability by preventing odd–even decoupling. For unstructured meshes, however, it was found that this procedure was not sufficient in satisfying the compatibility relation owing to the truncation errors present in arbitrary meshes.

Briley [13] suggested the addition of correction terms to the source term of the Poisson equation for structured meshes in order to artificially satisfy the compatibility relation. An alternative solution by Henshaw [14] is to augment the system of equations with one extra equation, which could be designed specifically to obtain a solvable system of equations. Sundaresan and Deshpande [15] reported that the satisfaction of the compatibility relation *per se* is not a crucial factor, but the satisfaction of the continuity constraint is crucial. This was described by Gresho and Sani [16] who linked the compatibility relation ('solvability constraint') to the satisfaction of mass conservation and put the stress on the latter.

On a collocated mesh, whether structured or unstructured, satisfaction of the compatibility relation is guaranteed only by a discretization of the Laplacian operator in two parts as the divergence of a gradient operator, where the divergence operator is the same as that used to calculate flow conservation and where the gradient operator is the same as that used to obtain the velocity corrections. When both operators are defined on a control volume consisting of all cells around a node and collocation of variables is used, this leads to an extended discretization of the Laplacian involving all second neighbours of a node. Sotiropoulos and Abdallah [17], working on Cartesian meshes, added correction terms to their compact discretization of the Laplacian to conform it to an extended discretization. The extended discretization was, however, very unstable because it was totally transparent to wiggles in the pressure field. Tafti

[18] solved this problem for structured meshes by developing different gradient and divergence operators that contained higher-order terms designed to prevent odd–even decoupling.

2. FORMULATION OF THE FLOW SOLVER

The present work is concerned with the fully incompressible Navier–Stokes equations as described by

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla\cdot(\mathbf{u}\otimes\mathbf{u}) + \nabla p = \nabla\cdot\underline{\tau} \tag{4}$$

$$\nabla\cdot\mathbf{u} = 0 \tag{5}$$

$$\rho = \text{constant} \tag{6}$$

where $\underline{\tau}$ is the shear stress tensor.

The solution of the following semi-discretized system of equations is sought:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -\nabla\cdot(\mathbf{u}\otimes\mathbf{u})^n - \nabla p^n + \nabla\cdot\underline{\tau}^n \tag{7}$$

$$\nabla\cdot\mathbf{u}^{n+1} = 0 \tag{8}$$

The flow solver follows the pressure-correction method of Hirt and Cook [7], with an explicit prediction step of the momentum equation followed by a correction step to satisfy the continuity equation. The predicted velocity is evaluated from Equation (9) and will be used in Equation (10) to obtain the final velocity. Note that the pressure term is at the correct time level, as discussed in Gresho *et al.* [19].

$$\mathbf{u}^* = \mathbf{u}^n - \Delta t(\nabla\cdot(\mathbf{u}\otimes\mathbf{u})^n + \nabla p^{n-1} - \nabla\cdot\underline{\tau}^n) \tag{9}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* + \mathbf{u}' \tag{10}$$

Taking the divergence of the semi-discretized momentum equation (7) and substituting the relations for the predicted velocity in Equations (9) and (10), yields the PPE (Equation (11)) in terms of the pressure-correction (Equation (12))

$$\nabla\cdot\nabla p' = \frac{\nabla\cdot\mathbf{u}^*}{\Delta t} \tag{11}$$

$$p^n = p^{n-1} + p' \tag{12}$$

The PPE is solved and the velocity correction is obtained from Equation (13) to update the velocity as described above

$$\mathbf{u}' = -\Delta t \nabla p' \tag{13}$$

The stability limit for forward Euler integration is given by a combined advection–diffusion limit as described by Gresho *et al.* [20]. To preserve time accuracy, the minimum time step over the whole domain is used as a global time step, which is very restrictive.

The sub-cycling procedure of Gresho *et al.* [20] was tested in an attempt to increase the time step between pressure-correction steps. The prediction step is repeated a number of times before carrying out a single correction step—with the advantage that the prediction steps are much cheaper than the correction step and therefore a larger effective time step is obtained at a lower cost. This, however, was not found to increase the allowable time step between pressure-correction steps. Gresho *et al.* suggested halting the sub-cycling procedure when the norm of the divergence of the prediction velocity reaches a certain threshold. Taking this as a criterion in the present solver shows that larger time steps cannot be obtained as the norm of the divergence increases very rapidly with each sub-cycle. This is due to the much larger truncation errors associated with unstructured meshes compared with structured or regular meshes. The simple forward Euler time integration was therefore retained.

Equation (14) was used for the time step, with an empirically defined $c = 0.5$, where Δl is the smallest cell height around a node and $|\mathbf{u}|$ is the magnitude of the velocity at that node

$$\Delta t_{\text{global}} = \Delta t_{\min} = c\left(\frac{\Delta l}{|\mathbf{u}|}\right)_{\min} \tag{14}$$

3. BOUNDARY CONDITIONS

Spatial boundary conditions are applied for the velocity during the prediction step and for both the velocity and the pressure during the correction step. Globally, the inflow velocity is specified, the no-slip condition is enforced on viscous walls and the velocity is tangential at inviscid walls while the pressure is fixed at outflow.

During the prediction step, the velocity is specified at the inflow and on viscous walls and is extrapolated elsewhere. The correction step affects both the momentum and the continuity equations, such that boundary conditions need to be consistent and satisfy both of these equation at the same time. Dirichlet conditions are enforced on the pressure-correction at outflow, thereby specifying the exit pressure. Neumann conditions on the pressure-correction are used elsewhere. These are designed such that at inflow a zero velocity correction is obtained while at solid boundaries the wall-normal velocity correction balances out the normal component of the predicted velocity leading to tangential velocities at the new time step as shown in Equation (15).

$$\begin{array}{ll} \text{Inflow b.c.} & \nabla p' \cdot \mathbf{n} = 0 \\ \text{Outflow b.c.} & p' = 0 \\ \text{Wall b.c.} & \nabla p' \cdot \mathbf{n} = \dfrac{\mathbf{u}^* \cdot \mathbf{n}}{\Delta t} \end{array} \tag{15}$$

The above condition on the pressure-correction at the wall is equivalent to a condition of zero acceleration on the normal component of the momentum equations as shown in Equation (16), such that the conditions are consistent with each other

$$(-\nabla \cdot (\mathbf{u} \otimes \mathbf{u})^n - \nabla p^n + \nabla \cdot \underline{\tau}^n) \cdot \mathbf{n} = \left(\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} \right) \cdot \mathbf{n} = 0 \tag{16}$$

The normal at a boundary was taken to be an area-weighted average of the normals of each boundary face around a node. Specifying a zero normal velocity component with this definition can be seen to be different from specifying zero mass flow through the individual faces, such that some leakage can be expected to occur on highly curved boundaries and especially on boundary nodes that are at the edges between two boundary surfaces. This introduced a global mass flux error in the flow solution, although this was usually less than 0.1% of the mass flow.

The initial condition is taken to be a divergence-free flow. In practice, a potential flow solution is used to begin the solution if a beginning solution is not available. This is equivalent to solving flow for an impulse start, as shown by Batchelor [21], and discussed in depth in Reference [19].

4. SPATIAL DISCRETISATION AND THE POISSON SOLVER

The flow variables are collocated at cell vertices and assumed to be piecewise linear over each tetrahedron. The prediction step, Equation (7), is discretized using the integral form of the equations following the finite volume procedure as described by Dawes [22]. Average face fluxes for each face f are summed over the control volume consisting of all cells j around a node k, as shown in Figure 1. At boundaries, the half volume consisting of all cells touching on a boundary node is used. This is depicted in two dimensions in Figure 2, where fluxes are summed over the boundary faces as well as those opposite to the node.

The use of centred spatial discretization with forward Euler time integration leads to an underdiffusive scheme. Balancing tensor diffusivity, see Gresho *et al.* [20], could be used to offset this. In the present work, however, artificial dissipation is used as a simple yet effective expedient as described in the next section.

Discretization of the Poisson equation is the critical step in the formulation of a pressure-correction solver, as described above. In order to satisfy the continuity constraint exactly, irrespective of mesh quality, the gradient operator used as the first step of the Laplacian in Equation (11) must be the same as the operator used to obtain the final velocity correction in Equation (13). This must also be the same as the operator used to define the Neumann

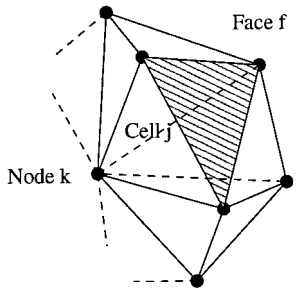

Figure 1. Finite volume discretization.

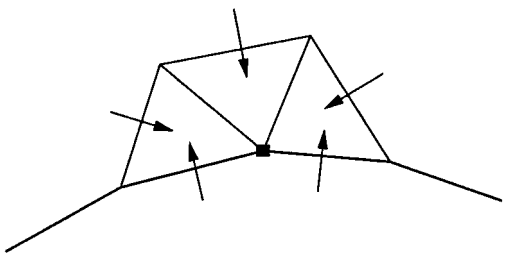

Figure 2. Discretization of Neumann boundary conditions.

boundary conditions in Equation (15). For the velocity correction to balance the divergence field of the prediction step, the divergence operator on the right- and left-hand sides of Equation (11) must also be the same. The finite volume procedure described above is used for both the gradient and the divergence operator. Figure 3 shows the flux sums used to obtain the gradient operator at a node in two dimensions. On the right, in dashed lines is the control volume contributing to the gradient operator and in solid lines is the new control volume for the divergence operator at the centre node. This can be seen to lead to an extended discretization of the Laplacian operator similar to the Cartesian operator of Sotiropoulos and Abdallah, but making use of all second neighbours to a node, not just those in the cardinal directions. While this discretization is necessary to preserve consistency of the pressure and velocity corrections, it is prone to pressure–velocity decoupling, which will be addressed in the section on artificial dissipation. For the boundary conditions, the half-volume described above is used to obtain the gradient operator at a boundary node.

The PPE is often solved with stationary iterative methods, such as Jacobi or Gauss–Seidel iteration, which rely on diagonal dominance for convergence. The diagonal term of the Laplacian is equal to the sum of the off-diagonal terms. On structured meshes, all of these are positive, such that the system of equations is neutrally dominant and relies on boundary conditions for dominance. For unstructured meshes, however, numerical experiments show that these terms are often negative, even for seemingly regular meshes, such that the system of

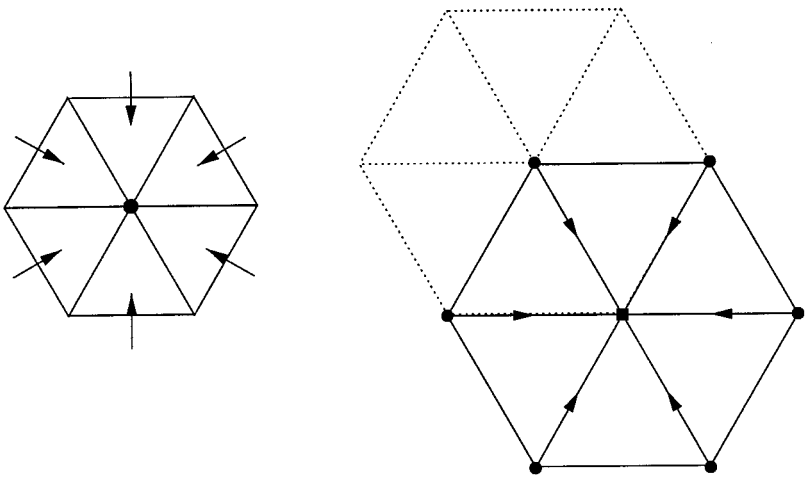

Figure 3. Extended discretization of Laplacian operator.

equations is not diagonally dominant. The boundary conditions make matters worse as the Neumann conditions have large off-diagonal terms, thereby excluding the use of stationary iterative methods.

Conjugate gradient solvers are often used for the PPE. The basic conjugate gradient solver works very well for symmetric matrices. While the terms of the Laplacian operator are symmetric, the boundary conditions introduce asymmetric terms so that the biconjugate gradient stabilized (Bi-CGSTAB) method of van der Vorst [23] is used.

The Laplacian of the pressure-correction can be performed by evaluating the divergence and gradient operators sequentially using flux sums over the control volume. In practice, however, it was found that it was up to eight times faster to store coefficients explicitly and perform the Laplacian as a matrix multiplication. Coefficients for the effect of each node on the Laplacian at a node connected to it by an edge, or at the node itself, are calculated and stored. This incurred a substantial memory overhead as coefficients are stored for all second neighbours. Because CPU time, rather than memory, was the limiting factor, this was deemed acceptable. The coefficients were stored in the sparse column format. This enabled the use of ILU preconditioning using zero fill-in, which in this case corresponds to using all second neighbours of a node. The actual preconditioner used was taken from the netlib library.

Reverse Cuthill–McKee renumbering [24] was used to speed up the flow solver in two ways. First, it reduces the bandwidth of the Laplacian matrix by up to a factor of ten, such that the ILU preconditioner is a much improved approximation of the Laplacian matrix. This leads to a factor of reduction of approximately seven in the number of iterations required to reach convergence for a number of test cases. Second, the matrix multiplication is performed approximately twice as fast because a more efficient node numbering leads to more of the multiplied terms being within the memory cache.

5. ARTIFICIAL DISSIPATION

The solution scheme uses forward Euler integration with centred finite volume discretization, which is known to be underdiffusive such that artificial dissipation is necessary on the momentum equations to preserve stability. Fourth-order edge sum smoothing is used based on the work of Mavriplis [25]. A second difference is constructed as an approximation to the Laplacian using the edge sum procedure described by Equation (17). For a variable w, defined at all nodes, the second difference is the sum of the difference over all edges connecting the central node k to its first neighbours n.

$$\mathrm{d}^2 w_k = \sum_n [w_k - w_n] \tag{17}$$

The process is then repeated with the inclusion of scaling terms in order to obtain a fourth difference.

$$\mathrm{d}^4 w_k = \epsilon_4 \frac{\Delta t}{\mathrm{Vol}_{CV}} \sum_n \left(\frac{\Lambda_k + \Lambda_n}{2} \right) [\mathrm{d}^2 w_k - \mathrm{d}^2 w_n] \tag{18}$$

$$\Lambda_k = \sum_{\text{faces}} |\mathbf{u} \cdot \Delta \mathbf{A}| + a|\Delta \mathbf{A}| \tag{19}$$

Vol_{CV} is the volume of all the cells around node k. $\Delta\mathbf{A}$ is the area vector of a cell face on the outside of the control volume around a node, with the summation carried out over all of these. In Mavriplis, a is the speed of sound and Λ_k is an approximation to the maximum eigenvalue for the Euler equations for compressible flow. This is no longer applicable for incompressible flow. However, a is used as an empirical constant to preserve smoothing in stagnation regions and was assigned a value of 400. The smoothing parameter ϵ_4 was empirically set to 0.0001, with some variation based on the individual test cases. Artificial dissipation is added to the velocity field on the right-hand side of the prediction step (9) ensuring that continuity is not corrupted by the smoothing.

Smoothing was essential for pressure as the extended discretization of the Laplacian is totally transparent to wiggles leading to velocity–pressure decoupling. This was added explicitly when the pressure field was updated with the pressure correction in Equation (12) and was found to control instabilities without corrupting the solution. It in effect enforces an extra requirement implicit in the pressure-correction approximation—that the pressure field has a smooth and continuous second derivative.

6. VALIDATION

The order of accuracy of the flow solver was verified on the two-dimensional inviscid flow over a hump for an extruded mesh three nodes thick. Calculations were performed on three different meshes with regular equilateral cells in the x–y plane, containing $(6 \times 16 \times 3)$,

$(11 \times 31 \times 3)$ and $(21 \times 61 \times 3)$ nodes. Two different measures of loss were used, a global one and one based on the local production of stagnation pressure loss.

The global loss coefficient was obtained by integrating the stagnation pressure loss at the exit with respect to the mass flux, as described by

$$Yp_g = \frac{1}{\frac{1}{2}\rho U_1^2} \frac{\int_{\text{exit}} (Po - Po_1)\, \mathrm{d}\dot{m}}{\int_{\text{exit}} \mathrm{d}\dot{m}} \tag{20}$$

The second measure was obtained at every node by summing the mass integrated fluxes of stagnation pressure of the control volume, as defined by

$$Yp_{CV} = \frac{1}{\frac{1}{2}\rho U_1^2} \frac{\sum (Po - Po_1)\Delta\dot{m}}{\sum \Delta\dot{m}} \tag{21}$$

A global figure was obtained from this measure by taking the root-mean-square (r.m.s.) of the nodal values.

The resulting loss coefficients are plotted on a logarithmic scale versus mesh spacing in Figures 4 and 5. The gradient of Yp_g was 3.0 and that of Yp_{CV} was 1.7. A value of 3.0 is very high for a solver that uses only second-order accurate discretizations for the convection and viscous fluxes. A value of 1.7 is to be expected from a second-order accurate flow solver, and is lower than 2.0 due to the inaccuracies introduced by the irregularity of the mesh and due to the artificial dissipation scheme used. Yp_g was found to yield a less consistent measure of the order of accuracy of the solver on a range of different flow solutions. This is due to its sensitivity to very small mass flow leakages in the domain. For this particular case, there is leakage along the edges between the hump and the side walls of the mesh as discussed above.

A laminar boundary layer was calculated in order to test the basic viscous capability of the solver. The domain extended one chord length upstream of the leading edge and 2.6 chord lengths in the normal direction. The mesh contained roughly ten nodes in the boundary layer. Figure 6 shows the growth of the displacement and momentum thicknesses versus the ideal Blasius solution, denoted by circles, showing good agreement between the two.

Unsteady vortex shedding behind a circular cylinder was then calculated to validate the unsteady behaviour of the flow solver. Different domain sizes of (10×20), (25×30) and (35×50) diameters were calculated, with the last two showing very little variation suggesting that blockage effects had become negligible. A grid refinement study was also performed using meshes with 1066, 2771, 4000 and 6893 nodes per x–y plane. There was negligible variation in the results for the last three meshes. The results presented were calculated on the mesh with 4000 nodes per plane. The Strouhal number obtained at different Reynolds numbers is shown in Figure 7. They are slightly underpredicted in the range below 500. This may be attributed

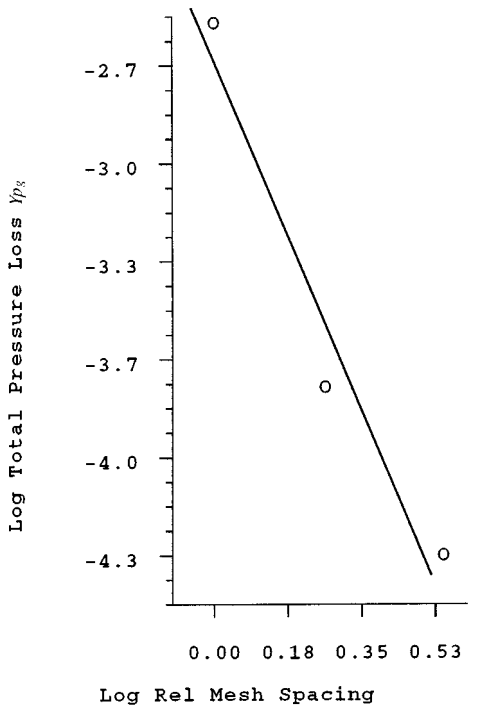

Figure 4. Total pressure loss coefficient Yp_g versus relative mesh spacing. Gradient = 3.0.

to the fact that the flow solver is first-order accurate in time whereas the results of Rosenfeld *et al*. [26], for example, were obtained with second-order accuracy in time. The forward Euler integration is underdiffusive but the addition of artificial dissipation leads to an increase in the actual viscosity, leading to lower values of the predicted Strouhal number.

7. RESULTS

Results were calculated for three test cases: a stalling airfoil, flow past a bridge deck and flow in model alveoli.

In the first case, the dynamic stall of flow at a Reynolds number of 1000 over a NACA0012 airfoil at an incidence of 34° was calculated. The results are compared with those of Guermond and Quartapelle [1] in Figure 8, showing very good agreement with their two-dimensional flow solver.

Flow over the bridge deck of the Great Belt East suspension bridge under construction in Denmark was calculated for the second case. The design of long span bridges is limited by the wind-induced motions of the bridge, leading to mounting interest in predicting the lift and drag

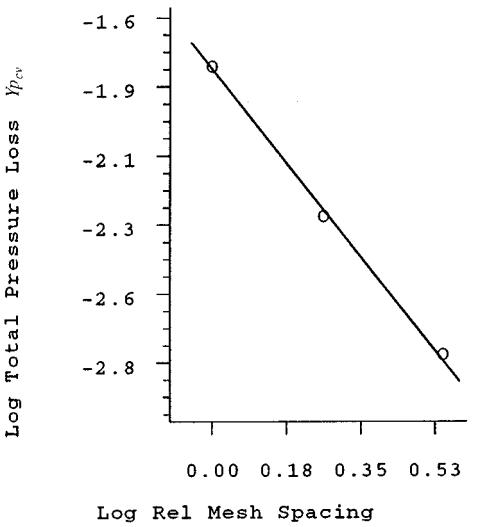

Figure 5. Total pressure loss coefficient Yp_{CV} versus relative mesh spacing. Gradient = 1.7 respectively.

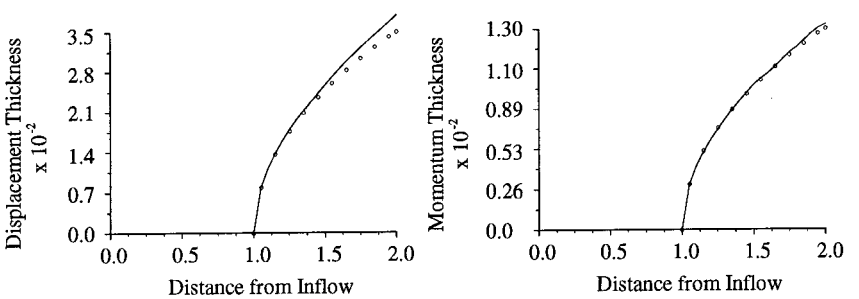

Figure 6. Growth of a laminar boundary layer versus streamwise distance. The exact Blasius solution is denoted by ○ for comparison.

forces as well as the Strouhal numbers for flow over these geometries on both steady meshes and on moving meshes. For the latter, the calculations are coupled with finite element calculations of the bridge movements. Results are presented in Table II alongside the experimental results of Larsen [27]. The Strouhal number is correctly predicted. The drag coefficient is underpredicted because the crash barriers were not included in the calculation and no turbulence model was used. Figure 9 shows streamlines along with shading of vorticity.

While the test cases presented so far have been two-dimensional, they were calculated on three-dimensional meshes with three parallel planes of nodes. To demonstrate the three-dimensional capability of the code, the third case involved the preliminary calculation of flow

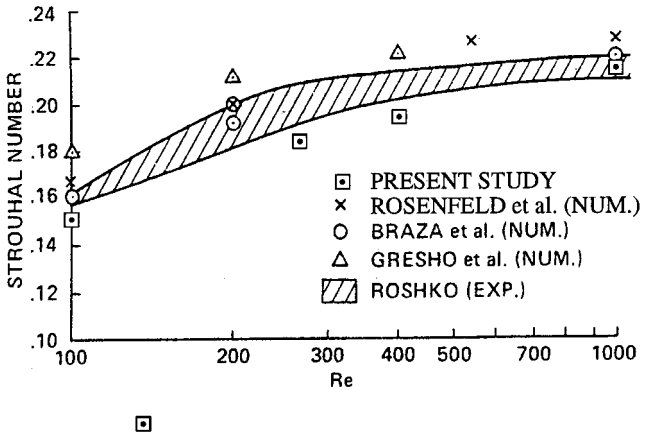

Figure 7. Strouhal number versus Reynolds number for vortex shedding behind a cylinder (from Rosenfeld *et al.* [26]).

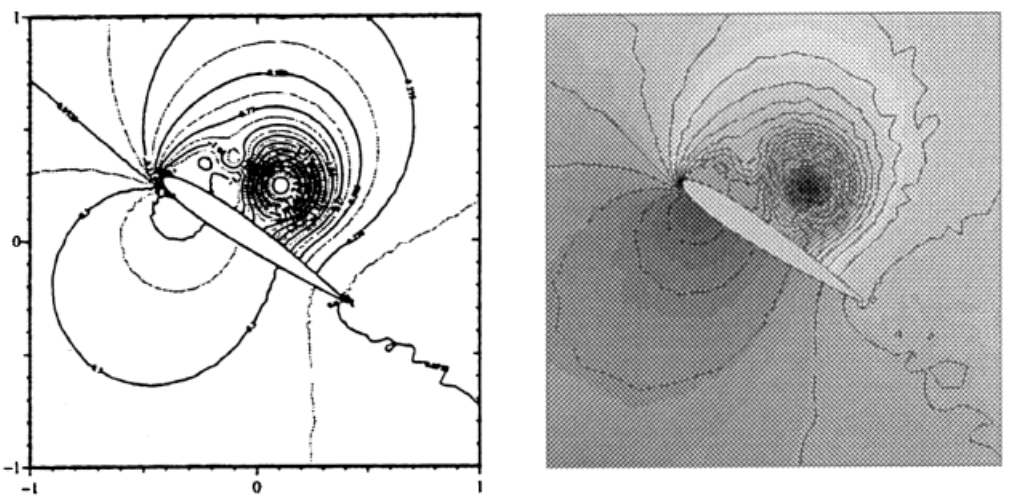

Figure 8. Pressure coefficient contours at $T = 1.6$. Left: results of Guermond and Quartapelle [1]; right: current results (contour interval $Cp = 0.23$).

in a model alveoli geometry calculated with the same flow solver as the two-dimensional cases. There is increasing use of CFD in biofluids simulation, with interest in the unsteady nature of flow in the lungs due to opposite considerations—the desire to maximize drug transport from inhalers and the estimation of the effects of deposition of particulates pollution. Figure 10 shows stream ribbons for the steady low Reynolds number flow in a geometry representing several alveoli.

Table II. Strouhal number, drag coefficient and lift coefficient for the Great Belt East suspension bridge section.

	Strouhal number	r.m.s. (C_D)	$C_{L_{max}}$	$C_{L_{min}}$	r.m.s. (C_L)	Av (C_L)
Present paper	0.17	0.38	0.237	−0.120	0.14	0.05
Larsen [27]	0.16	0.54	—	—	—	0.01

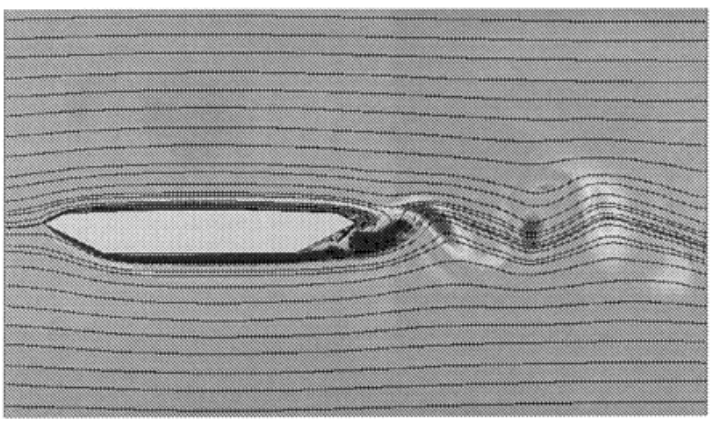

Figure 9. Instantaneous streamlines and vorticity shading for flow past the Great Belt East suspension bridge.

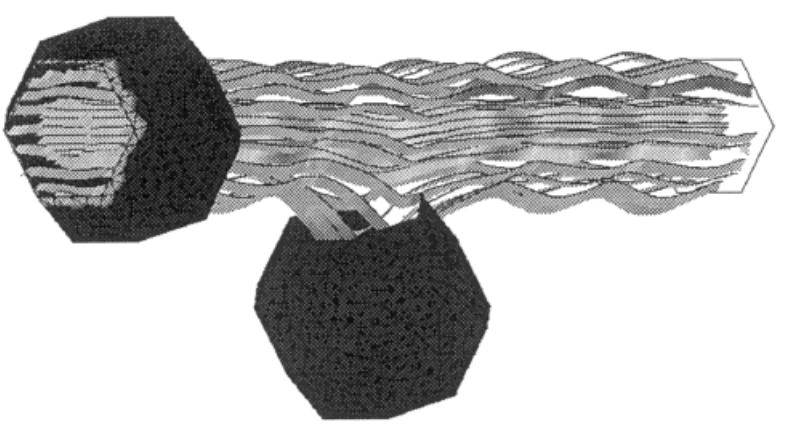

Figure 10. Model alveoli geometry, shading of ribbons with x-velocity.

8. CONCLUSION

A flow solver was developed for unsteady, incompressible, three-dimensional flow for unstructured meshes using collocation of variables at the cell vertices. The method was based on the pressure-correction method. A consistent discretization of the Laplacian operator was found to be essential in obtaining the correct solution of the Poisson equation for pressure. The boundary conditions imposed on this equation were designed so as to be consistent with the conditions on the momentum equations.

The speed of the code is determined by the computational cost of solving the Poisson equation and by the size of the time step used for the prediction step. Smoother meshes and/or more regular operators may enable increased time steps for the predicted velocity. Different preconditioning techniques as well as renumbering techniques may yield improvements in the robustness of the Poisson solver and its convergence.

Turbulence models were not implemented in the current work. In order to extend the applicability of the flow solver, this should be performed using one of the standard models.

While the flow solver is time accurate, moving meshes were not implemented. This may be the next step of interest to calculate an oscillating bridge or a breathing lung.

ACKNOWLEDGMENTS

The first author is very grateful to St John's College, Cambridge, for support via a Benefactor's Scholarship and for a grant from the EPSRC, UK. J.B. Frandsen and F.A. McRobie of the Engineering Department of Cambridge University provided the information and the impetus for the bridge aerodynamics test case. Dr A. Leeming of the Centre for Biological and Medical Systems at Imperial College was instrumental in the bio-medical application of the code and generated the mesh used for the flow calculations.

REFERENCES

1. Guermond J-L, Quartapelle L. Finite element solution of unsteady viscous flows past multiple airfoils. In *15th International Conference on Numerical Methods in Fluid Dynamics*, Monterey, CA, USA, June, vol. 490 of Lecture Notes in Physics. Springer Verlag: Berlin, 1996; 566–571.
2. Ravachol M. Unstructured finite element for incompressible flows. In *13th AIAA Computational Fluid Dynamics Conference*, Snowmass Village, CO, USA, vol. 1, 1997; 290–296. AIAA 97-1864.
3. Ramamurti R, Löhner R. A parallel implicit incompressible flow solver using unstructured meshes. *Computers and Fluids* 1996; **25**(2): 119–132.
4. Rida S, McKenty F, Meng FL, Reggio M. A staggered control volume scheme for unstructured triangular grids. *International Journal for Numerical Methods in Fluids* 1997; **25**: 697–717.
5. Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids* 1965; **8**: 2182–2189.
6. Patankar SV, Spalding DB. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer* 1972; **15**: 1787–1806.
7. Hirt CW, Cook JL. Calculating three-dimensional flows around structures and over rough terrain. *Journal of Computational Physics* 1972; **10**: 324–340.
8. Chorin AJ. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics* 1967; **2**: 12–26.
9. Watterson JK. A pressure-based flow solver for the three-dimensional Navier–Stokes equations on unstructured and adaptive meshes. In *25th AIAA Fluid Dynamics Conference*, Colorado Springs, CO, USA, June, 1994. AIAA 94-2358.

10. Watterson JK. A new, pressure-based, unstructured mesh, Navier–Stokes solver and application to 3D compressible vortex/boundary layer interactions. PhD thesis, University of Cambridge, UK, 1994.
11. Peyret R, Taylor TD. *Computational Methods for Fluid Flow*. Springer: Berlin, 1983.
12. Rhie CM, Chow WL. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal* 1983; **21**(11): 1525–1532.
13. Briley WR. Numerical method for predicting three-dimensional steady viscous flow in ducts. *Journal of Computational Physics* 1974; **14**: 8–28.
14. Henshaw WD. A fourth-order accurate method for the incompressible Navier–Stokes equations on overlapping grids. *Journal of Computational Physics* 1994; **113**: 13–25.
15. Sundaresan S, Deshpande SM. On pressure compatibility condition in numerical simulation of incompressible viscous flows using primitive variable formulation. In *15th International Conference on Numerical Methods in Fluid Dynamics*, vol. 490 of Lecture Notes in Physics, Kutler P, Flores J, Chattot J-J (eds). Springer: Berlin, 1996.
16. Gresho PM, Sani RL. On pressure boundary conditions for the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 1987; **7**: 1111–1145.
17. Sotiropoulos F, Abdallah S. The discrete continuity equation in primitive variable solutions of incompressible flow. *Journal of Computational Physics* 1991; **95**: 212–227.
18. Tafti D. Alternate formulations for the pressure equation Laplacian on a collocated grid for solving the unsteady incompressible Navier–Stokes equations. *Journal of Computational Physics* 1995; **116**: 143–153.
19. Gresho PM, Sani RL, Engelman MS. *Incompressible Flow and the Finite Element Method*. Wiley: New York, 1998.
20. Gresho PM, Chan ST, Lee RL, Upson CD. A modified finite element method for solving the time-dependent, incompressible Navier–Stokes equations. Part 1: theory. *International Journal for Numerical Methods in Fluids* 1984; **4**: 557–598.
21. Batchelor GK. *An Introduction to Fluid Dynamics*. Cambridge University Press: Cambridge, 1967.
22. Dawes WN. Simulation of three-dimensional viscous flow in turbomachinery geometries using a solution-adaptive unstructured mesh methodology. *Journal of Turbomachinery* 1992; **114**(3): 528–537.
23. van der Vorst HA. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing* 1992; **13**(2): 631–644.
24. Cuthill E, McKee J. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings 24th ACM National Conference*. Brandon Press: New Jersey, 1969; 157–172.
25. Mavriplis DJ. Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes. *AIAA Journal* 1990; **28**(2): 213–221.
26. Rosenfeld M, Kwak D, Vinokur M. A fractional step solution method for the unsteady incompresible Navier–Stokes equations in generalized coordinate systems. *Journal of Computational Physics* 1991; **94**(1): 102–137.
27. Larsen A. Storebaelt, East Bridge, tender evaluation, suspension bridge alternative sections, section model tests, I. Technical report 91023-10.00 (Restricted), DMI, March, 1993.
28. Despotis GK, Tsangaris S. Fractional step method for solution of incompressible Navier–Stokes equations on unstructured triangular meshes. *International Journal for Numerical Methods in Fluids* 1995; **20**: 1273–1288.
29. Thomadakis M, Leschziner M. A pressure-correction method for the solution of incompressible viscous flows on unstructured grids. *International Journal for Numerical Methods in Fluids* 1996; **22**: 581–601.
30. Murthy JY, Mathur S. Periodic flow and heat transfer using unstructured meshes. *International Journal for Numerical Methods in Fluids* 1997; **25**: 659–677.
31. Davidson L. A pressure correction methods for unstructured meshes with arbitrary control volumes. *International Journal for Numerical Methods in Fluids* 1996; **22**: 265–281.
32. Anderson WK, Rausch RD, Bonhaus DL. Implicit/multigrid algorithms for incompressible turbulent flows on unstructured grids. *Journal of Computational Physics* 1996; **128**: 391–408.
33. Weiss JM, Smith WA. Preconditioning applied to variable and constant density flows. *AIAA Journal* 1995; **33**(11): 2050–2057.
34. Jiang Y, Przekwas AJ. Implicit, pressure-based incompressible Navier–Stokes equations solver for unstructured meshes. In *32nd Aerospace Sciences Meeting*, Reno, NV, USA, January, 1994; AIAA 94-0305, 1–11.

