European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2006
P. Wesseling, E. Oñate, J. Périaux (Eds)

SIMULATION OF LOW-MACH-NUMBER FLOW USING A FULLY-COUPLED IMPLICIT RESIDUAL-DISTRIBUTION METHOD

Nicholas Waterson* and Herman Deconinck†

*Mott MacDonald Limited, St Anne House, Wellesley Road, Croydon CR9 2UL, UK.
e-mail: nicholas.waterson@mottmac.com
Web page: http://www.mottmac.com

†von Karman Institute for Fluid Dynamics,
Waterloosesteenweg 72, B-1640 Sint-Genesius-Rode, Belgium.
e-mail: deconinck@vki.ac.be
Web page: http://www.vki.ac.be

Key words: residual distribution, convection, multidimensional upwind, coupled

Abstract. *An effective approach is presented for the numerical solution of the equations governing laminar and turbulent flow, heat and mass transfer at low Mach number. The approach adopted combines a compact and accurate discretization using the residual-distribution (RD) approach with a fully-coupled implicit solution procedure.*

The system RD approach adopted employs genuinely-multidimensional upwinding to achieve accurate and stable discrete equations on a highly compact computational stencil. This combines naturally with a fully-implicit coupled solution procedure for which the number of non-linear iterations required is essentially independent of the grid size. This contrasts with other widely-used segregated approaches in which the pressure-velocity system is discretized and solved as a set of scalar equations. The compact nature of the discretization allows the full convection and diffusion terms in all equations to be treated implicitly without any form of deferred correction.

The present implementation solves the 2D, axisymmetric or 3D, laminar or Reynolds-averaged Navier-Stokes equations in incompressible or weakly-compressible form on unstructured grids of triangles or tetrahedra. The RD form of the system Lax-Wendroff scheme is applied to the convection and pressure terms while the viscous terms are treated using the Galerkin finite-element method. The system scheme for convection provides natural stabilization, allowing a collocated variable arrangement to be used. The discrete equation system is solved using a fully-coupled implicit approach based on Picard/Newton linearization with the linear system solved using standard Krylov subspace methods (e.g. GMRES or BiCGStab) with ILU(0) preconditioning.

The approach has been extensively validated on a range of test cases. Three cases are presented here for steady laminar flow in a shear-driven cavity, a two-level bifurcation (both 2D) and a 90º pipe bend (3D). These include direct comparisons with commercial unstructured flow solvers with very promising results – showing equivalent levels of accuracy but reduced computational times and less sensitivity to grid quality.

1 INTRODUCTION

Despite the highly coupled nature of the governing equations of fluid flow, incompressible and low-Mach-number flows are frequently both discretized and solved using techniques which treat them as a set of scalar equations rather than as an equation system[1]. The approach presented here applies a system discretization technique to create a set of discrete equations which are then solved using a fully-coupled implicit procedure. Of particular significance is the highly compact nature of the residual-distribution scheme adopted which allows almost all terms to be treated in a fully implicit manner without recourse to deferred correction.

The residual-distribution (RD) method, also referred to variously as the cell-vertex or fluctuation-splitting approach, originates from the work of Ni[2], later extended by Denton[3], Hall[4], Morton *et al.*[5], Giles *et al.*[6], Roe[7], Deconinck *et al.* [8,9] and others. It uses a cell-vertex arrangement, with the flux balances or residuals calculated over the primary grid cells and distributed in some manner to the computational nodes which lie at their vertices, as shown in Figure 1. The accumulated nodal residuals are then used to update the nodal solution values in either an explicit or implicit manner.

Residual-distribution methods have attracted particular interest because they provide a very convenient framework in which to construct convection schemes which are genuinely multidimensional in nature, rather than applying upwinding along arbitrary grid directions. This attempt to reflect better the underlying physics of the governing equations leads to schemes which are both less dependent upon grid quality and achieve accurate solutions using highly compact computational stencils. Until recently applications of the RD approach have focused in particular on high-speed aerodynamics while applications of the method to low-speed flow, the subject of the present study, have remained relatively neglected.

The present paper describes the main principles of the approach developed and demonstrates its behaviour on three laminar, isothermal test cases, chosen to illustrate the key features of its performance. Further detail of the method and results for cases with turbulence and heat transfer are given by Waterson[10].

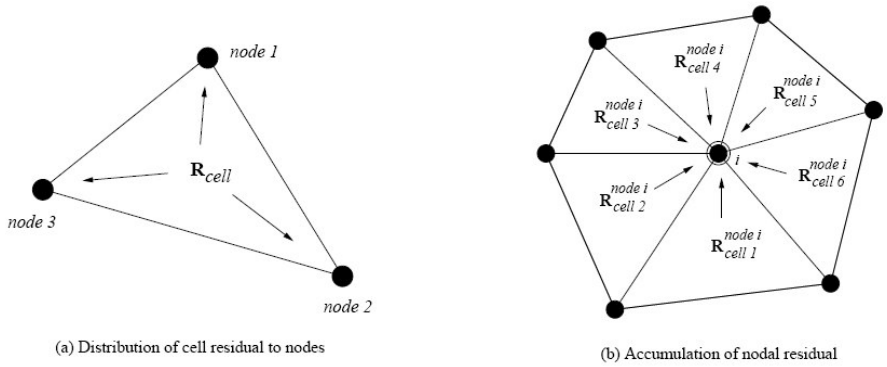
2 DISCRETIZATION APPROACH

In common with the vertex-centred finite-volume and linear finite-element approaches, the computational nodes are located at the vertices of the primary grid cells, as shown in Figure 1. The basic approach may be summarized in the following steps, also illustrated in Figure 1:

1. *Compute* a cell residual, r_c (scalar case), or residual vector, R_c (system case), for each primary grid cell using the solution values stored at the cell vertices.
2. *Distribute* the cell residual to the nodes of the cell, leading to the accumulation of a set of nodal residuals, r_i (scalar case), or residual vectors, R_i (system case):

$$R_i = \sum_{c \in \Omega_i} R_c^i \tag{1}$$

3. *Update* the solution values at the nodes using the accumulated nodal residuals in either an explicit or implicit manner.



(a) Distribution of cell residual to nodes

(b) Accumulation of nodal residual

Figure 1: Diagrams of residual distribution (left) and accumulation (right)

For non-linear equations or discretizations these steps must be repeated until a converged solution is obtained (when the nodal residuals R_i have been driven close to zero). It should be noted that the first two steps can be achieved in a single loop over all of the primary grid cells while the third step requires a further loop over all of the computational nodes. The present section will focus on the first two steps while the following, concerning the solution procedure, will deal with the update step.

While the residual-distribution approach is not limited to triangles and tetrahedra, the present implementation has been restricted to these simplex elements in order to exploit the associated simplicity and efficiency. The discretization is defined entirely using the geometry of the primary grid cells over which it is assumed that the dependent variables vary linearly. The only geometric data required are the vertex coordinates, the scaled cell-face normals and the cell volumes. In the computer code developed only the vertex coordinates are stored, from which all other geometric data are recomputed as required. All expressions used here are common to both 2D and 3D and this synergy extends to the computer code itself (largely avoiding the need for conditional statements based on the number of dimensions).

The following three quantities are employed:

- N_d : number of dimensions (*i.e.* either 2 or 3);
- N_{vc} : number of vertices per grid cell (3 for triangles, 4 for tetrahedra);
- N_{vf} : number of vertices per cell face (2 for triangles, 3 for tetrahedra).

For the triangular and tetrahedral elements used here, the primary cell faces (and their associated geometric properties and fluxes) are numbered according to the node opposite the face with the face normals chosen to be inward facing.

2.1 Equation system

In the present work only steady-state solutions of the governing equations have been considered (though an extension to unsteady flow is quite feasible). The equations solved describe either incompressible or low-Mach-number compressible flow in either conservative or non-conservative forms. For the isothermal cases considered here, the dependent variables chosen are the relative static pressure, p, and the velocity, $\boldsymbol{u}$. For brevity and clarity only the two-dimensional, laminar, isothermal form will be described here for which the vector of

dependent variables, U, is $(p\ u\ v)^{\mathrm{T}}$. The governing equations can be written in the following system form:

$$\boldsymbol{A}\cdot\nabla U = \nabla\cdot\boldsymbol{F}_v + S \tag{2}$$

where $\boldsymbol{A} = A\mathbf{i} + B\mathbf{j}$ is the vector of system matrices comprising all of the convection and pressure terms, $\boldsymbol{F}_v$ is the vector of viscous or diffusive fluxes and S is the vector of source terms. The system matrices are:

$$A = \begin{pmatrix} - & \rho c^2 & - \\ 1 & \rho u & - \\ - & - & \rho u \end{pmatrix},\ B = \begin{pmatrix} - & - & \rho c^2 \\ - & \rho v & - \\ 1 & - & \rho v \end{pmatrix} \tag{3}$$

in which it can be seen that the continuity equation has been multiplied by the square of a scaling velocity, c, analogous to the sound speed used in artificial-compressibility methods, which controls the conditioning of the final discrete equation system. The system has a set of three real eigenvalues (in which $\boldsymbol{n}$ is an arbitrary real vector):

$$\rho\boldsymbol{u}\cdot\boldsymbol{n} \text{ and } {}^1\!/_2\{\rho\boldsymbol{u}\cdot\boldsymbol{n} \pm [(\rho\boldsymbol{u}\cdot\boldsymbol{n})^2 + 4\rho c^2]^{1/2}\} \tag{4}$$

showing that as long as c is of the same order as $|\boldsymbol{u}|$ then the system can be expected to be reasonably well conditioned, which has important implications for solution convergence.

2.2 Cell residual

The viscous terms are dicretized using a standard Galerkin finite-element treatment while the pressure-velocity system is treated using a system residual-distribution approach. The convective cell residual, R_c, for a system of equations written in quasi-linear form, can be obtained by integrating the left-hand side of equation (2) over a primary grid cell:

$$\int_{\Omega_c} A\cdot\nabla U d\Omega = \int_{\Omega_c} A d\Omega\cdot\nabla U \tag{5}$$

where ∇U can be assumed to be constant over a grid cell as U is assumed to vary linearly. The discrete cell residual can be written in the following convenient form:

$$R_c = \frac{1}{N_{vf}}\tilde{\boldsymbol{A}}_c\cdot\sum_{j\in\Omega_c} U_j\boldsymbol{n}_j = \sum_{j\in\Omega_c} K_j U_j \tag{6}$$

in which $\tilde{\boldsymbol{A}}_c$ is the vector of system matrices based on cell-average quantities and the K_j are termed *inflow matrices*. In the scalar case the analogous *inflow coefficients* are positive in the case of an inflow opposite node j and negative for an outflow, see Figure 2. Both inflow matrices and coefficients have the property of summing to zero over a cell. It should be noted that the above discretization is non-conservative at the level of the primary grid cells *i.e.* the fluxes through the primary cell faces do not sum precisely to zero. Various alternative conservative discretizations have been considered however it has been found that these were less robust with respect to convergence and for most of the low-Mach-number cases considered made little difference to the solution.

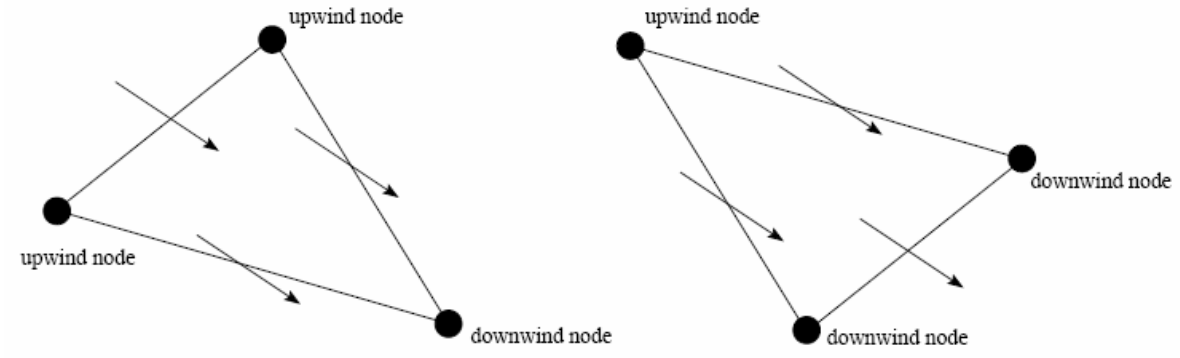


Figure 2: One-inflow (left) and two-inflow (right) cases illustrated on triangular meshes.

2.3 Distribution scheme

The residual vector, R_c, calculated for each cell using equation (6) must be distributed to the computational nodes located at the vertices of the cell using some form of *distribution scheme*. In order to maintain conservation a basic requirement of the distribution scheme is that the sum of the nodal contributions, R_c^i, within a cell must equal the cell residual, *i.e.* the whole cell residual (neither more nor less) must be distributed to the nodes.

Another important property, which is not shared by all schemes, is that of *linearity preservation* or the ability to reproduce linear steady-state solutions exactly (the highest order of polynomial for which this is possible using linear elements). It can be shown that linearity-preserving (LP) system schemes can be written in the form:

$$R_c^i = B_c^i R_c \tag{7}$$

where the B_c^i are referred to as *distribution matrices* which must be bounded and sum to the identity matrix. While linearity preservation does not guarantee second-order accuracy on all grids, in practice it has been found that linear, scalar LP schemes achieve second-order accuracy for sufficiently smooth solutions on general triangular grids[11].

Upwind bias is achieved by distributing the cell residual in the downwind direction, so that these nodes experience more upwind influence, see Figure 2. A summary of various distribution schemes is given for example by Deconinck *et al.*[8,9] and Carette *et al.*[11]. In the present work a form of the Lax-Wendroff scheme has been employed for the pressure-velocity system. This scheme is linear, linearity-preserving and upwind-biased (rather than fully upwind) in nature. The distribution matrix can be written:

$$B_c^i = \frac{1}{N_{vc}} I + \frac{\tau_c}{2V_c} K_c^i \tag{8}$$

in which τ_c is a cell-based time scale defined here as h_c/u_g where u_g is a global velocity scale and h_c is is a cell-based length scale. The first term on the RHS simply distributes the cell residual equally to all nodes while the second term introduces upwind bias into the scheme and importantly introduces a term based on the divergence of the momentum residual into the

continuity equation[10]. This latter term acts to stabilize the system, allowing the use of a collocated variable arrangement for pressure and velocity.

3 SOLUTION PROCEDURE

The basic form of the iteration procedure is as follows:

$$U^{n+1} = U^n + \alpha \Delta U^n \tag{9}$$

in which U is the global solution vector for all computational nodes in the domain and α is a linear-relaxation factor (between zero and one). The goal of the iteration procedure is to drive the global nodal residual vector, $R(U^n)$, towards zero by finding a correction vector ΔU^n which will reduce the residual on each iteration. The correction vector can be found using the simple first-order expression:

$$\left(\frac{\partial R}{\partial U}\right)_n \Delta U^n = -R(U^n) \tag{10}$$

which is a global matrix equation encompassing all solution variables at all nodes. For a single node, i, the i^{th} row of equation (10) can be written:

$$\sum_{j \in \Omega_i} M_{ij}^n \Delta U_j^n = -R(U_i^n) \tag{11}$$

in which each element of the matrix, M_{ij}, is itself a matrix representing the dependence of the residual vector at node i on the solution values at node j:

$$M_{ij}^n = \frac{\partial R_i^n}{\partial U_j^n} \tag{12}$$

and where the compact nature of the discretization means that Ω_i need consist only of the immediate nodal neighbours for a fully-implicit treatment. The Jacobian (12) can either be computed quite exactly, as an analytical or numerical derivative, or in various approximate forms. The final converged solution depends only on the definition of the nodal residuals (determined by the chosen discretization of the equations being solved) while the speed and robustness of the solution procedure depends upon the definition of the M_{ij} matrix elements.

In the present work, two approaches have been adopted: (i) a full Newton method which evaluates the Jacobian by numerical differentiation, as proposed by Issman *et al.*[12]; and (ii) a Picard method which exploits the algebraic form of the nodal residual to obtain an approximate form of the Jacobian, as follows:

$$M_{ij}^n = \sum_{c \in \Omega_i} B_c^i K_c^j \tag{13}$$

in which the dependence of the distribution and inflow matrices on the solution vector is ignored for the purposes of the differentiation.

The main advantages of the Picard method over the full Newton are that the Jacobian entries are significantly cheaper to evaluate and that it is in general more robust (*i.e.* less dependent upon the quality of the initial guess) and therefore more widely applicable. The

main advantage of the Newton method is that in some circumstances it can achieve quadratic convergence behaviour. In the case of laminar flow the Picard and Newton methods have been found to be complementary with the Picard method being used to start the solution before switching to Newton below a certain convergence level. In the case of turbulent flow it has been found that the Newton method was of little interest as storage requirements and the numerical sensitivity of turbulence models preclude its application to the full equation set while its application to the pressure-velocity system alone gives slower convergence rates than the Picard method, as discussed in detail by Waterson[10].

Once constructed, the matrix equation (11) must be solved for the correction vector, ΔU, using some form of linear solver. All of the results presented here have been computed using the ILU(0) preconditioner combined with the GMRES, BiCGStab or TFQMR solvers, as implemented in the Aztec solver library (Version 1.1)[13]. These linear solvers are summarized and compared for example by Saad[14]. The ILU(0) preconditioner uses ILU factorization with zero fill in and requires a single extra copy of the Jacobian matrix to be stored. ILU(0) is sensitive to the matrix structure and solver performance can be improved significantly by node reordering, such as reverse-Cuthill-McKee, to minimize the bandwidth[10]. As the solvers vary in speed, stability and the amount of storage required it has been found useful to have a choice of methods available.

The convergence level for the linear solver is specified via the relative reduction in the L_2-norm of the linear residual vector. If the stopping criterion is set too low then unnecessary computational time is expended to no effect, whereas if it is set too high convergence of the non-linear solution may stall or be slowed. In the present work values in the range 0.5e-2 to 1.0e-2 have been found to be optimal for Picard iterations and 0.5e-4 to 1.0e-4 for Newton iterations (to achieve quadratic convergence behaviour).

It has been found in practice that little or no solution relaxation is required for most of the cases considered. For turbulent flow cases, in which the mean flow and turbulence equations are segregated, linear relaxation of α=0.8, equation (9), was found necessary. A number of further detailed issues relating to the solution procedure are discussed by Waterson[10], including parallelization, solution segregation and domain decomposition.

4 RESULTS

The present method has been extensively validated and a wide range of cases, both laminar and turbulent, with and without heat transfer, are considered by Waterson[10]. Results are presented here for three test cases, all laminar and isothermal, chosen to illustrate specific aspects of the performance of the method and comparisons with other techniques. The cases considered are a shear-driven cavity (2D), a bifurcation (2D) and a 90º pipe bend (3D).

4.1 Shear-driven cavity

The shear-driven cavity is considered here as a standard test case for the evaluation and comparison of discretization methods for the incompressible Navier-Stokes equations. The reference solution chosen is that of Ghia *et al.*[15]. A number of studies have used this case to compare various cell-centred finite-volume convection schemes on Cartesian quadrilateral

meshes[16,17,18,19,20] and de Mulder[21] applied the SUPG/PSPG finite-element method to this case on an irregular triangular mesh, which is also used here.

The geometry consists of a unit square with one side, the lid, moving at a fixed velocity which drives a recirculating flow. The Reynolds number (based on the cavity width and lid velocity) can be varied to create different flow regimes. Three Reynolds numbers are considered here: 100, 1000 and 10000, covering the full range of results presented Ghia *et al.*[15]. While at the two lower Reynolds numbers the two-dimensional flow is quite stable, at Re=10000 the flow is rather unstable and a steady two-dimensional flow under these conditions is probably unphysical.

Three levels of uniformly-triangulated Cartesian grids have been used here: *coarse* 35x35 (1225 nodes), *medium* 51x51 (2601 nodes) and *fine* 75x75 (5625 nodes). In addition an irregular (unstructured) triangular grid with 3013 nodes, created by de Mulder[21], has been employed. In each of the Cartesian grids the computational nodes are clustered towards the walls using a symmetric power law with exponent 1.6. The cell aspect ratios vary from 1 to 20. The three grid sizes were chosen so as to achieve grid independence for the Re=1000 case on the medium grid. The Re=10000 case comes close to grid independence only on the fine grid while the Re =100 solution is essentially grid-independent on the coarse grid.

All computations for this case were carried out on a Pentium 200 processor. For Re = 100 and 1000, the BiCGS linear solver was used with a convergence threshold of 1.e-2 for Picard steps and 1.e-4 for Newton steps. For the Re=10000 case, where it was found difficult to achieve convergence, the TFQMR solver was employed, with the same thresholds. It was not necessary to use any form of under-relaxation for Re=100 or 1000 however for Re=10000 linear under-relaxation of 0.8 was required.

Figure 3 shows the velocity vectors and pressure contours for the Re=1000 case on the medium grid. The pressure contours are shown to demonstrate the efficacy of the pressure stabilization in suppressing spurious oscillations. The distributions of horizontal and vertical velocity components, respectively on the vertical and horizontal centre-lines of the cavity, are compared with the reference solution in Figure 4 for Re=100 and 1000. As can be seen, good agreement is achieved already on the medium grid.

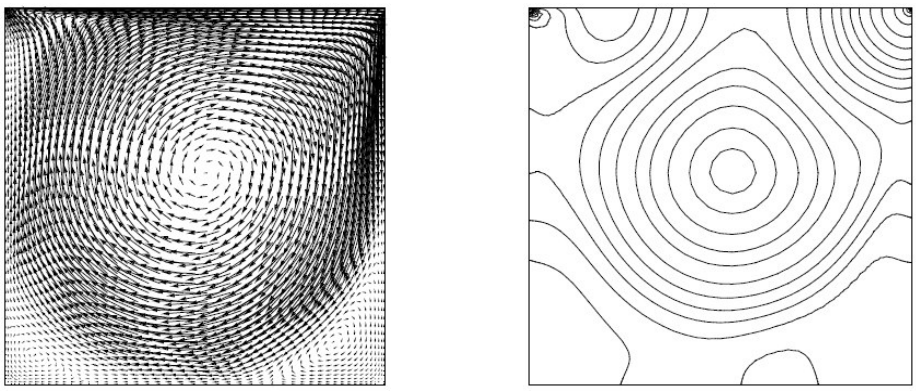


Figure 3: Shear-driven cavity - velocity vectors and pressure contours for Re=1000.

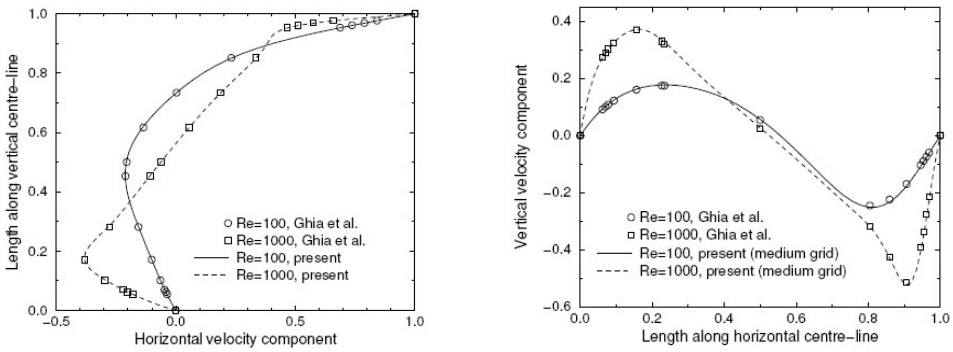


Figure 4: Shear-driven cavity - velocity distributions along horizontal and vertical centre-lines for Re=100 and 1000 on medium grid (comparison with data of [58]).

Convergence histories for all three Reynolds numbers are presented in Figure 5 for the medium grid and summarized in Table 1 for all grid levels and Reynolds numbers. At Re=100 a fully-converged solution can be achieved in 6 Newton steps at all grid levels. At Re=1000 the problem is less stable and the computation must be initiated using the Picard linearization. Newton's method is activated when the global rms pressure residual has reduced by one order of magnitude. A fully converged solution is achieved at all grid levels after 6 Picard and 5 Newton steps. At Re=10000 the flow is much less stable which is reflected in the significantly increased difficulty in achieving a converged solution. Between 9 and 13 Picard steps are required followed by up to 8 Newton steps. For both Re=100 and 1000 the computational time required scales with the number of nodes to the power 1.3, though for the more challenging (and probably unphysical) Re=10000 case the exponent goes up to 2.0.

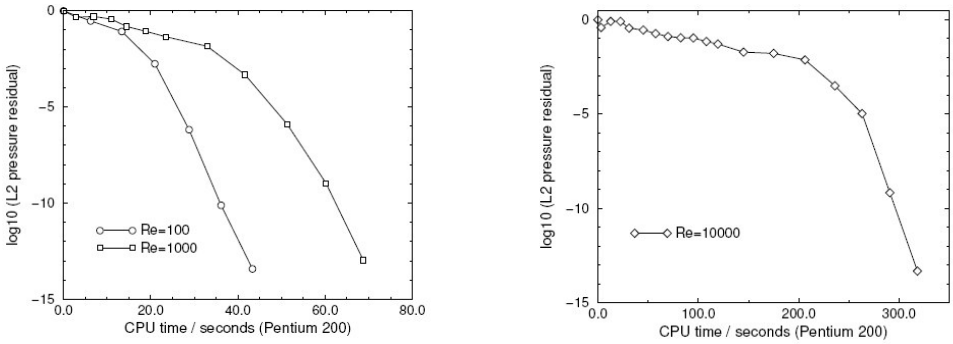


Figure 5: Shear-driven cavity - convergence histories for Re=100, 1000 and 10000 on medium grid (one symbol for each non-linear iteration).

A survey of papers using cell-centred finite-volume convection schemes on quadrilateral meshes for this case[16,17,18,19] reveals that for Re=1000, solutions using standard second-order convection schemes such as QUICK are still grid dependent on a uniform 41x41 Cartesian mesh but are probably grid independent on a uniform 81x81 mesh (though in fact much better results are achieved if the cells are clustered close to the walls). The present authors[20] implemented a number of both linear and non-linear convection schemes within a standard

cell-centred finite-volume code with a staggered grid, comparing the performance of the different schemes for the present case at Re=1000. It was found that the results for the three linear schemes (Fromm's, QUICK and $\kappa=1/3$) were close to grid independence on a 41x41 Cartesian grid with the cells clustered to the walls with a power-law exponent of 1.4 and that they all gave reasonable agreement with the reference solution. Computations with the present method on a similar 41x41 grid show that in fact the solution is only marginally less accurate than that achieved on the present medium grid and is equivalent in accuracy to the results achieved with the three cell-centred linear schemes on quadrilateral meshes. Comparing with the results of de Mulder[21], obtained using the SUPG/PSPG finite-element method on the same unstructured mesh as used here, it is found that the present method gives very similar results which is not surprising as the two approaches have a close relationship.

Grids	**Coarse**	**Medium**	**Unstr.**	**Fine**
Dimension	35x35	51x51	(55x55)	75x75
No. Nodes	1225	2601	3013	5625
Itns: Re=100	6N	6N	6N	6N
Itns: Re=1000	6P+5N	6P+5N	6P+5N	6P+5N
Itns: Re=10000	9P+5N	11P+7N	13P+8N	12P+8N
CPU: Re=100	17.0	43.3	60.8	122.8
CPU: Re=1000	26.3	68.8	91.9	200.0
CPU: Re=10000	64.2	308.2	394.5	1438.0

Table 1: Shear-driven cavity - numbers of non-linear iterations (P:Picard, N:Newton) and CPU time (Pentium 200).

The present method achieves good agreement with the reference solutions of Ghia *et al.*[15] over the full range of Reynolds numbers. The number of non-linear iterations required remains almost invariant with grid size though it is sensitive to the Reynolds number. For the two physical Reynolds numbers the CPU time required scales as the number of nodes to the power 1.3. Using the combined Picard/Newton iteration procedure the present method is able to achieve an accurate solution for the unstable Re=10000 case. The present method is able to achieve similar levels of accuracy on triangular grids as standard second-order cell-centred schemes such as QUICK or Fromm's scheme on quadrilateral meshes.

4.2 Two-level bifurcation

The two-dimensional, laminar flow through a two-level bifurcation, as shown in Figure 6, is well-suited to analysis using unstructured triangular grids. The arrangement considered here is a two-dimensional representation of the experimental set up described by Ramuzat[22] and examined using laser-doppler velocimetry. The case has also been examined numerically by Wilquem[23], using a block-structured, hexahedral, cell-centred finite-volume approach, and by Giraud[24] using the Fluent/UNS (Version 4.1)[25] on unstructured triangular meshes. Comparison is therefore permitted not only with experimental results but also with two alternative numerical methods.

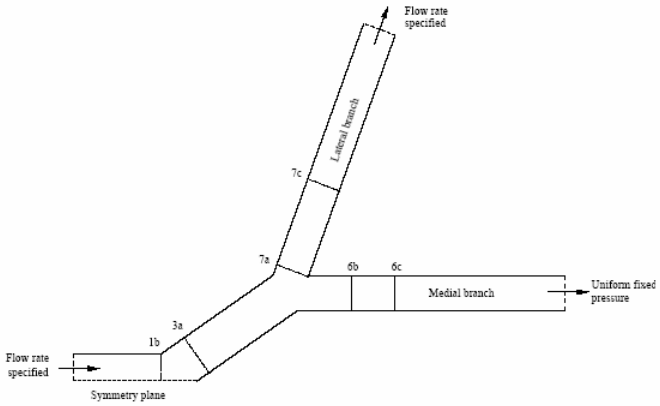


Figure 6: Bifurcation case – schematic diagram and measurement stations

The geometry considered loosely reflects that of a section through a bifurcation in the 5th to 7th generations of a human lung[23]. Such a pulmonary flow is evidently both three-dimensional and unsteady and the validity of a two-dimensional steady representation is discussed by Wilquem[23] and Giraud[24]. The simulation presented here is viewed only as an attempt to recreate the flow on the central plane of the experimental arrangement used by Ramuzat[22] and its biological implications will not be considered.

The computational domain and boundary conditions are shown in Figure 6. The inlet duct undergoes two bifurcations and the presence of a symmetry plane through the first bifurcation means that only half of the experimental geometry need be represented. Two Reynolds numbers were considered: 293 and 410, based upon the inlet-channel bulk velocity and width. A parabolic inlet velocity profile was specified representing fully-developed laminar flow. The outlet conditions were set according to the "forced ventilation" configuration[22,23], which best reflects the experimental conditions. It was assumed that fully-developed flow had been re-established at both outlets, with the division of flow rates between the two fixed by specifying 45% of the flow rate (with a parabolic profile) through the lateral branch and fixing the pressure in the medial branch.

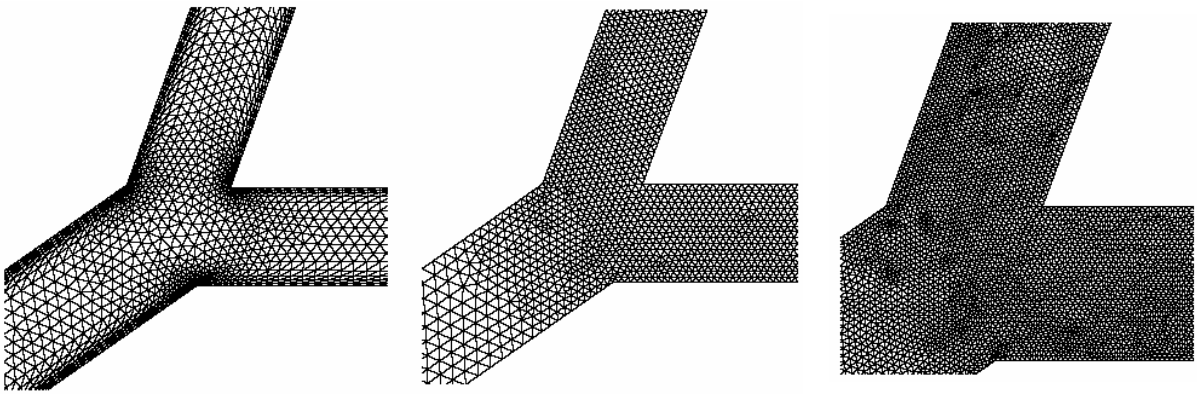


Figure 7: Bifurcation case - detail of non-isotropic coarse grid (3608 nodes) from Daedalus (left), isotropic grid (3358 nodes) from Geomesh (centre) and unsmoothed mesh from Geomesh (right).

Three levels of grid refinement were employed: *coarse* (3608 nodes), *medium* (6063 nodes) and *fine* (14541 nodes), generated using the Daedalus triangular grid-generation program[26]. Thin "viscous" layers of anisotropic cells were generated along all walls, to capture the higher velocity gradients there, while the remaining regions were filled with more isotropic triangles using the frontal Delaunay approach. The resulting cell aspect ratios vary from 1 to 11. The average numbers of nodes across the medial and lateral branches are: 17, 22 and 31 for the coarse, medium and fine grids, respectively.

A detail of the second bifurcation region, showing the viscous and isotropic regions, can be seen in Figure 7. Giraud[24] generated a number of meshes for this case using the Fluent-Uns/Geomesh[25] grid generator. A detail of one of these grids, of a similar size to the coarse mesh, is also shown in Figure 7, where it can be seen that the size of the grid cells does not vary across the duct, though it does vary somewhat in the streamwise direction. The average number of nodes across the lateral and medial branches is the same as for the coarse grid though the distribution is significantly different. An unsmoothed grid, also produced with Geomesh, and used to examine the treatment of distorted grids is also shown in Figure 7.

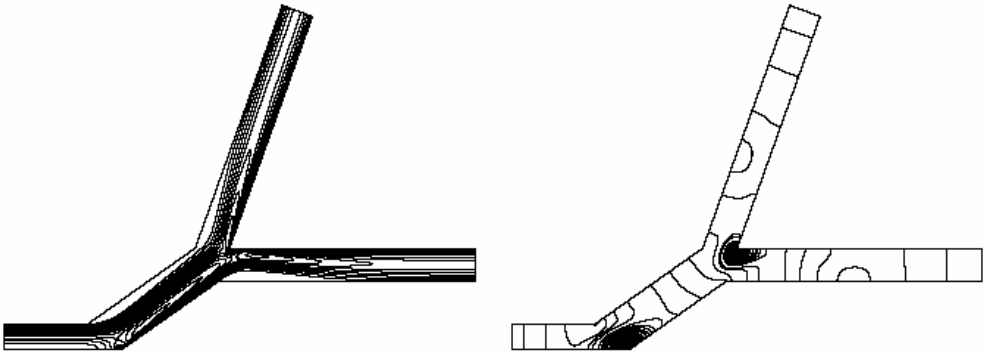


Figure 8: Bifurcation case, Re=410 - contours of velocity magnitude (left) and pressure (right).

The overall flow fields for the higher Reynolds number are illustrated in Figure 8, showing contours of static pressure and velocity magnitude. The straight pressure contours in the inlet passage and close to the outlets of the lateral and medial branches reveal the presence of Poiseuille-type fully-developed laminar flow in those regions. This flow is however heavily distorted in the vicinity of the two bifurcations and strong pressure concentrations can be seen around each of the sharp cusps. Examination of the contours of velocity magnitude in Figure 8 shows that the two-dimensional simulations predict flow separation at each of the sharp convex corners for both cases while the experimental data showed full separation only in the medial branch and at the higher Re value. As the predicted behaviour seems quite reasonable for such a two-dimensional flow, it is expected that this difference is mainly related to a lack of two-dimensionality in the experiments, a consideration addressed by Giraud[24].

Figure 9 shows pressure contours around the second bifurcation for Re=410 computed using the present method and Fluent/UNS on the unsmoothed mesh. It is clear that for the present method the difference in grid quality makes little difference to the quality of the solution (compare with Figure 8) however in the case of Fluent/UNS the reduced grid quality produces noticeable spatial oscillations in the pressure field, not present on the smoothed grid.

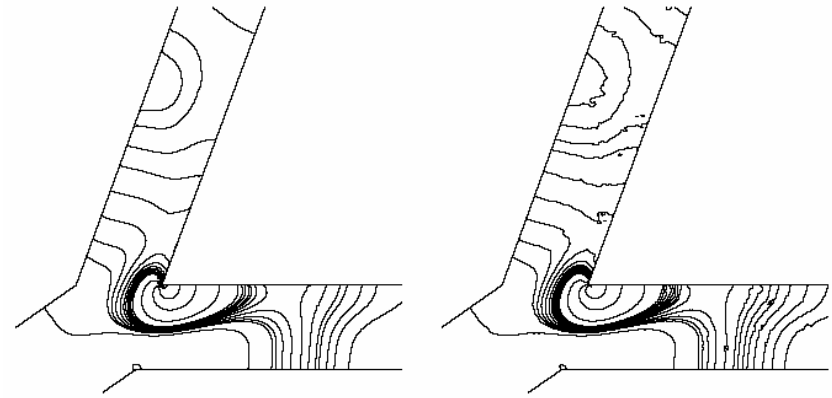


Figure 9: Bifurcation case, Re=410 – comparison of pressure contours on distorted grid: present method (left) and Fluent-Uns (right).

Figure 10 shows comparisons of the predicted axial velocity profiles and experimental data at a series of locations for Re=410 (locations shown in Figure 6). Good agreement is achieved with the experiments at the first two stations and reasonable agreement at the first stations in the medial and lateral branches. Further downstream in the medial branch, at station 6c, the agreement deteriorates somewhat at both Reynolds numbers. These differences may largely be attributed to the three-dimensionality of the experimental set up. Wilquem[23] obtained better agreement by imposing the experimental velocity profile at station 6c.

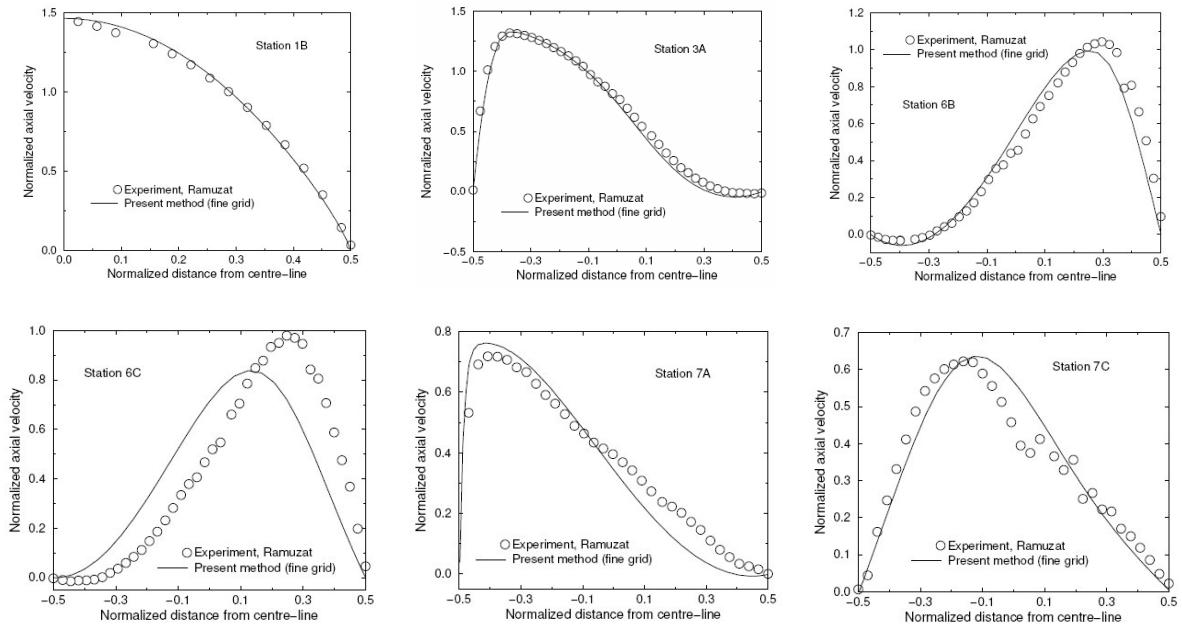


Figure 10: Bifurcation case, Re=410 – comparison of predicted (2D) and experimental axial velocities at six stations. See Figure 6 for locations.

4.3 Pipe bend

A useful three-dimensional test case is the laminar flow in a 90° pipe bend of circular cross section. Experimental results for such a flow are given by Enayet *et al.*[27] for a Reynolds number of 500, based on the bulk velocity and pipe diameter, *d*. Despite the relative simplicity of the geometry, which can be seen in Figure 11, the flow is fully three-dimensional in nature with a pair of counter-rotating vortices being generated as the flow negotiates the bend. These arise as the pressure gradient necessary to turn the high-speed flow in the centre of the pipe imposes itself upon the slow moving fluid close to walls.

This case has been used here both as a basic validation of the present method against experimental data and to allow comparisons with two well-established commercial solvers: CFX-4.3[28] and CFX-5.3[29] from Ansys Inc. These solvers each use very different discretization approaches and solution procedures making the comparisons particularly interesting. Computational results have also been given by Tamamidis and Assanis[30] using a collocated, cell-centred finite-volume approach on a structured hexahedral mesh.

The CFX-4.3 code uses a standard hexahedral multiblock-structured, cell-centred finite-volume discretization with a collocated variable arrangement and Rhie-Chow stabilization. The SIMPLEC solution procedure is used and for the computations carried out here the scalar algebraic multigrid matrix solver was employed with the QUICK scheme for the convection discretization. Linear under-relaxation of 0.95 was used for the momentum equations with no under-relaxation on the pressure. The CFX-5.3 code uses a control-volume finite-element method on unstructured hybrid meshes[31]. There is a collocated variable arrangement with a version of the Rhie-Chow stabilization approach. For all solutions discussed here the fully second-order convection discretization was employed. The solution procedure is a coupled implicit Picard method with an algebraic multigrid matrix solver[32]. For all of the CFX-5.3 computations the CFL number was set to 10^6 (in effect solving the steady equations with no form of relaxation).

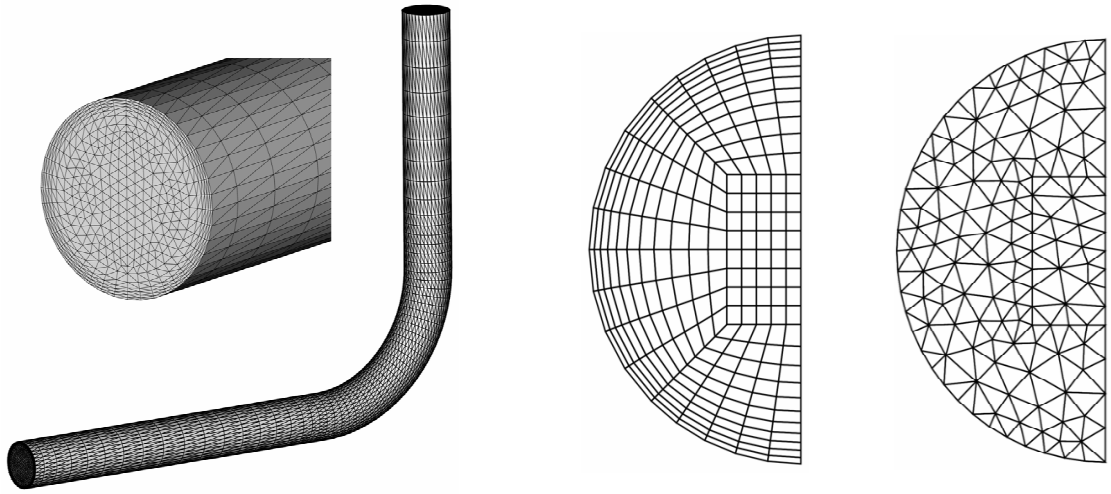


Figure 11: Pipe bend meshes - extruded tetrahedral (left), hexahedral (centre) and isotropic tetrahedral (right).

As the flow is symmetrical in nature, the computational domain represents only half of the pipe with a symmetry boundary condition on the centre plane (though in some of the figures this is reflected for clarity). Examination of the experimental data just upstream of the bend reveals that the flow is not fully developed. For the present study a uniform velocity profile was imposed at the inlet which was located 7.73*d* upstream of the bend, giving the correct axial velocity profile at the experimental station 0.58*d* upstream of the bend. A fixed uniform static-pressure boundary condition was applied 11*d* downstream of the bend.

Three different series of meshes have been generated for the present study, illustrated in Figure 11:

- *Hexahedral*: consisting of two structured blocks – a C-mesh block conforming to the pipe wall enclosing an H-mesh block in the centre. These meshes were used for both CC-FV and CV-FEM computations, using CFX-4.4 and CFX-5.3 respectively.
- *Extruded tetrahedral*: with thin "viscous" layers close to the walls and all tetrahedra stretched in the axial direction, created by extruding a 2D triangular mesh from the inlet plane. These meshes were used to generate the reference solutions using the present method. A overall view and detail of the coarse extruded mesh are shown in Figure 11.
- *Isotropic tetrahedral*: with a relatively uniform distribution of computational nodes (though not strictly isotropic). These meshes were used to allow a direct comparison between the present method and CFX-5.3 on the same tetrahedral mesh (for practical reasons this was not possible using the extruded meshes).

Table 2 gives the grid dimensions for each of the three mesh types. The extruded tetrahedral and hexahedral meshes are made up of a series of transverse planes allowing a good control of the transverse and axial grid distributions. Isotropic tetrahedral meshes are not well suited to this type of geometry (with the axial dimension much greater than the transverse). The flow solution is much more sensitive to the transverse than the axial distribution however for the isotropic tetrahedral mesh the same refinement must be used in all directions. For this reason, for a given total number of computational nodes, the number of nodes in any transverse plane is much lower for the isotropic mesh than for the other two. For the fine extruded mesh the cell aspect ratios vary from 3.8 to 476 while those of the fine isotropic mesh vary between 1 and 6.7.

Mesh type	**Coarse**	**Medium**	**Fine**
Tetrahedral (extr.)	14040 (60x234)	25256 (44x574)	-
Tetrahedral (iso.)	14599 (~281x52)	32650 (~344x95)	66043 (~443x149)
Hexahedral	14280 (60x238)	32640 (60x544)	65280 (120x544)

Table 2: Pipe bend case - numbers of computational nodes in each mesh (with no. of transverse planes x no. of nodes in each plane).

For the present method all solutions were obtained without any form of relaxation and using either the Picard or Picard/Newton linearizations. The ILU(0)/BiCGS matrix solver was employed with linear threshold of 5.e-2 for Picard iterations and 5.e-4 for Newton iterations. Two convergence histories are shown in Figure 14. In order to facilitate direct comparison

with the CFX codes which both use single-precision arithmetic, the present code was recompiled in single precision for the comparison computations.

Experimental data are available for axial velocity on five transverse planes along the pipe. Figure 12 shows predicted axial velocity vectors on the pipe symmetry plane along with transverse velocity vectors in two of the experimental planes (75º around the bend and 1*d* downstream), computed using the present method on the medium extruded tetrahedral mesh.

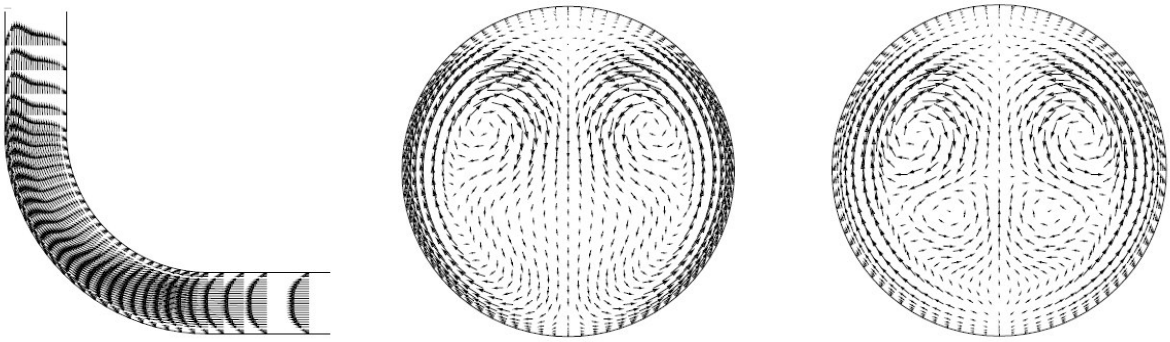


Figure 12: Velocity vectors on symmetry plane (left) and on transverse planes 75º around bend (centre) and 1*d* downstream (right)

As can be seen in Figure 12, as the flow moves around the bend it gradually loses its axial symmetry, with a faster jet of fluid forming closer to the outer wall. Downstream of the bend it gradually returns to the standard axisymmetric pipe flow. The loss of axial symmetry is associated with the development of secondary vortices – initially a single pair but then breaking down into two pairs further downstream.

Figure 13 shows distributions of axial velocity on the centre-lines of two of the transverse experimental planes (30° and 75° around the bend). Results are presented using the present method on the medium extruded tetrahedral mesh and using CFX-4.3 on the medium hexahedral mesh. It can be seen that on both of the planes the two numerical solutions are almost indistinguishable and that good agreement is obtained with the experimental results. It should be noted also that the numerical results are very similar to the CFX-5.3 results on the hexahedral mesh and to those of Tamamidis and Assanis[30].

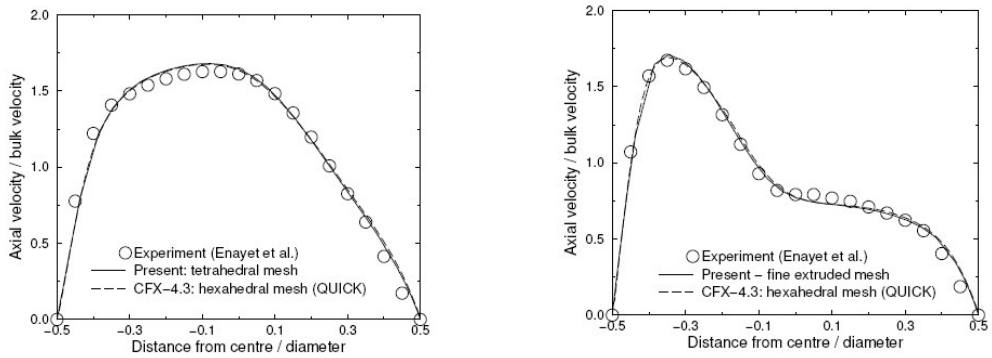


Figure 13: Axial-velocity distribution on symmetry plane 30º (left) and 75º (right) around bend

Further downstream (1*d* from the exit of the bend) there is a discrepancy between the experimental data and the numerical predictions in the centre of the pipe, though good agreement is still obtained in the fast-moving outer flow. This difference is evident for all of the discretizations considered here: the present method, CFX-4.3, CFX-5.3 and the results of Tamamidis and Assanis[30].

Performance comparisons have been carried out of the present method on the isotropic tetrahedral meshes with CFX-4.3 and CFX-5.3 on the hexahedral meshes and with CFX-5.3 on the same isotropic tetrahedral meshes. Results for the finest level considered are given in Table 3 in terms of the storage requirements, number of iterations and computational time. In order to compare like with like the convergence criterion is taken to be a 4.5 order-of-magnitude reduction in the rms (L_2-norm) pressure residual. All computations use identical boundary conditions and initial fields.

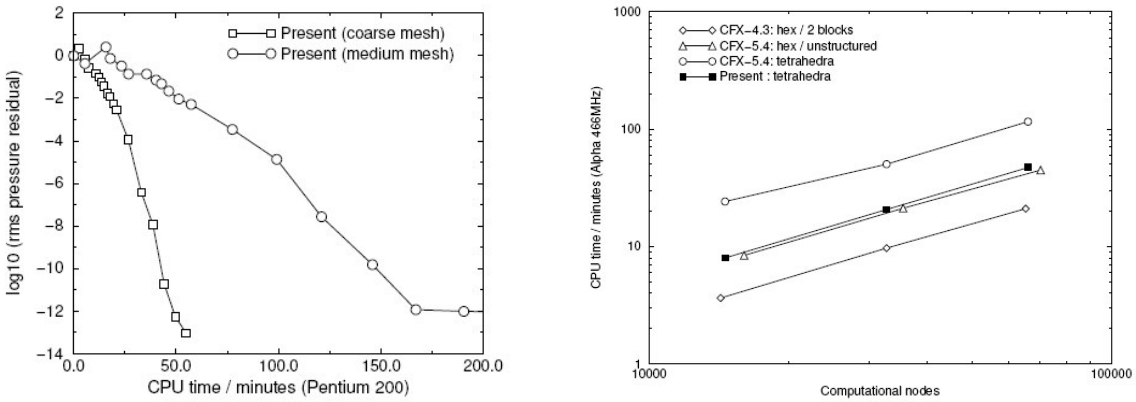


Figure 14: Convergence histories on extruded tetrahedral mesh (left) and CPU time scaling with mesh size for CFX 4.3, 5.4 and present method (right)

It can be seen from Table 3 that though the structured multiblock solution from CFX-4.3 uses 200 iterations to achieve the convergence criterion, it does so with less than half of the CPU time and storage requirements of any of the other methods. It should be noted however that the present case is well suited both to the SIMPLEC solution procedure and a structured grid and might be sensitive for example to the number and arrangement of the blocks. The CFX-5.3 solution on the same mesh, with a coupled unstructured treatment, uses fewer iterations but takes twice as long and uses twice as much storage.

Code	Nodes	Cells	Storage/Mb	Iterations	CPU/mins
CFX-4.3 (hex)	65280	65280	45	200	21.0
CFX-5.3 (hex)		65280	90	32	44.8
CFX-5.3 (tet)	66043	330215	181	52	115.5
Present (tet)	66043	330215	154	18	50.2

Table 3: Pipe bend case - comparative performance figures on fine hexahedral and isotropic tetrahedral meshes (CPU time on 500MHz DEC Alpha).

Unsurprisingly both solutions on the isotropic tetrahedral mesh require more CPU time and more storage than for the hexahedral solutions, however the present method requires less than one half of the computational time of CFX-5.3 and 15% less storage. Also of interest is the scaling of the computational time with the number of computational nodes which is shown in Figure 14 for all four approaches. It can be seen that for the problem and range of meshes considered the scaling is quite similar for all approaches, with exponents of between 1.1 and 1.2, *i.e.* close to linear.

5 CONCLUSIONS

A method has been presented for the solution of the governing equations of low-Mach-number and incompressible flow using the residual-distribution method. The approach combines a system residual-distribution scheme, which uses multidimensional upwinding to achieve an accurate discretization on a very compact stencil, with a fully-coupled implicit solution procedure. This combination has been demonstrated to be effective in producing accurate solutions on triangular and tetrahedral meshes of variable quality with little or no solution relaxation. The method has been compared with other methods, including commercial flow solvers, with favourable results in terms of both computational time and lack of sensitivity to grid quality.

REFERENCES

1. S.V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Corporation, (1980).
2. R.-H. Ni, "A multiple-grid scheme for solving the Euler equations", *AIAA J.*, Vol. 20, 11:1565-1571, (1982).
3. J.D. Denton, "An improved time marching method for turbomachinery flow calculation", *ASME Journal of Engineering for Power*, 105:514-524, Academic Press, (1983).
4. M.G. Hall, "Cell-vertex multigrid schemes for solution of the Euler equations", In *Proceedings of the Conference on Numerical Methods for Fluid Dynamics*, Reading, K.W. Morton and M.J. Baines, eds., 303-345, Oxford University Press, (1985).
5. K.W. Morton, P.I. Crumpton and J.A. Mackenzie, "Cell vertex methods for inviscid and viscous flows", *J. Computers and Fluids*, Vol. 22, 91-102, (1993).
6. M.B. Giles, W.K. Anderson and T.W. Roberts, "Upwind control volumes: a new upwind approach", *AIAA Paper* 90-0104 (1990).
7. P.L. Roe, "Discrete models for the numerical analysis of time-dependent multidimensional gas dynamics", *J. Computational Physics*, 63:458-476, (1986).
8. H. Deconinck, R. Struijs, G. Bourgois and P.L. Roe, "Compact advection schemes on unstructured grids", in *Lecture Series LS-1993-04: Computational Fluid Dynamics*, von Karman Institute, Sint-Genesius-Rode, Belgium, (1993).
9. H. Deconinck and G. Degrez, "Multidimensional upwind residual distribution schemes and applications", *2nd International Symposium on Finite Volumes for Complex Applications*, Duisburg, Germany, (1999).

10. N. P. Waterson, *Simulation of turbulent flow, heat and mass transfer using a residual-distribution approach*, Doctoral Thesis, Delft University of Technology, published by von Karman Institute for Fluid Dynamics, Sint-Genesius-Rode, Belgium, (2003).
11. J.C. Carette, H. Deconinck, H. Paillère and P.L. Roe, "Multidimensional upwinding: its relation to finite elements", *Int. J. Numerical Methods in Fluids*, Vol. 20, 935-955, (1995).
12. E. Issman, G. Degrez and H. Deconinck. "Implicit upwind residual distribution Navier-Stokes solver on unstructured meshes", *AIAA J.*, Vol. 34, 10:2021-2028, (1996).
13. S.A. Hutchinson, J.N. Shadid, and R.S. Tuminaro, *Aztec User's Guide: Version 1.1*, Technical Report SAND95-1559, Sandia National Laboratories, U.S.A., (1995).
14. Y. Saad, "Krylov subspace techniques, conjugate gradients, preconditioning and sparse matrix solvers", in *Lecture Series LS-1994-05: Computational Fluid Dynamics*, von Karman Institute, Sint-Genesius-Rode, Belgium, (1994).
15. U. Ghia, K.N. Ghia and C.T. Shin, "High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method", *J. Computational Physics*, 48:387-411, (1982).
16. T. Han, J.A.C Humphrey and B.E. Launder, "A comparison of hybrid and quadratic-upstream differencing in high Reynolds number elliptic flows", *Computer Methods in Applied Mechanics and Engineering*, 29:81-95, (1981).
17. T. Hayase, J.A.C. Humphrey and R. Greif, "A consistently formulated QUICK scheme for fast and stable convergence using finite-volume iterative calculation procedures", *J. Computational Physics*, 98:108-118, (1992).
18. P.G. Huang, B.E. Launder and M.A. Leschziner, "Discretization of nonlinear convection processes: a broad-range comparison of four schemes", *Computer Methods in Applied Mechanics and Engineering*, 48:1-24, (1985).
19. S. Thakur and W. Shyy, "Some implementational issues of convection schemes for finite-volume formulations", *Numerical Heat Transfer*, Part B, 24:31-55, (1993).
20. N.P. Waterson, *Development of a bounded higer-order convection scheme for general industrial applications*, Project Report 1994-33, von Karman Institute, Sint-Genesius-Rode, Belgium, (1994).
21. T. De Mulder, *Stabilized Finite Element Methods for Turbulent Incompressible Single-Phase and Dispersed Two-Phase Flows*, PhD thesis, Katholieke Universiteit Leuven, Belgium, (1997).
22. A.Ramuzat, *Experimental investigation of flows within two-dimensional bifurcations*, Project Report 1996-22, von Karman Institute, Sint-Genesius-Rode, Belgium, (1996).
23. F. Wilquem, *A high resolution multiblock strategy for the resolution of the incompressible Navier-Stokes equations. (Applications to physiology.)*, PhD thesis, Faculté Polytechnique de Mons, Belgium, (1997).
24. D. Giraud, *Numerical investigation of air flows in multiple lung bifurcations*, Project Report 1998-10, von Karman Institute, Sint-Genesius-Rode, Belgium, (1998).
25. *Fluent/UNS*, Version 4.1, Fluent Inc., www.fluent.com.
26. J. C. Carette, *Adaptive Unstructured Mesh Algorithms and SUPG Finite Element Method for Compressible High Reynolds Number Flows*, PhD thesis, Université Libre de Bruxelles, Belgium, (1997).

27. M.M. Enayet, M.M. Gibson, A.M.K.P. Taylor and M. Yianneskis, *Laser Doppler Measurements of Laminar and Turbulent Flow in a Pipe Bend*, NASA Contract Report CR-3551, (1982).
28. *CFX, Version 4.3*, Ansys Inc. (formerly owned by AEA Technology PLC), (2001).
29. *CFX, Version 5.4*, Ansys Inc. (formerly owned by AEA Technology PLC), (2001).
30. P. Tamamidis and D.N. Assanis, "Three-dimensional Incompressible Flow Calculations with Alternative Discretization Schemes", *Numerical Heat Transfer*, Part B, 24, 57-76, (1993).
31. G.E. Schneider and M.J. Raw, "A skewed, positive influence coefficient upwinding procedure for control-volume-based finite-element convection-diffusion computation", *Numerical Heat Transfer*, 9, 1-26, (1986).
32. M.J. Raw, "A coupled algebraic multigrid method for the 3D Navier-Stokes equations. Fast Solvers for Flow Problems", in *Proceedings of the 10th GAMM Seminar*, Kiel, W. Hackbusch and G. Wittum eds., Notes on Numerical Mathematics, Vol. 49, 204-215, Vieweg, (1994).