INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 25, 371–392 (1997)

THE THREE-DIMENSIONAL PROLONGED ADAPTIVE UNSTRUCTURED FINITE ELEMENT MULTIGRID METHOD FOR THE NAVIER–STOKES EQUATIONS

S. Ø. WILLE*

Faculty of Engineering, Oslo College, Cort Adelersgate 30, N-0254 Oslo, Norway

SUMMARY

This paper presents the development of the three-dimensional prolonged adaptive finite element equation solver for the Navier–Stokes equations. The finite element used is the tetrahedron with quadratic approximation of the velocities and linear approximation of the pressure. The equation system is formulated in the basic variables. The grid is adapted to the solution by the element Reynolds number. An element in the grid is refined when the Reynolds number of the element exceeds a preset limit. The global Reynolds number in the investigation is increased by scaling the solution for a lower Reynolds number. The grid is refined according to the scaled solution and the prolonged solution for the lower Reynolds number constitutes the start vector for the higher Reynolds number. Since the Reynolds number is the ratio of convection to diffusion, the grid refinements act as linearization and symmetrization of the equation system. The linear equation system of the Newton formulation is solved by CGSTAB with coupled node fill-in preconditioner. The test problem considered is the three-dimensional driven cavity flow. © 1997 by John Wiley & Sons, Ltd.

Int. J. Numer. Meth. Fluids, **25**: 371–392 (1997).

No. of Figures: 17. No. of Tables: 6. No. of References: 17.

KEY WORDS: adaptive grids; finite element method; iterative solvers

1. INTRODUCTION

The tri-tree grid generation algorithm has previously been developed and described by Wille.[1] This grid generator composes a structured irregular grid. Later the grid generation algorithm was coupled with a two-dimensional grid adaption algorithm for solving the Navier–Stokes equations.[2] The effects of adaptive linearization and iterations with respect to grid adaption were studied in Reference 3. The finite element equations from the Navier–Stokes equations are integrated analytically and the analytical integration formulae can be found in Reference 4.

The development of iterative linear equation solvers for non-symmetric equation systems[5,6] made it possible to solve such systems without storing the full factorized equation matrix. The corresponding reduction in storage requirements meant that larger problems could be examined more accurately.[7,8] A new incomplete coupled node fill-in preconditioner[8] speeded up the convergence rate of the iterative equation solver CGSTAB[6] considerably.

* Correspondence to: S. Ø. Wille, Faculty of Engineering, Oslo College, Cort Adelersgate 30, N-0254 Oslo, Norway.

Contract grant sponsor: Norwegian Research Council
Contract grant number: NN2461 K

CCC 0271–2091/97/040371–22 $17.50 *Received October 1996*
© 1997 by John Wiley & Sons, Ltd.

In the grid adaption or refinement–recoarsement algorithm the element Reynolds number is used as refinement–recoarsement indicator.[9] This indicator tends to symmetrize and linearize the equation system. However, other refinement–recoarsement indicators have been investigated. Kallinderis[10] monitors the values of velocity differences and gradients across cell edges in order to divide/delete grid cells for invicid transonic flow. The grid generation method uses an oct-tree as superior structure and the grid is constructed from triangles in 2D and tetrahedra and prisms in 3D.

A posteriori error estimation of the solution has been used as refinement criterion for the linear Stokes equations.[11,12] The rules for two-dimensional triangular divisions are similar to those used in References 1 and 10.

Zienkiewicz and Zhu[13] demonstrated that finite element grid refinements reveal advantageous properties for both stabilizing the Navier–Stokes differential equation system and resolving the solution of the equations at locations of high convection and gradients. The refinement indicator used by Zienkiewicz and Zhu[13] was also *a posteriori* error estimation.

The tri-tree grid generation method has also been explored[14] together with the traditional multigrid method.[15,16]

The present investigation of the prolonged adaptive multigrid method is extended and performed on three-dimensional cavity flow.

2. TRI-TREE GRID GENERATION

The data structure for the tri-tree in three dimensions is based on tetrahedra. During refinements, each tetrahedron is divided into eight new tetrahedra. Successive divisions are organized as a tree structure as shown in Figures 1 and 2.

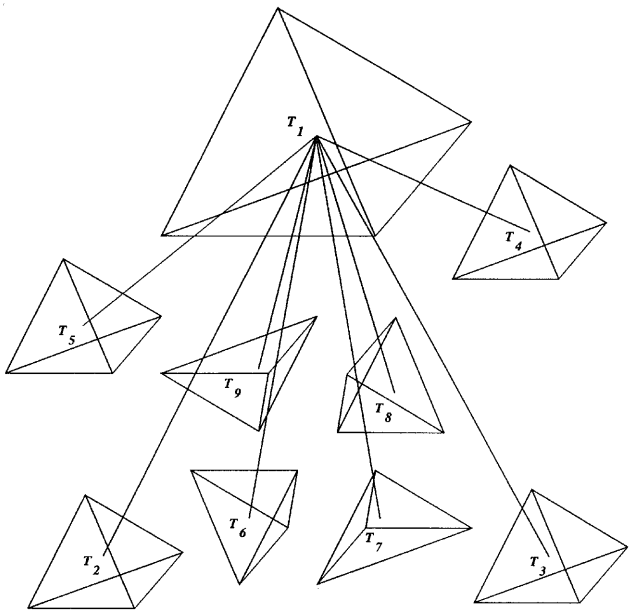


Figure 1. Tri-tree structure for subsequent divisions

 © 1997 by John Wiley & Sons, Ltd.

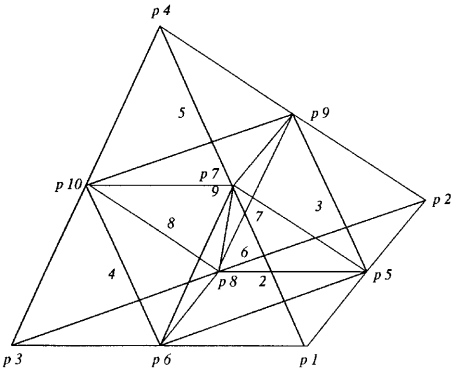


Figure 2. Initial equilateral tetrahedron consisting of corners $T_1 = P_1, P_2, P_3, P_4\}$ which is divided into eight new tetrahedra

The basic tetrahedron, which consists of the corners $T_1 = \{P_1, P_2, P_3, P_4\}$, is divided into eight tetrahedra with the corners

$$\begin{aligned} T_2 &= \{P_1, P_5, P_6, P_7\}, & T_3 &= \{P_5, P_2, P_8, P_9\}, \\ T_4 &= \{P_6, P_8, P_3, P_{10}\}, & T_5 &= \{P_7, P_9, P_{10}, P_4\}, \\ T_6 &= \{P_8, P_5, P_6, P_7\}, & T_7 &= \{P_5, P_7, P_8, P_9\}, \\ T_8 &= \{P_6, P_8, P_7, P_{10}\}, & T_9 &= \{P_7, P_9, P_{10}, P_8\}. \end{aligned} \tag{1}$$

The tree structure requires 14 integers in three dimensions in order to keep the necessary information at each level of subdivision (Figure 3). The first integer in the tri-tree record contains the level of the division. The initial tetrahedron is at level 1. As a division occurs, the level number is increased by one. The level number then indicates the size of the division and all tetrahedra of equal

1	1	p1	p2	p3	p4	2	3	4	5	6	7	8	9	0
2	-2	p1	p5	p6	p7	6	0	0	0					1
3	-2	p5	p2	p8	p9	0	7	0	0					1
4	-2	p6	p8	p3	p10	0	0	8	0					1
5	-2	p7	p9	p10	p4	0	0	0	9					1
6	-2	p8	p5	p6	p7	2	8	7	0					1
7	-2	p5	p7	p8	p9	9	3	0	6					1
8	-2	p6	p8	p7	p10	9	0	4	6					1
9	-2	p7	p9	p10	p8	0	8	7	5					1

Figure 3. Information in tri-tree structure in three dimensions. The record length of information for each division consists of 14 integers. The first integer describes the level of refinement. The next four integers are the indices to the corners of the present tetrahedron. If the refinement level integer is positive, the next eight integers point to the tetrahedra into which the tetrahedron is refined. If the tetrahedron is terminal, the subsequent three integers are pointers to the neighbour tetrahedra. The last index points to the parent tetrahedron

© 1997 by John Wiley & Sons, Ltd.

size will have the same level number. When a division is terminal, the level number is given a negative sign. In addition to the level number, a point index to each of the corners of the structure is stored. This is not strictly necessary, because the co-ordinates of each point can be calculated when they are needed. However, if the corner points are stored, the computing time will be considerably reduced. The next positions in the structure record are pointers to the records of the divisions.

When a tetrahedron is terminal, some of these pointers are used as pointers to the neighbour tetrahedra instead. The last integer in the record points to the record of the parent tetrahedron. It is therefore possible to perform both up and down searches in the tri-tree.

When searching for a tetrahedron which inscribes a given point, the tri-tree is traversed towards the root if the point is outside the present tetrahedron and towards the leaves if it is inside. Let the co-ordinates of the point be $\{x, y, z\}$, the area co-ordinate of the tetrahedron L_k and the co-ordinates of the nodes x_k, y_k, z_k for $k = 1, 2, 3, 4$. A simple test to exclude the tetrahedra which do not contain the point is to decide whether the point is outside the circumscribing box. If one or more of the conditions are valid, the point is outside the box if

$$\left(\max_k x_k < x\right), \quad k = 1, 2, 3, 4, \tag{2}$$

$$\left(\min_k x_k > x\right), \quad k = 1, 2, 3, 4, \tag{3}$$

$$\left(\max_k y_k < y\right), \quad k = 1, 2, 3, 4, \tag{4}$$

$$\left(\min_k y_k > y\right), \quad k = 1, 2, 3, 4, \tag{5}$$

$$\left(\max_k z_k < z\right), \quad k = 1, 2, 3, 4, \tag{6}$$

$$\left(\min_k z_k > z\right), \quad k = 1, 2, 3, 4. \tag{7}$$

If the point is inside the box, the following equation system has to be solved to determine whether the point is inside the tetrahedron:

$$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \tag{8}$$

If $0 \leqslant L_k \leqslant 1$ for $k = 1, 2, 3, 4$, then the point is inside the tetrahedron.

When a tetrahedron is divided, the midpoint on each line between the corners is calculated. This point may already exist if the neighbour has a larger level number. If a point does not already exist, it is added to the list of points. In order to be able to search for and add points quickly, the list is organized as a binary tree (Figure 4).

The binary tree is sorted lexically on the point co-ordinates x, y, z. A point with co-ordinates x, y, z as defined to be less than a point with co-ordinates u, v, w is defined above the binary tree in Figure 4.

From Figure 5 it is seen that the tetrahedra contains more than one node at their sides. However, this node distribution will lead to very narrow tetrahedra, which will be disadvantageous when

 © 1997 by John Wiley & Sons, Ltd.

$$P = [x, y, z] < Q = [u, v, w]$$

$$\begin{array}{llll} x < u & or & & \\ x = u & and & y < v & or \\ x = u & y = v & and & z < w \end{array}$$

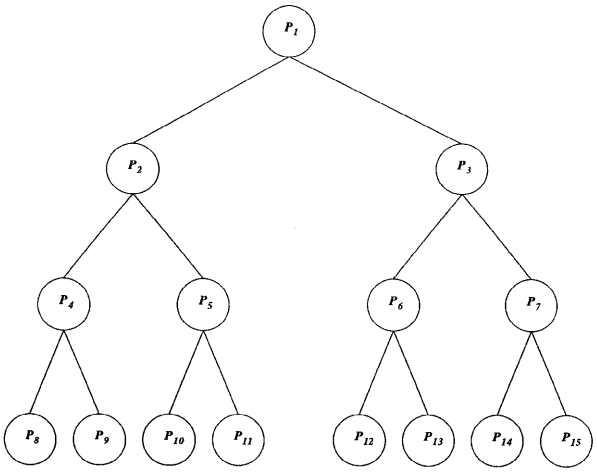


Figure 4. Binary search tree for points

integrating the differential equations over the elements. The triangulation in Figure 5 is therefore balanced so that no triangle has more than one excess node on one of its sides. Figure 6 shows the result of the balancing procedure.

After balancing the tri-tree, the tri-tree structure is triangulated into finite elements as shown in Figure 7.

Figures 8 and 9 show the two allowed ways of subdividing the tetrahedra into finite elements. The tetrahedron is divided into two tetrahedral finite elements as shown in Figure 8.

In Figure 9 the tetrahedron is divided into four tetrahedral finite elements as shown. These subdivisions show the finite elements which deviate most from the equilaterial shape.

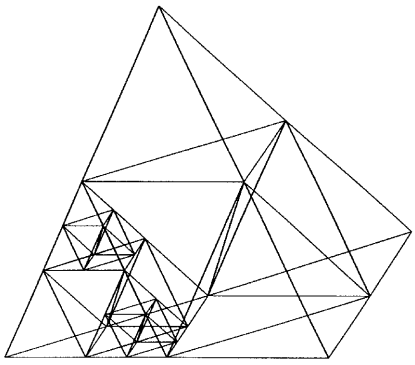


Figure 5. Division of tetrahedron until desired degree of refinement is obtained. The triangulation shown is arbitrary and it can be seen that the tetrahedron has several nodes at one of its edges. A triangulation of this structure will result in very narrow triangles

© 1997 by John Wiley & Sons, Ltd.

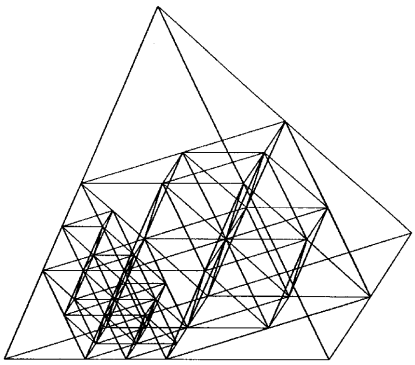


Figure 6. Balanced triangular structure of triangulation in Figure 5

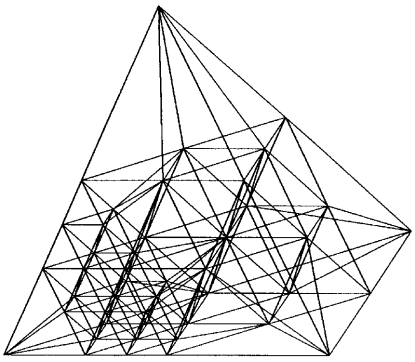


Figure 7. Triangulated finite element grid from tri-tree structure in Figure 6

The tri-tree balancing procedure is defined by the following algorithm. Let L_p be the level number of the tetrahedron p and let L_n be the level number of its neighbours n. Let n_{l} be the number of neighbours with level number less than and n_{g} be the number of neighbours with level number greater than the level number of tree element p respectively. The refinement algorithm is then given as follows.

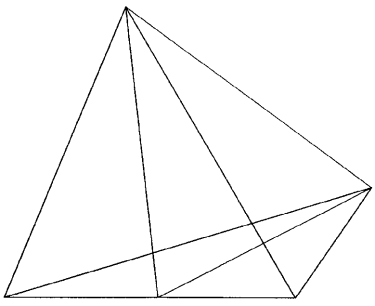


Figure 8. Equilateral tetrahedron divided into two finite elements, depending on division of neighbour elements

 © 1997 by John Wiley & Sons, Ltd.

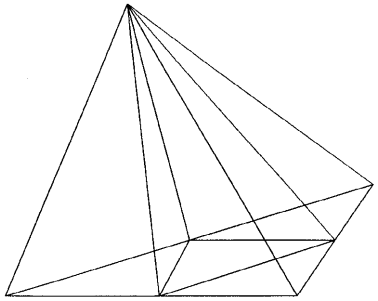


Figure 9. Equilateral tetrahedron divided into four finite elements. The shape of these elements deviates most from the equilateral shape, but they still have an excellent shape, well suited for integration

Algorithm 1

$$\begin{array}{lll} 1 & \text{if } L_p < L_n - 1 & \text{refine}(p) \\ 2 & \text{if } L_p > L_n + 1 & \text{refine } (n) \\ 3 & \text{if } L_p > L_n & n_{\mathrm{g}} = n_{\mathrm{g}} + 1 \\ 4 & \text{if } L_p < L_n & n_{\mathrm{l}} = n_{\mathrm{l}} + 1 \\ 5 & \text{if } n_{\mathrm{l}} > 1 & \text{refine}(p) \\ 6 & \text{if } n_{\mathrm{l}} > 0 \quad \text{and } n_{\mathrm{g}} > 0 & \text{refine}(n). \end{array} \tag{9}$$

In line 1 of Algorithm 1 the level of each tree element is compared with the level number of its neighbours. If the level number of a tree element is less than the level number of a neighbour minus one, then the element is refined. In line 2 a neighbour tree element is refined if the level number of the actual element is greater than the level number of the neighbour element plus one. In lines 3 and 4 the numbers of neighbour elements which have level numbers greater and less respectively are counted. In line 5 a tree element is refined if more than one neighbour has a smaller level number. In line 6 a neighbour tree element with the smallest level number is refined if an element has neighbours with both smaller and greater level numbers.

In three dimensions additional rules have to be applied to ensure the generation of valid finite elements. Let n_m be the number of midpoint nodes at the edges of the tetrahedron.

Algorithm 2

$$7 \quad \text{if } n_m \neq 1 \text{ and } n_m \neq 3 \quad \text{refine}(p). \tag{10}$$

After Algorithm 1 is applied, there will only be nodes at the midpoints of the edges of the tetrahedra. If the number of midpoint nodes is different from one and three, the element is refined.

Both Algorithms 1 and 2 are applied iteratively. First Algorithm 1 is repeated until no refinements occur. Then Algorithm 2 is applied. The whole balancing procedure terminates when no refinements occur either in Algorithm 1 or in Algorithm 2.

© 1997 by John Wiley & Sons, Ltd.

3. THE NAVIER–STOKES EQUATIONS

The non-linear Navier–Stokes equations are given by

$$-\mu\nabla^2\mathbf{v} + \rho\mathbf{v}\cdot\nabla\mathbf{v} + \nabla p = 0 \quad \text{in } \Omega, \tag{11}$$

$$-\nabla\cdot\mathbf{v} = 0 \quad \text{in } \Omega, \tag{12}$$

where $\mathbf{v}$ is the velocity vector, p is the pressure and μ is the viscosity coefficient. The first equation is the equation of motion which contains a diffusion and a pressure gradient term. The second equation is the equation of continuity. A minus sign is introduced in the continuity equation in order to obtain the same sign for the pressure gradient as for the continuity equation in the finite element formulation. In the finite element formulation the velocities are approximated with quadratic basis functions and the pressure is approximated with linear basis functions on each element.[17] Denote the quadratic polynomials N_i and the linear polynomials L_i. Then by the Galerkin residual method and integration by parts the second-order finite element formulation of the Navier–Stokes equation system becomes

$$\begin{aligned}
\mathbf{F}_\mathbf{v} &= \int_\Omega \mu\nabla N_i\cdot\nabla\mathbf{v}\mathrm{d}\Omega + \int_\Omega \rho N_i\nabla\cdot\mathbf{v}\mathrm{d}\Omega - \int_\Omega \nabla N_i p\mathrm{d}\Omega - \int_{\delta\Omega}\mu N_i\frac{\partial\mathbf{v}}{\partial n}\mathrm{d}\delta\Omega + \int_{\delta\Omega} N_i p\mathrm{d}\delta\Omega = 0,\\
\mathbf{F}_\mathbf{p} &= -\int_\Omega L_i\nabla\cdot\mathbf{v}\mathrm{d}\Omega = 0.
\end{aligned} \tag{13}$$

There are several methods to linearize this equation system. The common linearization techniques involve computation of gradients or approximate gradients, e.g. the Newton method or steepest descent methods. The Newton linearization method is a global method of linearization.

4. NEWTON LINEARIZATION

The Navier–Stokes equations have one non-linear term, the convective acceleration, which requires a non-linear iterative solution procedure. The non-linear algorithm chosen is the Newton method, which is known to have second-order convergence rate. The Navier–Stokes equations (13) will then have to be differentiated with respect to the unknowns and the linear equation system which has to be solved at each Newton step is

$$\begin{bmatrix} \partial\mathbf{F}_\mathbf{v}^n/\partial\mathbf{v} & \partial\mathbf{F}_\mathbf{v}^n/\partial p \\ \partial\mathbf{F} - \mathbf{p}^n/\partial\mathbf{v} & 0 \end{bmatrix}\begin{bmatrix}\Delta\mathbf{v}\\ \Delta\mathbf{p}\end{bmatrix} = -\begin{bmatrix}\mathbf{F}_\mathbf{v}^n\\ \mathbf{f}_\mathbf{v}^n\end{bmatrix}, \tag{14}$$

where the matrix and the right side are given by

$$\begin{aligned}
&\begin{bmatrix}\int_\Omega[\mu\nabla N_i\nabla N_j + \rho N_i(\nabla\mathbf{v}N_j + \mathbf{v}\nabla N_j)]\mathrm{d}\Omega - \int_\Omega \nabla N_i L_j\mathrm{d}\Omega - \int_\Omega L_i\nabla N_j\mathrm{d}\Omega 0\end{bmatrix}\begin{bmatrix}\Delta\mathbf{v}\\ \Delta\mathbf{p}\end{bmatrix}\\
&\quad = -\begin{bmatrix}\int_\Omega(\mu\nabla N_i\cdot\nabla\mathbf{v} + \rho N_i\mathbf{v}\cdot\nabla\mathbf{v} - \nabla N_i p)\mathrm{d}\Omega + \int_{\delta\Omega}(-\mu N_i\partial\mathbf{v}/\partial n + N_i p)\mathrm{d}\omega\Omega\\ -\int_\Omega L_i\nabla\cdot\mathbf{v}\mathrm{d}\Omega\end{bmatrix},
\end{aligned} \tag{15}$$

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \Delta\mathbf{v}, \tag{16}$$

$$\mathbf{p}^{n+1} = \mathbf{p}^n + \Delta\mathbf{p}. \tag{17}$$

 © 1997 by John Wiley & Sons, Ltd.

If the initial solution $\mathbf{v}^0, p^0$ is chosen close enough to the final solution, convergence of the non-linear equation system is guaranteed. The solution is then updated at each Newton step by the correction found by solving (15).

5. ADAPTIVE LINEARIZATION

An alternative or a supplement to Newton linearization of the equation system is local grid adaption to the solution,[2,3] which will also contribute to the linearization of the equation system. Consider the ratio of the non-linear convection term to the linear diffusion term in the Navier–Stokes equations:

$$\frac{\left\| \rho \int_\Omega N_i \mathbf{v} \cdot \nabla \mathbf{v} \mathrm{d}\Omega \right\|}{\left\| \mu \int_\Omega \nabla N_i \cdot \nabla \mathbf{v} \mathrm{d}\Omega \right\|}. \tag{18}$$

For simplicity in the present derivations let the velocities be approximated by linear basis functions in one dimension:

$$\begin{aligned} N_1 &= \frac{x_2 - x}{x_2 - x_1}, & N_2 &= \frac{x - x_1}{x_2 - x_1}, \\ \frac{\partial N_1}{\partial x} &= -\frac{1}{x_2 - x_1}, & \frac{\partial N_2}{\partial x} &= \frac{1}{x_2 - x_1}. \end{aligned} \tag{19}$$

The convection–diffusion ratio can then be expressed as

$$\begin{aligned} \frac{\left\| \rho \int_\Omega N_i \mathbf{v} \cdot \nabla \mathbf{v} \mathrm{d}\Omega \right\|}{\left\| \mu \int_\Omega \nabla N_i \cdot \nabla \mathbf{v} \mathrm{d}\Omega \right\|} &= \frac{\left\| \rho \dfrac{\partial u}{\partial x} \displaystyle\int_\Omega N_i u \mathrm{d}\Omega \right\|}{\left\| \mu \dfrac{\partial N_i}{\partial x} \dfrac{\partial u}{\partial x} (x_2 - x_1) \right\|} \\ &= \frac{\dfrac{u_2 - u_1}{x_2 - x_1} a(u)(x_2 - x_1)}{\left\| \mu \dfrac{(-1)^i u_2 - u_1}{x_2 - x_1} \dfrac{}{x_2 - x_1} (x_2 - x_1) \right\|} = \frac{a(u)(x_2 - x_1)}{b(u)} = \frac{a(u) l}{b(u)}, \end{aligned} \tag{20}$$

where $a(u)$ is the mean value of $\rho N_i u$ and $a(u)(x_2 - x_1) = \rho \int_\Omega N_i u \mathrm{d}\Omega$, $b(u) = \mu$ and $l = x_2 - x_1$. The same expression can be derived for higher-order basis functions in two and three dimensions:

$$\frac{\left\| \rho \int_\Omega N_i \mathbf{v} \cdot \nabla \mathbf{v} \mathrm{d}\Omega \right\|}{\left\| \mu \int_\Omega \nabla N_i \cdot \nabla \mathbf{v} \mathrm{d}\Omega \right\|} = \frac{a(\mathbf{v}) l}{b(\mathbf{v})}. \tag{21}$$

In the above formula, $a(\mathbf{v})$ and $b(\mathbf{v})$ are only functions of the velocities inside the element and are independent of the element size. The length l is some characteristic length of the element. The formulae show that the magnitude of the matrix coefficient of the convection can be reduced arbitrarily compared with the diffusion coefficient in the implicit equation system by local refinements. The above relation is valid in both two and three dimensions and for first- and second-order polynomial approximations of the Navier–Stokes equations. By reducing the element size where the convection is large, the equation system becomes more and more linear and symmetric.

© 1997 by John Wiley & Sons, Ltd.

Provided that the local element size is reduced sufficiently, this implicit adaptive linearization will for many Navier–Stokes applications prove to be sufficient and satisfactory.

6. TRI-TREE GRID ADAPTION TO THE SOLUTION

The Reynolds number for fluid flow is usually defined as

$$Re = \rho U_b d/\mu, \tag{22}$$

where ρ is the density and μ the viscosity of the fluid. The velocity U_b and the length d are some characteristic velocity and diameter in the flow geometry. For flow in a straight tube U_b is the mean inlet velocity and d is the diameter. For more complex geometries it is not possible to use a single number to characterize the flow conditions. The element Reynolds number is defined as

$$Re_e = \frac{\sum_i N_i^c \left\| \rho \int_\Omega N_i \mathbf{v} \cdot \nabla \mathbf{v} d\Omega \right\|}{\sum_i N_i^c \left\| \mu \int_\Omega \nabla N_i \cdot \nabla \mathbf{v} d\Omega \right\|} < \varepsilon_A, \tag{23}$$

where N_i^c is the basis function evaluated at the geometrical centre. The refinements and recoarsements of elements[3,9] are decided from the element Reynolds number, which is calculated from the nodal values and weighted by the basis functions evaluated at the geometrical centre. The size of the element Reynolds number Re_e indicates the degree of non-linearity in the equation system. By reducing the element size by refinements, the magnitude of the non-linear coefficients in the equation matrix will also decrease:

$$Re_e > \varepsilon_A. \tag{24}$$

Before the finite element grid is adapted to the previous solution of lower Reynolds number, the solution at all nodes is scaled by U_b^n/U_b^o, the ratio between the new and the old velocity boundary condition. The scaled finite element grid solution is prolonged to the tri-tree grid. The element Reynolds number is computed for the tri-tree elements. First, the tri-tree elements are recoarsed. During recoarsements the element Reynolds number of the tri-tree element above the terminal element in the tri-tree is computed. If the element Reynolds number Re_e, of this tri-tree element is below the adaption limit ε_A, this tri-tree element is made terminal and the four leaf elements at the finer adaption levels are discarded. The recoarsement algorithm starts from the terminal leaves of the tri-tree and traverses towards the root of the tri-tree. Owing to the tri-tree's hierarchic tree structure, the recoarsement algorithm becomes recursive. At the end of the recoarsement procedure the tri-tree contains elements with element Reynolds numbers just below the adaption limit ε_A and elements with element Reynolds numbers above the adaption limit which have not been recoarsed. At this stage no elements in the tri-tree can be recoarsed without introducing a new tri-tree element with element Reynolds number above the adaption limit. When the recoarsement algorithm is finished, the refinements are performed. The tri-tree is then traversed from the root towards the leaves. The refinements will then be recursive. When the element Reynolds number is above the refinement limit, the tri-tree element is refined into four new tri-tree elements. The element Reynolds number of these four elements will be computed later in the refinement algorithm and they will be refined later if their element Reynolds number is above the refinement limit ε_A.

In all previous two-dimensional experiments[2,3,9] the element Reynolds number has been evaluated by quadratic polynomials. However, as the computation of quadratic velocity approximation is more time-consuming than that of linear velocity approximation, linear velocity approximation is used in the present work for the computations of the element Reynolds numbers. The effects on the grid of

 © 1997 by John Wiley & Sons, Ltd.

using linear instead of quadratic velocity approximation for the element Reynolds number were minor.

7. MULTIGRID ALGORITHM

The Navier–Stokes equations can be expressed as an equation system $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, where

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_\mathbf{v} \\ \mathbf{f}_\mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathscr{A}(\mathbf{F}_\mathbf{v}) \\ \mathscr{A}(\mathbf{F}_\mathbf{p}) \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} \mathbf{x}_\mathbf{v} \\ \mathbf{x}_\mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathscr{A}(\mathbf{v}) \\ \mathscr{A}(\mathbf{p}) \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} \mathbf{b}_\mathbf{v} \\ \mathbf{b}_\mathbf{p} \end{bmatrix}. \tag{25}$$

The function $\mathscr{A}$ denotes assembly of element vectors into a global vector. $\mathbf{f}_\mathbf{v}$ and $\mathbf{v}_\mathbf{p}$ are velocity and pressure equations assembled from the element velocity equation vector $\mathbf{F}_\mathbf{v}$ and pressure equations vector $\mathbf{F}_\mathbf{p}$. $\mathbf{b}_\mathbf{v}$ and $\mathbf{b}_\mathbf{p}$ are velocity and pressure parts of the right-hand side. The global velocity vector $\mathbf{x}_\mathbf{v}$ and vector pressure $\mathbf{x}_\mathbf{p}$ are assembled from $\mathbf{v}$ and $\mathbf{p}$ respectively.

8. CONVERGENCE CRITERIA

The tri-tree multigrid solver consists of three iterative algorithms inside each other. The inner iterative algorithms is CGSTAB, the linear equation solver. For each Newton iteration a set of linear equations is solved. The iterative Newton algorithm is performed for each grid level. The convergence criteria based on relative corrections and residuals of the solution are

$$\begin{aligned} \frac{\|\delta^\mathrm{L}\mathbf{x}_\mathbf{v}^k\|}{U_\mathrm{b}} &< \varepsilon_{\mathrm{L}_x}, & \frac{\|\delta^\mathrm{L}\mathbf{r}_\mathbf{v}^k\|}{U_\mathrm{b}} &< \varepsilon_{\mathrm{L}_r}, \\ \frac{\|\delta_\mathrm{N}\mathbf{x}_\mathbf{v}^k\|}{U_\mathrm{b}} &< \varepsilon_{\mathrm{N}_x}, & \frac{\|\delta^\mathrm{N}\mathbf{r}_\mathbf{v}^k\|}{U_\mathrm{b}} &< \varepsilon_{\mathrm{N}_r}, \end{aligned} \tag{26}$$

where U_b is the velocity boundary condition, $\delta^\mathrm{L}\mathbf{x}_\mathbf{v}$ is the update of the velocity solution in the linear equation solver, $\delta^\mathrm{N}\mathbf{x}_\mathbf{v}$ is the update of the velocity solution in the Newton iterations and $\delta^\mathrm{G}\mathbf{x}_\mathbf{v}^k$ is the velocity difference between the prolonged start vector and the velocity vector at grid level k. $\varepsilon_{\mathrm{L}_x}$ and $\varepsilon_{\mathrm{L}_r}$ are the linear and $\varepsilon_{\mathrm{N}_x}$ and $\varepsilon_{\mathrm{N}_r}$ the non-linear grid convergence criteria. The residual is defined by $\mathbf{r} = \mathbf{b} - \mathscr{A}\mathbf{x}$. The convergence of the iterations is obtained when the largest of the two norms has reached the convergence limit $\varepsilon_{\mathrm{L}_x} = \varepsilon_{\mathrm{L}_r}$.

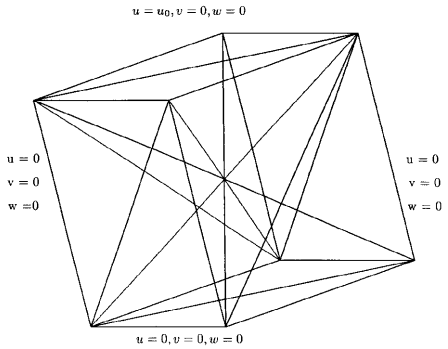
9. EXPERIMENTS

In the present investigation the limit of the element Reynolds number for the three-dimensional cavity in order to achieve convergence of the linear CGSTAB solver is established. The investigation of the prolonged adapted multigrid method consists of comparing the solution properties for the two- and the three-dimensional driven cavity.

9.1. The element Reynolds number

The initial three-dimensional driven cavity with boundary conditions is shown in Figure 10. This cavity is regularly refined to a cavity associated with an $8 \times 8 \times 8$ cavity which is used as basis for the computations. This cavity model consists of 6144 elements, 9009 nodes and 28,268 degrees of freedom. The boundary conditions are zero velocities at all sides, except at one side where a constant tangential velocity is specified. These boundary conditions, however, imply singularities in the

© 1997 by John Wiley & Sons, Ltd.



(27)

Figure 10. Initial three-dimensional $8 \times 8 \times 8$ grid with boundary conditions

velocities along the edges of the side where the boundary conditions are specified. These singularities are reflected in the solution.

The results of the flow simulations for different grid resolutions of the cavity are given in Table I. The simulations were performed for three grid sizes. When the maximum element Reynolds number Re_e in the grid was increased above the values in the grid, the linear equation solver CGSTAB diverged. Table 1 also shows that the error estimate based on the residual norm is consistently less than the error estimate based on the norm of the update in the linear iterations. The divergence for all three grids occurs when the element Reynolds number is approximately 30. The divergence occurs at lower element Reynolds number for finer grids.

Table II gives the results of the element Reynolds number investigations when the Reynolds number is increased by scaling the size of the cavity. The size of the cavity is scaled by a factor corresponding to an increase in the Reynolds number of 200. For the $4 \times 4 \times 4$ grid a convergent solution for the linear equation solver is obtained when the maximum element Reynolds number is less than 37. For the $8 \times 8 \times 8$ grid the maximum element Reynolds number limit is 28.

In previous work[9] it was shown experimentally that in two dimensions the maximum element Reynolds number limit Re_e in order to achieve convergence with the CGSTAB iterative solver was within the range 9·34–11·49. The maximum element Reynolds number limit for convergence in three dimensions also varies slightly for different grid resolutions and also with the way in which this limit is approached. The three-dimensional element Reynolds number limit is between 27·28 and 36·76, approximately three times greater than for the maximum element Reynolds number limit found in two dimensions.[9] In the comparison of the simulations in two and three dimensions the adaptive

Table I. Grid level, grid size, cavity Reynolds number and maximum element Reynolds number for grid. The last two columns are the error estimates based on the correction and residual respectively for the Newton iterations. The linear convergence level for CGSTAB is $\varepsilon_L = 10^{-4}$. The number of Newton iterations is fixed at six for all Reynolds numbers and grids. The element Reynolds number is increased by increasing the cavity boundary condition U_b in steps of 0·0001. For all grid levels the solution diverged when the boundary velocity U_b was further increased

Level	Grid size	$Re = \rho U_b d/\mu$	Max. Re_e	ε_{N_x}	ε_{N_r}
1	$2 \times 2 \times 2$	700	36·76	0·00116	0·0000001
2	$4 \times 4 \times 4$	1100	33·65	0·00246	0·0000007
3	$8 \times 8 \times 8$	1800	27·49	0·00105	0·0000001

 © 1997 by John Wiley & Sons, Ltd.

Table II. Maximum Re_e for fixed boundary velocity $U_b = 0{\cdot}0001$ and grid sizes $4 \times 4 \times 4$ and $8 \times 8 \times 8$. The linear convergence criterion is $\varepsilon_L = 10^{-4}$ and the number of Newton iterations is fixed at six. The cavity Reynolds number is increased by 200 by scaling the cavity co-ordinates

Grid	*Re*	Cavity size	Max. Re_e
$4 \times 4 \times 4$	400	4·0	11·60
	600	6·0	17·83
	800	8·0	23·95
	1000	10·0	30·42
	1200	12·0	36·71
$8 \times 8 \times 8$	400	4·0	7·25
	600	6·0	8·74
	800	8·0	11·83
	1000	10·0	15·86
	1200	12·0	18·00
	1400	14·0	21·97
	1600	16·0	24·16
	1800	18·0	27·28

maximum element Reynolds number limit is set to 1 in two dimensions and 3 in three dimensions to obtain an approximate grid resolution.

9.2. Two- and three-dimensional simulations

The adapted grids for two-dimensional flow are displayed in Figure 11. The corresponding data from the simulations are given in Tables III and IV. Table III gives the computational times for grid adaption, matrix assembly and the equation solver. The last column gives the number of degrees of freedom. The table shows that the time used for grid adaption is only a very small fraction of the times used for assembling the matrix and solving the equation system. The time used for assembling the matrix is also considerably less than the time used for solving the equation system. The error estimates based on the Newton update and the Newton residual are given in Table IV. The table shows that the error based on the residual is several orders smaller than the error based on the Newton update. Therefore the strictest convergence criterion appears by using the Newton update criterion. The last column gives the total number of linear iterations for each Reynolds number. The total number of linear iterations seems to be remarkably constant and independent of the Reynolds number.

The projection of the three-dimensional grids is shown in Figure 12 and the corresponding simulation data are given in Tables V and VI. The three-dimensional results also reveal that the computational time used in the grid adaption procedure is considerably less than the times used for matrix assembly and equation solution. However, in contrast with the two-dimensional simulations, the solution time is less than the assembly time in three dimensions. In three dimensions most of the time is used for generating the element matrix. The error estimates in three dimensions show that the error based on the Newton update is considerably smaller than the error based on the Newton residual. The total number of linear iterations is also fairly constant and independent of the Reynolds number.

The two-dimensional velocity and pressure isobar solutions for Reynolds number 1200 are shown in Figure 13. The corresponding three-dimensional solutions are shown in Figures 14–17.

© 1997 by John Wiley & Sons, Ltd.

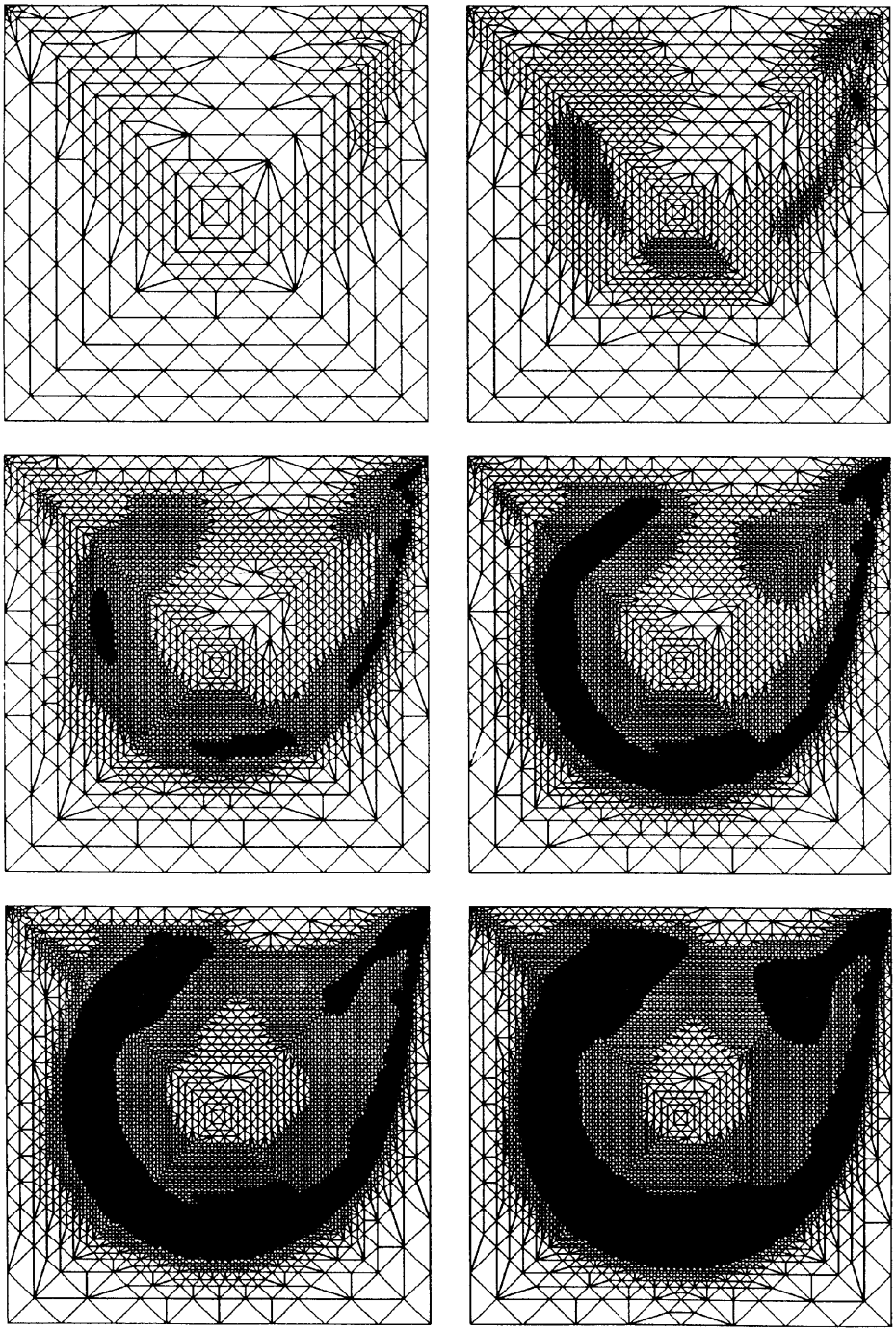


Figure 11. Two-dimensional adapted grids for Reynolds numbers 200, 400, 600, 800, 1000 and 1200

© 1997 by John Wiley & Sons, Ltd.

Table 3. Two dimensions: computational time for adapting grid to solution, time for assembling equation system, time for solving equation system and of unknowns for adaption level $\varepsilon_A = 1{\cdot}0$. The cavity Reynolds number is increases in velocity steps of 0·0002 with steps in the Reynolds number of 200. The time unit seconds

Re	Time for adaption	Time for assembly	Time for solution	Unknowns
200	1	45	82	2916
400	4	218	580	13556
600	11	533	1680	32777
800	24	1130	2570	68511
1000	40	1800	4120	108037
1200	65	2690	17700	159985

Table IV. Two dimensions: error estimates based on Newton increment and Newton residual and total number of linear iterations for each Reynolds number. The number of Newton iterations for each Reynolds number is six

Re	ε_{N_x}	ε_{N_r}	Linear iterations
200	0·0007	0·0000011	222
400	0·0025	0·0000012	319
600	0·0078	0·0000015	373
800	0·0063	0·0000008	279
1000	0·0048	0·0000033	286
1200	0·0041	0·0000032	269

The three-dimensional velocity vector solution is shown in Figure 14. The figure shows that the centre of the vortex moves towards the lower right corner as the Reynolds number is increased. Figure 15 shows the velocity profiles in the lower right corner of the cavity. The velocity profiles in the figure are all symmetric with respect to the centre plane, although some of them may look distorted owing to the rotation and projection of the three-dimensional velocity vectors. The figure shows that from Reynolds number 400 and upwards the reversed flow velocities are smaller in the central part than in the regions closer to the walls. The reason for the shape of these velocity profiles may be that more energy is needed to reverse the initial high velocities in the centre.

The pressure isobar solution is shown in Figure 16. The isobars in Figures 16 are drawn in vertical planes in the flow direction. The figure shows that there is a low-pressure zone at the upstream edge and a high-pressure zone at the downstream edge. A local pressure minimum is observed at the centre of the vortex. Figure 17 shows the pressure isobars drawn in vertical planes normal to the main direction of flow.

Although the two- and three-dimensional flows are quite different in nature, the flow in the centre plane of the three-dimensional cavity shows nearly the same properties as for two-dimensional cavity flow.

10. CONCLUSIONS

In the present work a three-dimensional prolonged adaptive multigrid method has been presented. The method has been tested for the three-dimensional driven cavity problem. The algorithm consists of the tri-tree grid generation algorithm, the element Reynolds number refinement algorithm and the

© 1997 by John Wiley & Sons, Ltd.

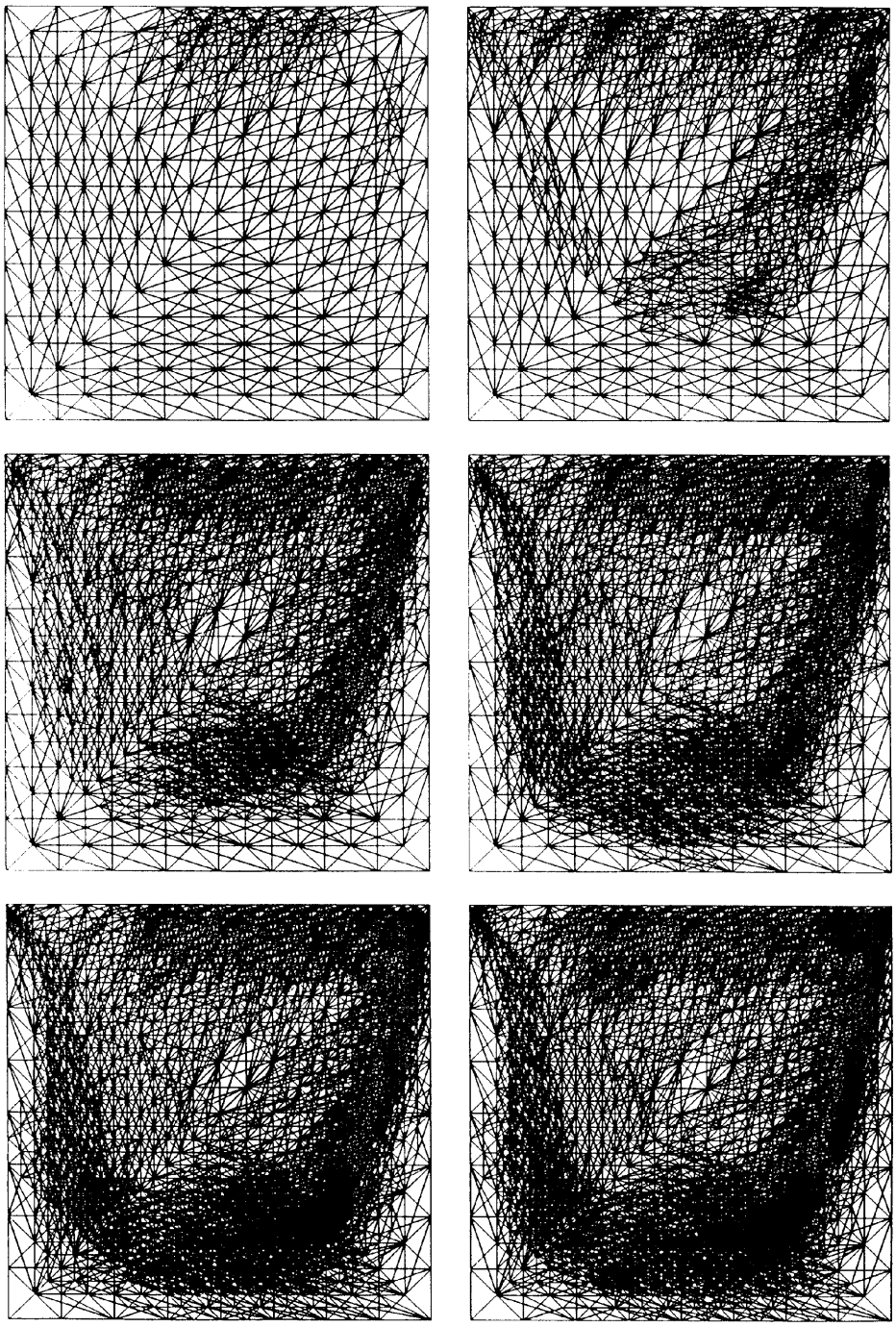


Figure 12. Three-dimensional adapted grids for Reynolds numbers 200, 400, 600, 800, 1000 and 1200

© 1997 by John Wiley & Sons, Ltd.

Table V. Three dimensions: computational time for adapting grid to solution, time for assembling equation system, time for solving equation system and number of unknowns for adaption level $\varepsilon_A = 3{\cdot}0$. The cavity Reynolds number is increased in velocity steps of 0·0002 with steps in the Reynolds number of 200. The time unit is seconds

Re	Time for adaption	Time for assembly	Time for solution	Unknowns
200	18	1770	371	28268
400	117	5430	2650	84756
600	3530	11900	3410	180200
800	9650	22200	8450	330632
1000	22200	32300	12600	513509
1200	35300	45900	14000	727067

Table VI. Three dimensions: error estimates based on Newton increment and Newton residual and total number of linear iterations for each Reynolds number. The number of Newton iterations for each Reynolds number is six

Re	ε_{N_x}	ε_{N_r}	Linear iterations
200	0·0004	0·00000002	64
400	0·0035	0·00000022	91
600	0·0021	0·00000017	87
800	0·0013	0·00000003	84
1000	0·0087	0·00000029	107
1200	0·0022	0·00000006	83

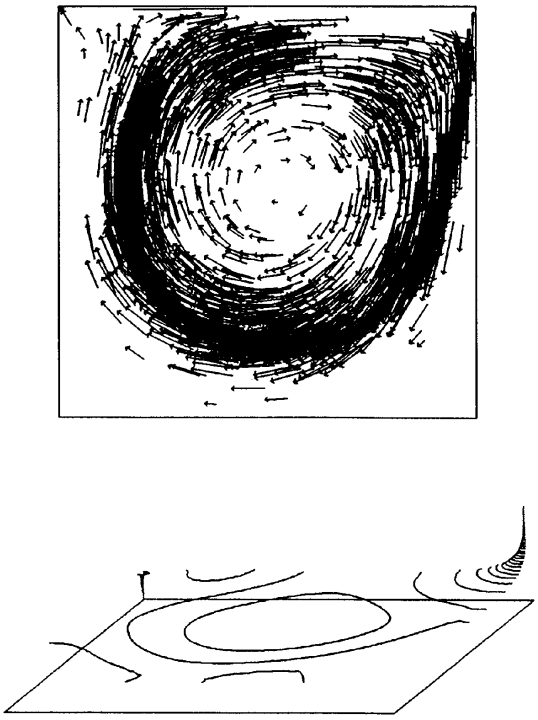


Figure 13. Two-dimensional velocity and pressure isobar solutions for Reynolds number 1200

© 1997 by John Wiley & Sons, Ltd.

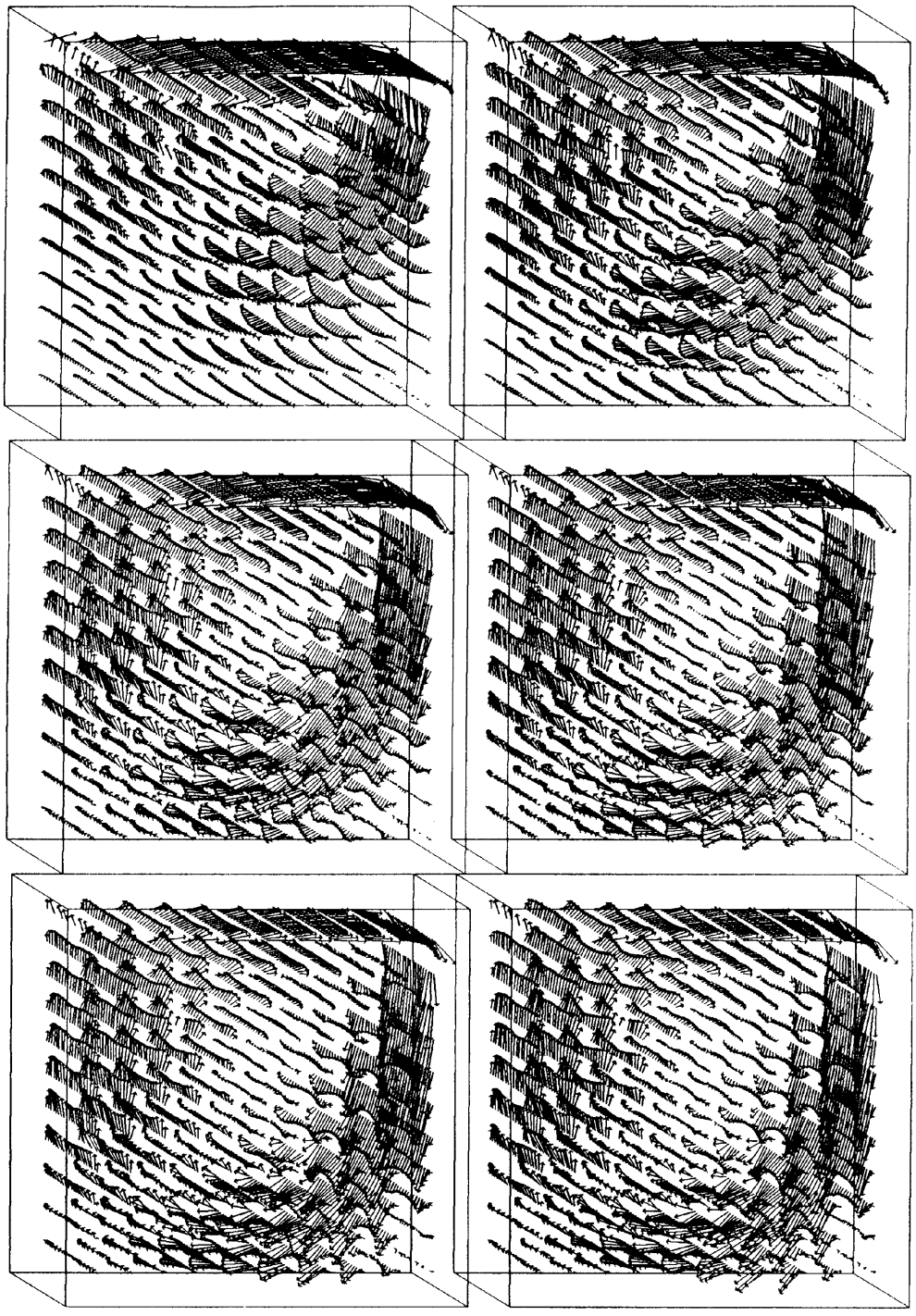


Figure 14. Three-dimensional velocity vectors for Reynolds numbers 200, 400, 600, 800, 1000 and 1200

© 1997 by John Wiley & Sons, Ltd.

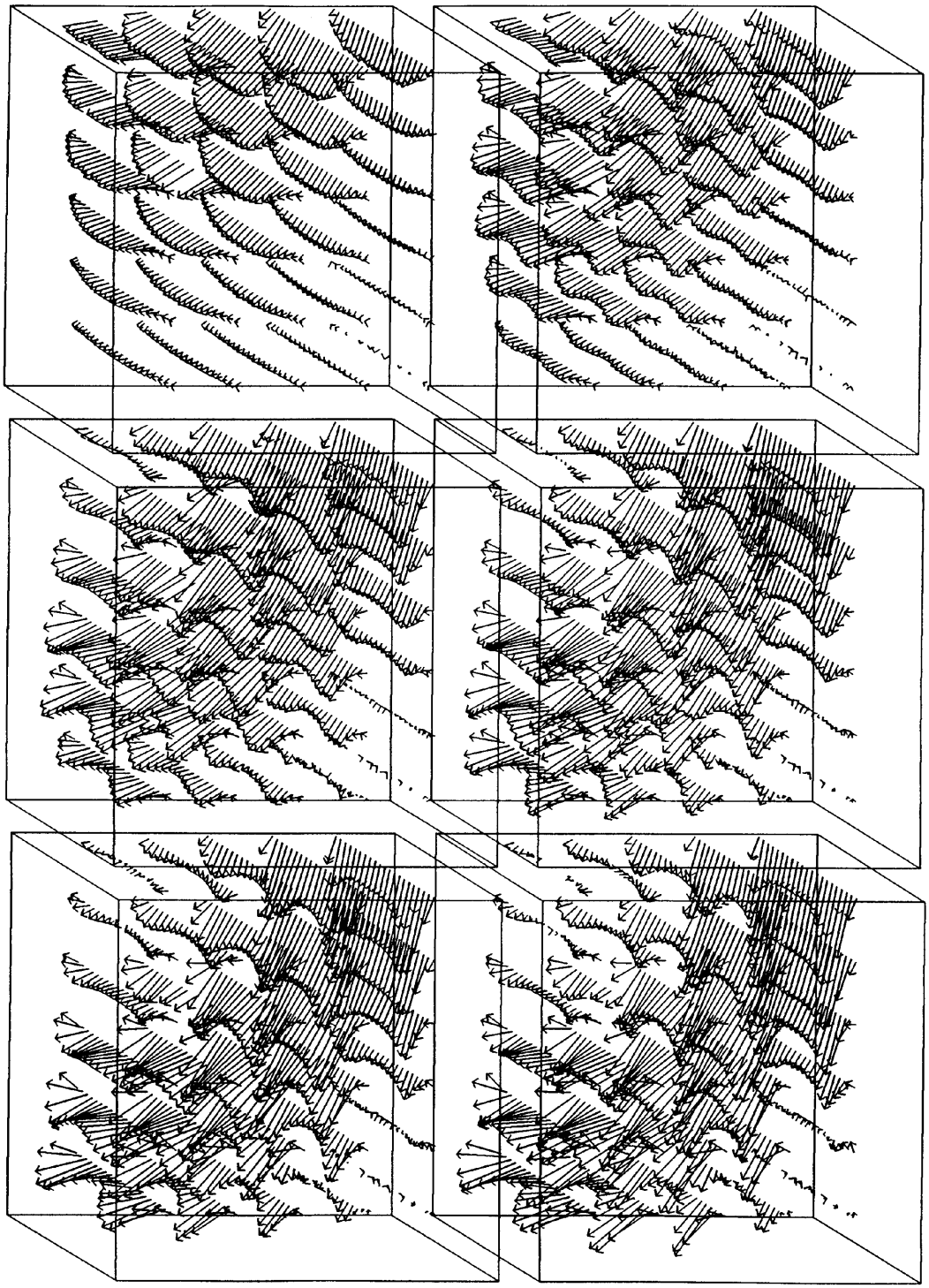


Figure 15. Velocity profiles in lower right corner of cavity for Reynolds numbers 200, 400, 600, 800, 1000 and 1200

© 1997 by John Wiley & Sons, Ltd.

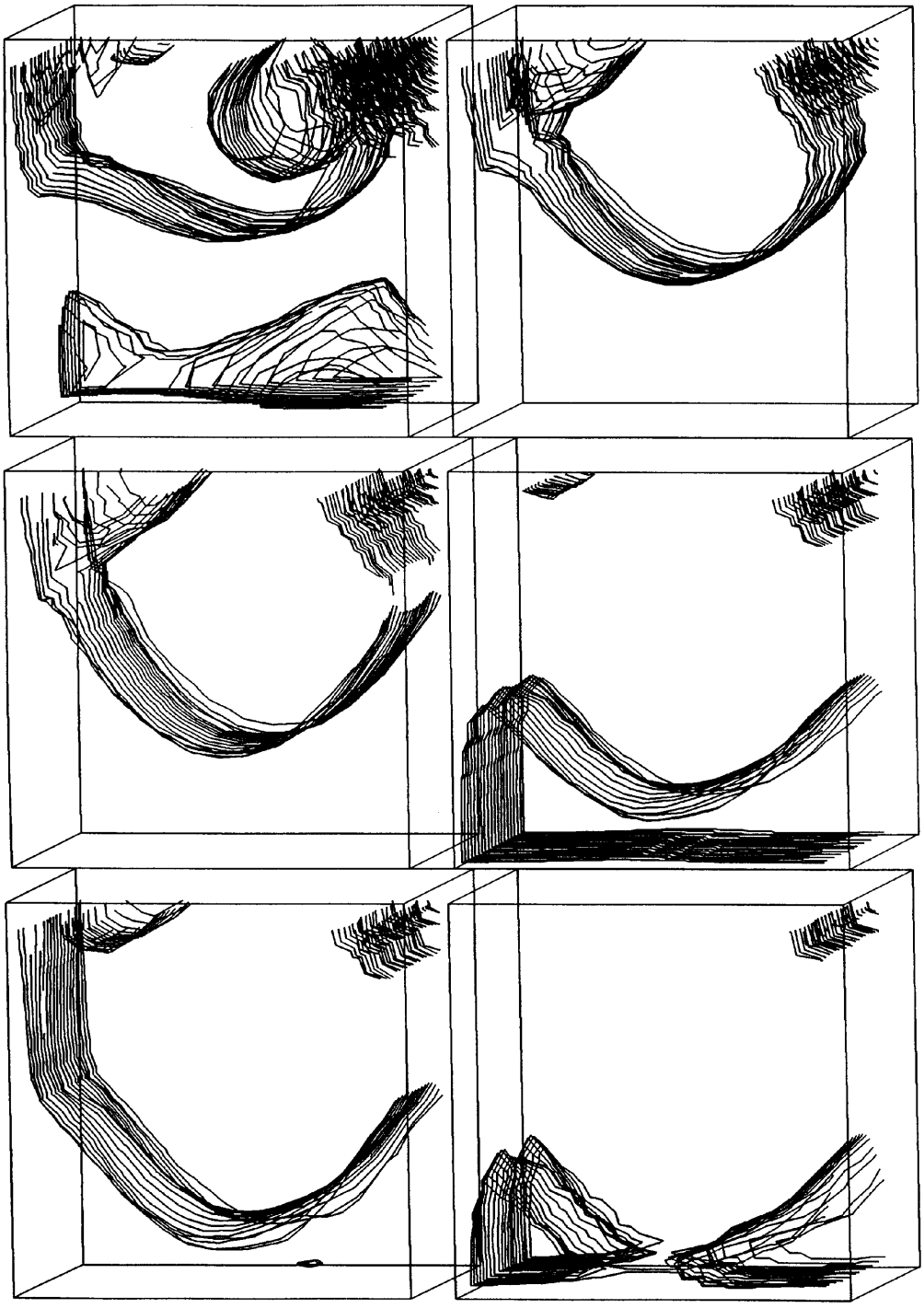


Figure 16. Three-dimensional pressure isobars for Reynolds numbers 200, 400, 600, 800, 1000 and 1200

© 1997 by John Wiley & Sons, Ltd.

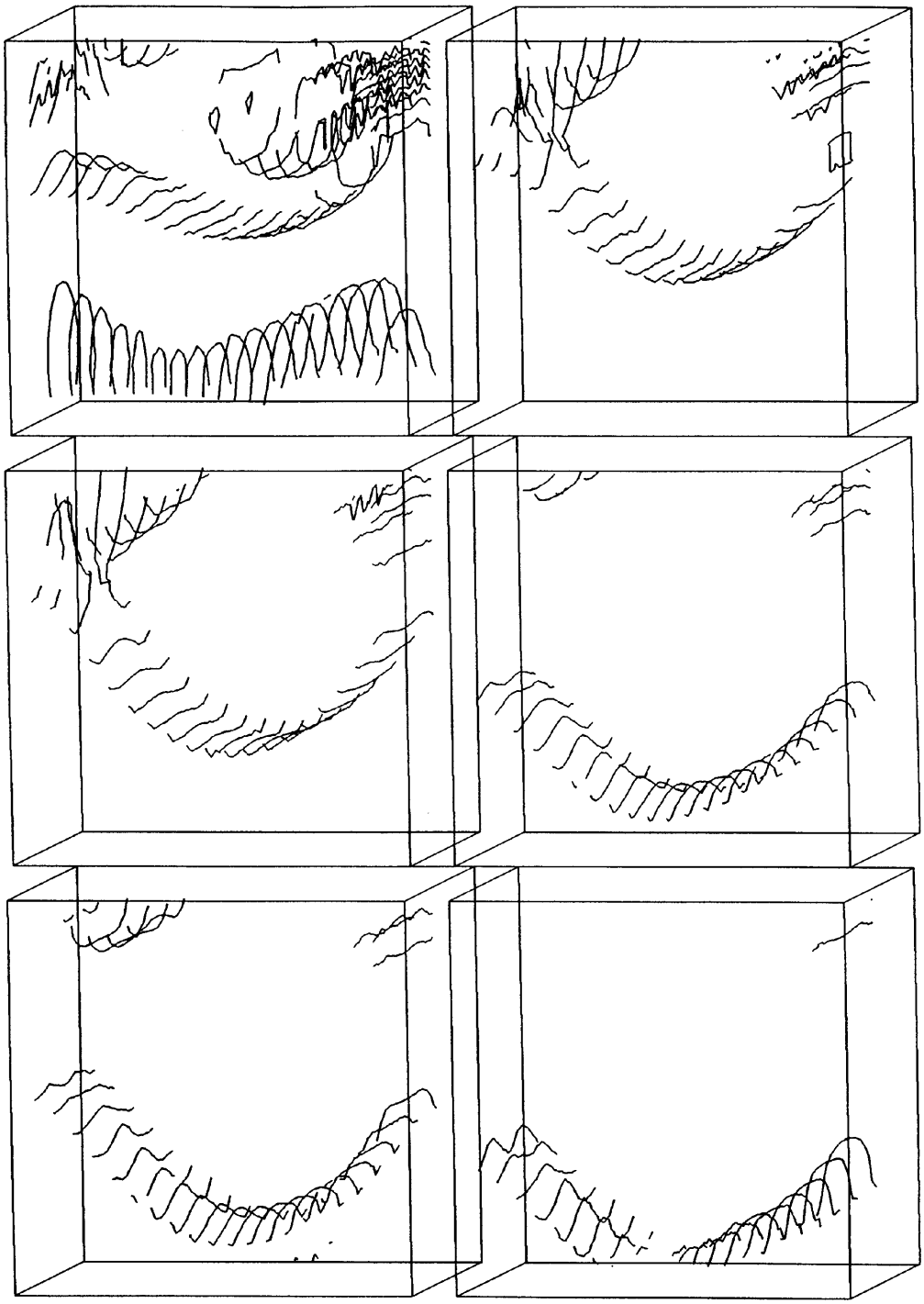


Figure 17. Three-dimensional pressure isobars in vertical planes normal to main flow direction for Reynolds numbers 200, 400, 600, 800, 1000 and 1200

© 1997 by John Wiley & Sons, Ltd.

coupled node fill-in preconditioned CGSTAB iterative solver. As expected, the prolonged adaptive multigrid method also works satisfactorily in three dimensions. Future work will be directed towards solving real three-dimensional problems.

ACKNOWLEDGEMENTS

The author is grateful to Demosthenes Skipitaris for skilled computer advice and to Trygve Svoldal for valuable suggestions and corrections of the manuscript. The project has been supported by the Norwegian Research Council, grant NN2461K, for partial financing of the computer run time expenses.

REFERENCES

1. S. Ø. Wille, 'A structured tri-tree search method for generation of optimal unstructured finite element grids in two and three dimensions', *Int. j. numer. methods fluids*, **14** 861–881 (1992).
2. S. Ø. Wille, 'A local solution adapted tri-tree multigrid generator and equation solver for mixed finite element formulation of the Navaier–Stokes equations', *Comput. Methods Appl. Mech. Eng.*, **131**, 109–132 (1996).
3. S. Ø. Wille, 'Adaptive linearization and grid iterations with the tri-tree multigrid refinement–recoarsement algorithm for the Navier–Stokes equations', *Int. j. numer. methods fluids*, **24**, 155–168 (1997).
4. S. Ø. Wille, 'A non-linear adaptive tri-tree multigrid solver for mixed finite element formulation of the Navier–Stokes equations', *Int. j. numer. methods in fluids*, **22**, 1041–1059 (1996).
5. P. Sonneveld, 'CGS, a fast Lanczos-type solver for nonsymmetric linear systems', *SIAM J. Stat. Comput.*, **10**, 36–52 (1987).
6. H. van der Vorst, 'A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear equations', *SIAM J. Sci. Stat. Comput.*, **13**, 631–644 (1992).
7. G. F. Carey, K. C. Wang and W. D. Joubert, Performance of iterative methods for Newtonian and generalized Newtonian flows', *Int. j. numer. methods fluids*, **9**, 127–150 (1989).
8. O. Dahl and S. Ø. Wille, 'An ILU preconditioner with coupled node fill-in for iterative solution of the mixed finite element formulation of the 2D and 3D Navier–Stokes equations', *Int. j. numer. methods fluids*, **15**, 525–544 (1992).
9. S. Ø. Wille, 'A local predictive convection–diffusion refinement indicator for the tri-tree adapted finite element multigrid algorithm of the Navier–Stokes equations', *Comput. Methods Appl. Mech. Eng.*, **134**, 181–196 (1996).
10. Y. Kallinderis, 'Adaptive hybrid prismatic/tetrahedral grids', *Int. j. numer. methods fluids*, **20**, 1023–1037 (1992).
11. R. Verfurth, '*A posteriori* error estimators for the Stokes equations', *Numer. Math.*, **55**, 309–325 (1989).
12. R. E. Banks and B. D. Welfert, '*A posteriori* error estimates for the Stokes equations: a comparison', *Comput. Methods Appl. Mech. Eng.*, **82**, 323–340 (1990).
13. O. C. Zienkiewicz and J. Z. Zhu, 'The superconvergent patch recovery and *a posteriori* error estimates. Part 2: Error estimates and adaptivity', *Int. j. numer. methods eng.*, **33**, 1365–1382 (1992).
14. S. Ø. Wille, 'A non-linear adaptive full tri-tree multigrid method for the mixed finite element formulation of the Navier–Stokes equations', *Int. j. numer. methods fluids*, in press (1997).
15. W. Hackbush, *Multi-Grid Methods and Applications*, Springer, Berlin, 1985.
16. P. Wesseling, *An Introduction to Multigrid Methods*, Wiley, Chichester, 1992.
17. C. Taylor and P. Hood, 'A numerical solution of the Navier–Stokes equations using the finite element technique', *Comput. Fluids*, **1**, 73–100 (1973).

 © 1997 by John Wiley & Sons, Ltd.