INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 24, 155–168 (1997)

ADAPTIVE LINEARIZATION AND GRID ITERATIONS WITH THE TRI-TREE MULTIGRID REFINEMENT–RECOARSEMENT ALGORITHM FOR THE NAVIER–STOKES EQUATIONS

SVEN ØIVIND WILLE

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

SUMMARY

The tri-tree algorithm for refinements and recoarsements of finite element grids is explored. The refinement–recoarsement algorithm not only provides an accurate solution in certain parts of the grid but also has a major influence on the finite element equation system itself. The refinements of the grid lead to a more symmetric and linear equation matrix. The recoarsements will ensure that the grid is not finer than is necessary for preventing divergence in an iterative solution procedure. The refinement–recoarsement algorithm is a dynamic procedure and the grid is adapted to the instant solution.

In the tri-tree multigrid algorithm the solution from a coarser grid is scaled relatively to the increase in velocity boundary condition for the finer grid. In order to have a good start vector for the solution of the finer grid, the global Reynolds number or velocity boundary condition should not be subject to large changes. For each grid and velocity solution the element Reynolds number is computed and used as the grid adaption indicator during the refinement–recoarsement procedure.

The iterative tri-tree multigrid method includes iterations with respect to the grid. At each Reynolds number the same boundary conditions are applied and the grid is adapted to the solution iteratively until the number of unknowns and elements in the grid becomes constant. In the present paper the following properties of the tri-tree algorithm are explored: the influence of the increase in boundary velocities and the size of the grid adaption indicator on the amount of work for solving the equations, the number of linear iterations and the solution error estimate between grid levels. The present work indicates that in addition to the linear and non-linear iterations, attention should also be given to grid adaption iterations.

KEY WORDS: tri-tree multigrid; ILU coupled node fill-in preconditioner; element Reynolds number; adaptive refinements; adaptive recoarsements; grid iterations

1. INTRODUCTION

The development of fast algorithms for solving the Navier–Stokes equations has been subject to extensive research. Baruzzi, *et al.*[1] use the finite element method with a second-order artificial viscosity scheme. The equations are linearized by the Newton method and solved with a parallel direct solver. Hill and Baskharone[2] have implemented quadratic interpolation for both velocity and pressure fields with streamline upwinding. The equations are solved by a multiblock technique with active and inactive blocks with respect to the direct equation solver. Zang and Street[3] have developed a multigrid method where the nodes of the grid at different levels do not necessarily coincide and they report that the composite grid solutions have an accuracy comparable with that of the single-grid solutions with similar grid size. Lorber and Carey[4] have developed an efficient vector–parallel

CCC 0271–2091/97/020155–14

Received November 1995
Revised March 1996

algorithm for multi-gigaflop performance rates for large-scale simulation of the Navier–Stokes equations. Kallinderis[5] monitors the values of velocity differences and gradients across cell edges in order to divide/delete grid cells for inviscid 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. The rules for triangle and tetrahedron division are similar to those presented by Wille.[6]

The purpose of the present work is eventually to develop algorithms suited for multiprocessing. In previous papers a new tri-tree grid generator algorithm[6] and a new incomplete preconditioning algorithm[7] have been described. The tri-tree grid generation algorithm was well suited to organizing and structuring grids at different levels of refinement.[8] The tri-tree grid generation algorithm was coupled to the solution algorithm to achieve an adaptive grid refinement–recoarsement algorithm.[9] An exploration of the iterative equation solver and the local element Reynolds number[10] revealed that the linear equation solver always converged when the element Reynolds number was less than 10. Recoarsements of the finite element grids[11] during the solution procedure were introduced in the tri-tree grid generation algorithm in order to reduce the number of degrees of freedom to a minimum. With the refinement–recoarsement algorithm the Reynolds number is estimated from the start vector of the solution algorithm. Tri-tree elements with Reynolds numbers above a limit are refined iteratively and tri-tree elements with Reynolds number below the limit are recoarsed iteratively until all element Reynolds numbers are just below the predefined limit. The coarsest possible element level is also defined to ensure a certain number of elements in the grid.

The present multigrid algorithm begins by solving the equation system for a coarse grid and a low Reynolds number. The solution from this grid is scaled up to a higher Reynolds number and the grid is then adapted to this scaled solution. In the grid adaptation process the elements are refined or recoarsed until the element Reynolds number is just below the element Reynolds number adaption limit ε_{A}. The smaller the adaption limit used, the more symmetric and linear the equation system becomes. The smaller the increase in the velocity boundary condition at each level of equation solving, the closer the start vector is to the final solution, which should minimize the number of linear and non-linear iterations. The adaption limit of the element Reynolds number and the increase in boundary velocity condition are explored.

2. EQUATIONS

The non-linear Navier–Stokes equations are given by

$$-\mu\nabla^2\mathbf{v} + \rho\mathbf{v}\cdot\nabla\mathbf{v} + \nabla p = 0 \quad \text{in } \Omega, \tag{1}$$

$$-\nabla\cdot\mathbf{v} = 0 \quad \text{in } \Omega, \tag{2}$$

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 by quadratic basis functions and the pressure is approximated by linear basis functions on each element.[12] Denote the quadratic

polynomials N_i and the linear polynomial 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

$$\begin{aligned}\mathbf{F}_\mathbf{v} &= \int_\Omega \mu \nabla N_i \cdot \nabla \mathbf{v}\, d\Omega + \int_\Omega \rho N_i \mathbf{v} \cdot \nabla \mathbf{v}\, d\Omega - \int_\Omega \nabla N_i p\, d\Omega - \int_{\delta\Omega} \mu N_i \frac{\partial \mathbf{v}}{\partial n} d\delta\Omega + \int_{\delta\Omega} N_i p\, d\delta\Omega = 0,\\ \mathbf{F}_\mathbf{p} &= \int_\Omega L_i \nabla \cdot \mathbf{v}\, d\Omega = 0.\end{aligned} \tag{3}$$

There are several methods to linearize this equation system. Usual linearization techniques involve computation of gradients or approximate gradients as in the Newton method or steepest descent methods. The Newton linearization method is a global method of linearization.

3. 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 a second-order convergence rate. The Navier–Stokes equations (3) 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

$$\begin{bmatrix} \partial \mathbf{F}_\mathbf{v}^n / \partial \mathbf{v} & \partial \mathbf{F}_\mathbf{v}^n / \partial p \\ \partial \mathbf{F}_p^n / \partial \mathbf{v} & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{v} \\ \Delta \mathbf{p} \end{bmatrix} = - \begin{bmatrix} \mathbf{F}_\mathbf{v}^n \\ \mathbf{F}_\mathbf{p}^n \end{bmatrix}, \tag{4}$$

where the matrix and the right side are given by

$$\begin{bmatrix} \displaystyle\int_\Omega [\mu \nabla N_i \nabla N_j + \rho N_i (\nabla \mathbf{v} N_j + \mathbf{v} \nabla N_j)]\, d\Omega & \displaystyle -\int_\Omega \nabla N_i L_j\, d\Omega \\ \displaystyle -\int_\Omega L_i \nabla N_j\, d\Omega & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{v} \\ \Delta_\mathbf{p} \end{bmatrix}$$

$$= - \begin{bmatrix} \displaystyle\int_\Omega (\mu \nabla N_i \cdot \nabla \mathbf{v} + \rho N_i \mathbf{v} \cdot \nabla \mathbf{v} - \nabla N_i p)\, d\Omega + \int_{\delta\Omega} (-\mu N_i \partial \mathbf{v} / \partial n + N_i p) d\delta\Omega \\ \displaystyle -\int_\Omega L_i \nabla \cdot \mathbf{v}\, d\Omega \end{bmatrix}, \tag{5}$$

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \Delta \mathbf{v}, \tag{6}$$

$$\mathbf{p}^{n+1} = \mathbf{p}^n + \Delta \mathbf{p}. \tag{7}$$

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 via the correction found by solving (5).

4. ADAPTIVE LINEARIZATION

An alternative or a supplement to Newton linearization of the equation system is local grid adaption to the solution, which will also contribute to the linearization of the equation system. From the analytic integrations[8] we obtain the formula

$$\frac{\int_\Omega \rho N_i \mathbf{v} \cdot \nabla \mathbf{v} \, d\Omega}{\int_\Omega \mu \nabla N_i \cdot \nabla \mathbf{v} \, d\Omega} = \frac{a(\mathbf{v}) l}{b(\mathbf{v})}. \tag{8}$$

In the above formula $a(\mathbf{v})$ and $b(\mathbf{v})$ are function only of the velocities inside the element and are independent of the element size. The length l is some characteristic length of the element. The formula shows 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. Provided that the local element size is reduced sufficiently, this implicit adaptive linearization will for many Navier–Stokes applications appear to be sufficient and satisfactory.

5. GRID ADAPTION

The Reynolds number for fluid flow is usually defined as

$$Re = \rho U_b d / \mu, \tag{9}$$

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 respectively 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

$$Re_e = \frac{\sum_i N_i^c \left\| \rho \int_\Omega N_i \mathbf{v} \cdot \nabla \mathbf{v} \, d\Omega \right\|}{\sum_i N_i^c \left\| \mu \int_\Omega \nabla N_i \cdot \nabla \mathbf{v} \, d\Omega \right\|} < \varepsilon_A, \tag{10}$$

where N_i^c is the basis function evaluated at the geometrical centre. The refinements and recoarsements of elements are decided based on the element Reynolds number, which is calculated from the nodal values and weighted with 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:

$$Re_e > \varepsilon_A. \tag{11}$$

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 solution is then projected from the finite element grid to the tri-tree grid. The element Reynolds number is computed for the tri-tree elements. First the tri-tree elements are recoarsed. During recoarsing, 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 hierarchical tree structure of the tri-tree, the recoarsement algorithm becomes recursive. At the end of the recoarsement procedure, the tri-tree contains elements with element Reynolds number 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 thus 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 the elements will be refined later if their element Reynolds number is above the refinement limit ε_A.

The multigrid solution algorithm consists of several iterative operations before a final solution is obtained. At present the most efficient way seems to be as follows.

Solve equation system for low velocities on a coarse grid
Repeat
  Increase the boundary velocity and scale the solution correspondingly
  Compute $Re_e < \varepsilon_A$ for each element using the scaled solution
  Recoarse the grid while $Re_e < \varepsilon_A$ for all elements
  Refine the grid until $Re_e < \varepsilon_A$ for all elements
  Project the coarse solution to the fine grid by linear interpolation
  Solve the equation system with the interpolated solution as start vector
Until convergence

6. CONVERGENCE CRITERIA

The tri-tree multigrid solver consists of three iterative algorithms inside each other. The inner iterative algorithm 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, which is the outer level of iterations. The three convergence criteria used are

$$\text{(i)} \quad \frac{\|\delta^{\mathrm{L}}\mathbf{x}_{\mathbf{v}}^{k}\|}{U_{\mathrm{b}}} < \varepsilon_{\mathrm{L}}, \qquad \text{(ii)} \quad \frac{\|\delta^{\mathrm{N}}\mathbf{x}_{\mathbf{v}}^{k}\|}{U_{\mathrm{b}}} < \varepsilon_{\mathrm{N}}, \qquad \text{(iii)} \quad \frac{\|\delta^{\mathrm{G}}\mathbf{x}_{\mathbf{v}}^{k}\|}{U_{\mathrm{b}}} < \varepsilon_{\mathrm{G}}, \tag{12}$$

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 velocity update of the solution in the Newton iterations and $\delta^{\mathrm{G}}\mathbf{x}_{\mathbf{v}}^{k}$ is the velocity difference between the projected start vector and the velocity solution at grid level k; ε_{L} is the linear, ε_{N} is the non-linear and ε_{G} is the grid convergence criterion.

7. EXPERIMENTAL DESIGN

The test problem is a driven cavity flow.[11] The boundary conditions for the cavity are zero velocities at three of the sides and a specified non-zero tangential velocity at the fourth side. The initial rectangular grid consists of 128 elements.[8] The start vector for all tests is the solution for this regular grid for $Re_{\mathrm{e}} = 400$. The Reynolds number is increased from 400 to 800 for velocity steps of 0·00002, 0·00004 and 0·00008, which correspond to Reynolds number steps of 20, 40 and 80. At each Reynolds number the solution from the previous Reynolds number is scaled to the present Reynolds number and the grid is adapted to this scaled solution. The experiments are repeated for different

adaption limits of the element Reynolds number, $\varepsilon_A = 0{\cdot}5, 1{\cdot}0, 2{\cdot}0$ and $4{\cdot}0$. The linear convergence criterion $\varepsilon_L = 10^{-4}$. In the Newton iterations the number of Newton iterations is fixed at five. The grid adaption iterations are performed for Reynolds number 680 with three adaption levels of the element Reynolds number, $\varepsilon_A = 0{\cdot}5, 1{\cdot}0$ and $2{\cdot}0$. In the grid iterations the boundary conditions are fixed and the grid is adapted to the obtained solution.

8. NUMERICAL RESULTS

The results of the simulations for velocity increments in steps of 0·00002 and Reynolds number increments in steps of 20 are given in Table I. The computational times for grid adaption and equation solution and the number of unknowns in the equation system are shown. The times required for both grid adaption and equation solution increase with the cavity Reynolds number, which is due to the increase in the number of elements and the number of unknowns. The times for grid adaption and solution are far less for larger adaption limits which will tolerate larger non-linearities in the equation system. Higher adaption limits results in both a smaller number of elements and a smaller number of unknowns.

In Table II the velocity step between computations is increased to 0·0004 and the Reynolds number increments is 40. The computational time for grid adaption is not changed for the computations at the same Reynolds numbers given in Table I. However, the computational time for solving the equation system is slightly larger for a larger increase in velocities, which is due to the fact that the start vector for the solution procedure is closer to the solution for small velocity increments. The number of unknowns is almost the same for the corresponding Reynolds numbers in the two experiments.

Table I. Computational time used for adapting grid to solution, time used for solving equation system and number of unknowns for different adaption levels $\varepsilon_A = 0{\cdot}5, 1{\cdot}0, 2{\cdot}0$. The cavity Reynolds number is increased in velocity steps of 0·00002 with steps in the Reynolds number of 20. The time unit in the table is seconds

Re	Time of adaption			Time of solution			No. of unknowns		
	$\varepsilon_A = 0{\cdot}5$	$\varepsilon_A = 1{\cdot}0$	$\varepsilon_A = 2{\cdot}0$	$\varepsilon_A = 0{\cdot}5$	$\varepsilon_A = 1{\cdot}0$	$\varepsilon_A = 2{\cdot}0$	$\varepsilon_A = 0{\cdot}5$	$\varepsilon_A = 1{\cdot}0$	$\varepsilon_A = 2{\cdot}0$
400	59	13	3	1116	402	71	22409	6398	2020
420	121	29	6	1520	500	93	36224	10378	2411
440	126	33	7	1684	409	104	39566	10953	2929
460	142	33	8	1673	397	147	43429	11286	2985
480	152	35	8	1727	465	152	46505	12245	3530
500	163	38	10	1749	471	161	50279	12999	3541
520	177	40	10	1905	514	156	53771	13733	3951
540	190	43	11	2482	546	159	57735	14901	3692
560	208	47	11	2192	546	189	62635	15918	4120
580	227	51	12	2452	605	198	68985	17239	4350
600	246	55	13	2887	710	203	73677	18754	4950
620	263	59	14	2800	722	272	77754	20010	4846
640	280	63	14	2919	713	271	82711	20973	5670
660	300	66	16	3182	754	309	88111	22301	5772
680	320	72	18	3230	941	303	93433	23419	6642
700	342	75	22	3805	1053	582	99252	24522	7645
720	368	79	23	4060	865	590	106365	25874	6932
740	397	85	23	5832	968	475	113876	27965	8238
760	441	92	27	4930	1737	410	124262	29342	9367
780	465	103	29	5118	1535	409	131109	33951	10385
800	504	119	31	5561	1456	412	141446	36903	10079

Table II. Computational time used for adapting grid to solution, time used for solving equation system and number of unknowns for different adaption levels $\varepsilon_A = 0{\cdot}5, 1{\cdot}0, 2{\cdot}0$. The cavity Reynolds number is increased in velocity steps of 0·00004 with steps in the Reynolds number of 40. The time unit in the table is seconds

Re	Time of adaption			Time of solution			No. of unknowns		
	$\varepsilon_A = 0{\cdot}5$	$\varepsilon_A = 1{\cdot}0$	$\varepsilon_A = 2{\cdot}0$	$\varepsilon_A = 0{\cdot}5$	$\varepsilon_A = 1{\cdot}0$	$\varepsilon_A = 2{\cdot}0$	$\varepsilon_A = 0{\cdot}5$	$\varepsilon_A = 1{\cdot}0$	$\varepsilon_A = 2{\cdot}0$
400	59	14	3	1116	405	71	22409	6398	2020
440	130	32	6	1692	498	107	39019	11170	2841
380	151	37	8	2180	475	175	46199	12938	3307
520	175	40	9	1981	461	200	53326	13692	3969
560	203	46	11	3750	658	199	61776	15837	4221
600	242	53	12	3189	694	253	73308	18675	4769
640	281	62	14	3625	745	277	82749	21088	5429
680	320	69	16	3521	907	426	93496	23383	5862
720	366	78	22	4107	1039	341	105730	25960	8382
760	420	88	27	5226	1158	542	120327	29346	9563
800	493	99	62	6280	2100	1229	138920	33213	20228

Table III. Computational time used for adapting grid to solution, time used for solving equation system and number of unknowns for different adaption levels $\varepsilon_A = 0{\cdot}5, 1{\cdot}0, 2{\cdot}0$. The cavity Reynolds number is increased in velocity steps of 0·00008 with steps in the Reynolds number of 80. The time unit in the table is seconds

Re	Time of adaption			Time of solution			No. of unknowns		
	$\varepsilon_A = 0{\cdot}5$	$\varepsilon_A = 1{\cdot}0$	$\varepsilon_A = 2{\cdot}0$	$\varepsilon_A = 0{\cdot}5$	$\varepsilon_A = 1{\cdot}0$	$\varepsilon_A = 2{\cdot}0$	$\varepsilon_A = 0{\cdot}5$	$\varepsilon_A = 1{\cdot}0$	$\varepsilon_A = 2{\cdot}0$
400	59	13	3	1116	402	71	22409	6398	2020
480	153	37	8	2128	788	127	46249	13103	3352
560	211	50	11	2549	664	162	63593	17318	3971
640	270	62	13	3832	840	281	81710	21322	5395
720	362	78	18	6653	1063	509	104408	25748	7146
800	492	98	28	9835	1488	560	138620	32562	10263

The results of the computations for a larger increment in the velocity between each computation are given in Table III. The computational time and the number of unknowns are almost the same as for the two previous experiments. A reduction in the computational time for solving the equation system is more pronounced for this velocity increment of 0·00008 than for the lower velocity increments previously shown in Table I and II. The results of these experiments show that the computer time used in the adaptation procedure is very small compared with the time used in obtaining a solution.

The results of the grid iterations for Reynolds number 680 and element Reynolds number refinement criterial $\varepsilon_A = 0{\cdot}5, 1{\cdot}0$ and $2{\cdot}0$ are given in Tables IV and V. Table IV shows the number of unknowns, the number of linear iterations in five Newton iterations and the error estimate ε_N, in the Newton iterations. For all three adaption levels the number of unknowns in the equation system has converged to the same value in grid iterations 5 and 6. The number of linear iterations in the five Newton iterations has also been considerably reduced in the last five grid iterations compared with the number of linear iterations executed in the first grid iteration. The error estimates ε_N in the Newton iterations are all of the same order of magnitude, 10^{-3}.

Table V shows the number of finite elements, the number of tri-tree refinements and the grid error estimate ε_G of the solution. The number of finite elements has reached a constant value in grid

Table IV. Number of unknowns, number of linear iterations and error estimate of solution, ε_N, with respect to Newton iterations. The number of Newton iterations is fixed at five. The cavity Reynolds number is 680 for the different adaption levels $\varepsilon_A = 0{\cdot}5, 1{\cdot}0, 2{\cdot}0$

Level	No. of unknowns			No. of linear iterations			Newton error ε_N		
	$\varepsilon_A = 0{\cdot}5$	$\varepsilon_A = 1{\cdot}0$	$\varepsilon_A = 2{\cdot}0$	$\varepsilon_A = 0{\cdot}5$	$\varepsilon_A = 1{\cdot}0$	$\varepsilon_A = 2{\cdot}0$	$\varepsilon_A = 0{\cdot}5$	$\varepsilon_A = 1{\cdot}0$	$\varepsilon_A = 2{\cdot}0$
1	93496	23383	5862	105	133	527	0·0012	0·0027	0·0095
2	97996	24504	7819	56	59	155	0·0005	0·0025	0·0076
3	98338	24693	7891	50	103	134	0·0022	0·0016	0·0098
4	98482	24693	7972	26	98	211	0·0012	0·0028	0·0127
5	98561	24729	8044	34	61	127	0·0007	0·0013	0·0080
6	98561	24729	8044	27	58	187	0·0008	0·0015	0·0172

Table V. Number of finite elements, number of tri-tree refinements and error estimate of solution, ε_G, with respect to grid level iterations. The number of Newton iterations is fixed at five. The cavity Reynolds number is 680 for the different adaption levels $\varepsilon_A = 0{\cdot}5, 1{\cdot}0, 2{\cdot}0$.

Level	No. of finite elements			No. of tri-tree refinements			Grid error ε_G		
	$\varepsilon_A = 0{\cdot}5$	$\varepsilon_A = 1{\cdot}0$	$\varepsilon_A = 2{\cdot}0$	$\varepsilon_A = 0{\cdot}5$	$\varepsilon_A = 1{\cdot}0$	$\varepsilon_A = 2{\cdot}0$	$\varepsilon_A = 0{\cdot}5$	$\varepsilon_A = 1{\cdot}0$	$\varepsilon_A = 2{\cdot}0$
1	20719	5155	1272	1022	266	93	0·1308	0·4273	0·3225
2	21719	5403	1703	186	39	78	0·0203	0·0204	0·1173
3	21795	5445	1719	11	5	3	0·0136	0·0033	0·0918
4	21827	5445	1737	5	0	2	0·0015	0·0009	0·0121
5	21844	5453	1753	2	1	3	0·0212	0·0016	0·0099
6	21844	5453	1753	0	0	0	0·0009	0·0016	0·0030

iterations 5 and 6. The number of corresponding tri-tree refinements has accordingly converged to zero. The grid error estimate of the solution, ε_G, has reached the level of 10^{-3}.

Figure 1 shows the velocity vectors and isobars for Reynolds number 800, computed with velocity increments of 0·00002 between grid levels and with an adaption limit $\varepsilon_A = 1{\cdot}0$. The velocity vectors indicate the locations of high velocities and also very roughly indicate the areas with large convection. The isobars indicate high pressure gradients at the distal wall of the specified boundary velocity.

Figure 2 shows the grid generated for Reynolds numbers 400, 480, 560, 640, 720 and 800 with adaption level $\varepsilon_A = 2{\cdot}0$. The smallest elements are found where the largest element Reynolds numbers are expected to occur. The grid becomes finer with increasing cavity Reynolds number. Close inspection of the grids reveal that the elements with the highest element Reynolds numbers are contained in a narrow belt which moves towards the cavity walls with increasing cavity Reynolds number.

Figure 3 shows the grids for the same cavity Reynolds numbers as in Figure 2 but for a smaller adaption level $\varepsilon_A = 1{\cdot}0$. The number of elements for each grid is increased compared with the corresponding grids in Figure 2. In addition to a concentration of fine elements where the local Reynolds number is large, a concentration of finite elements occurs where high pressure gradients are found.

Figure 4 shows the grids for the same cavity Reynolds numbers as in Figures 2 and 3 but for the adaption level $\varepsilon_A = 0{\cdot}5$. The grids consist of more smaller elements than for the other two adaption levels. In this figure a more distinct local high-Reynolds-number zone occurs.

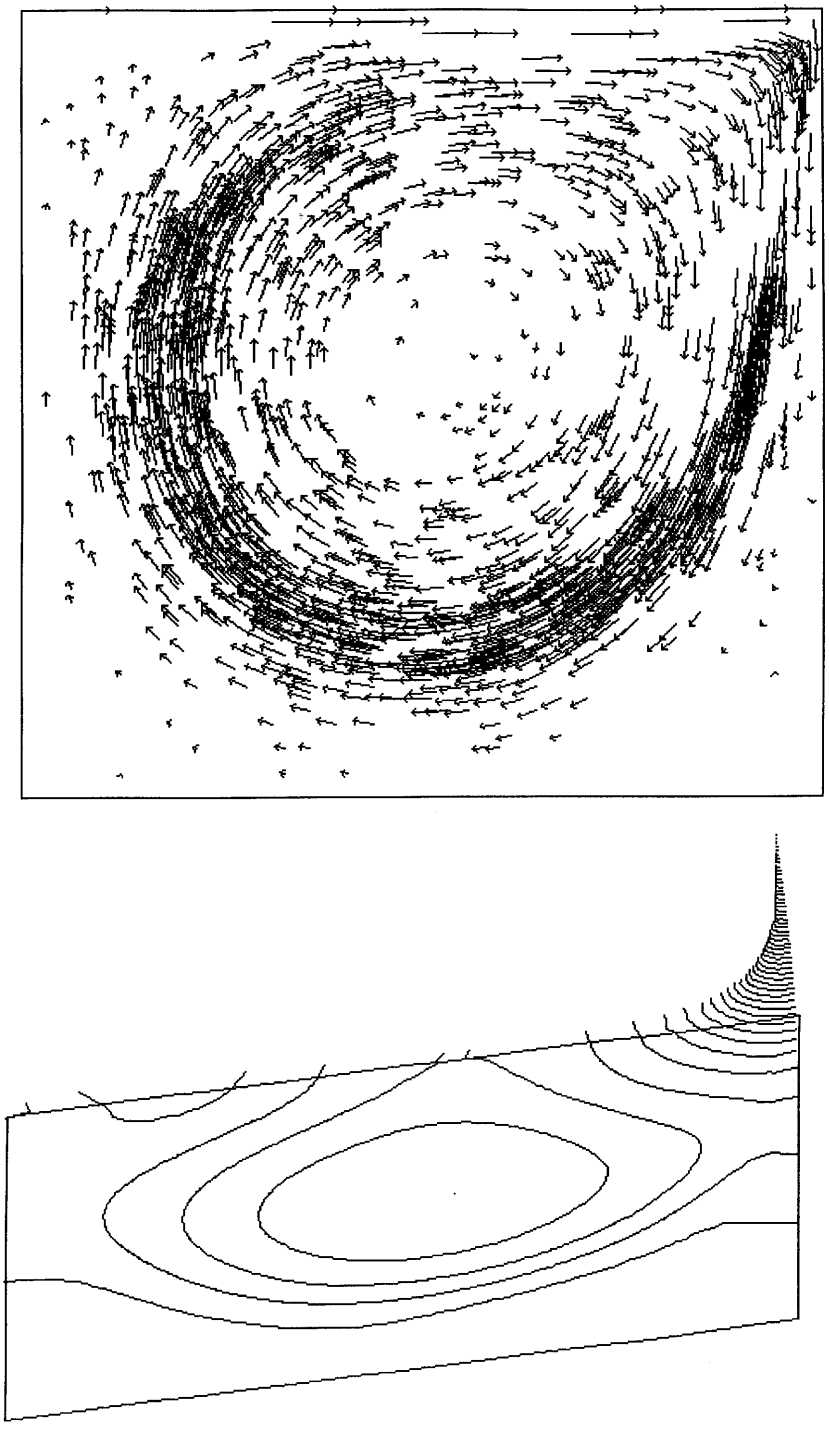

Figure 1. Pressure and flow field for cavity flow. The upper part shows the flow velocity and the lower part shows the isobars. The cavity Reynolds number is 800 and the adaption levle is 1·0

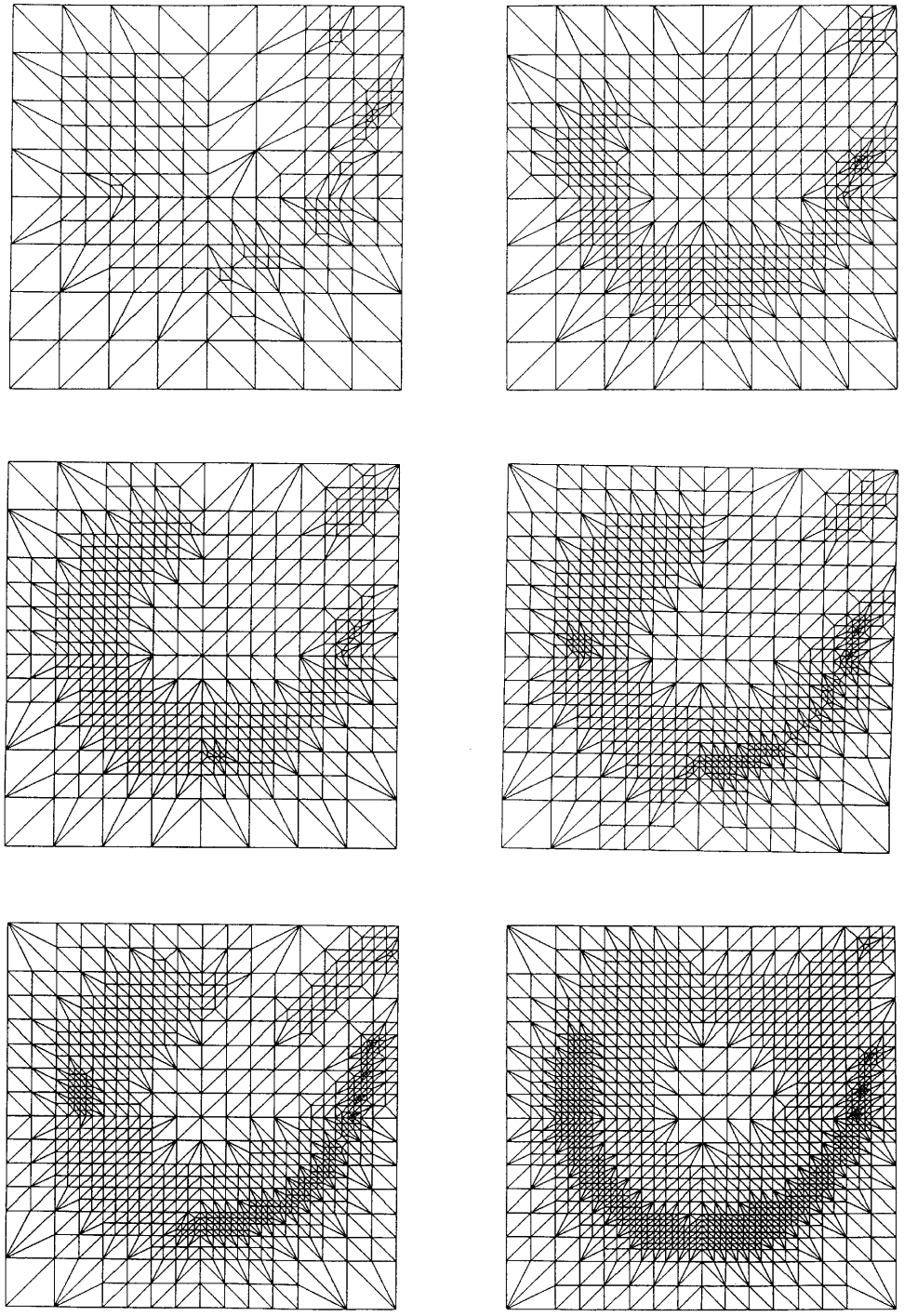

Figure 2. Grids generated by tri-tree algorithm at selected Reynolds numbers. The adaption level is $\epsilon_A = 2{\cdot}0$. The cavity Reynolds numbers from the upper left are $Re = 400$, 480, 560, 640, 720, 800. The Reynolds number between grids is increased by increasing the velocity in steps of 0·00002

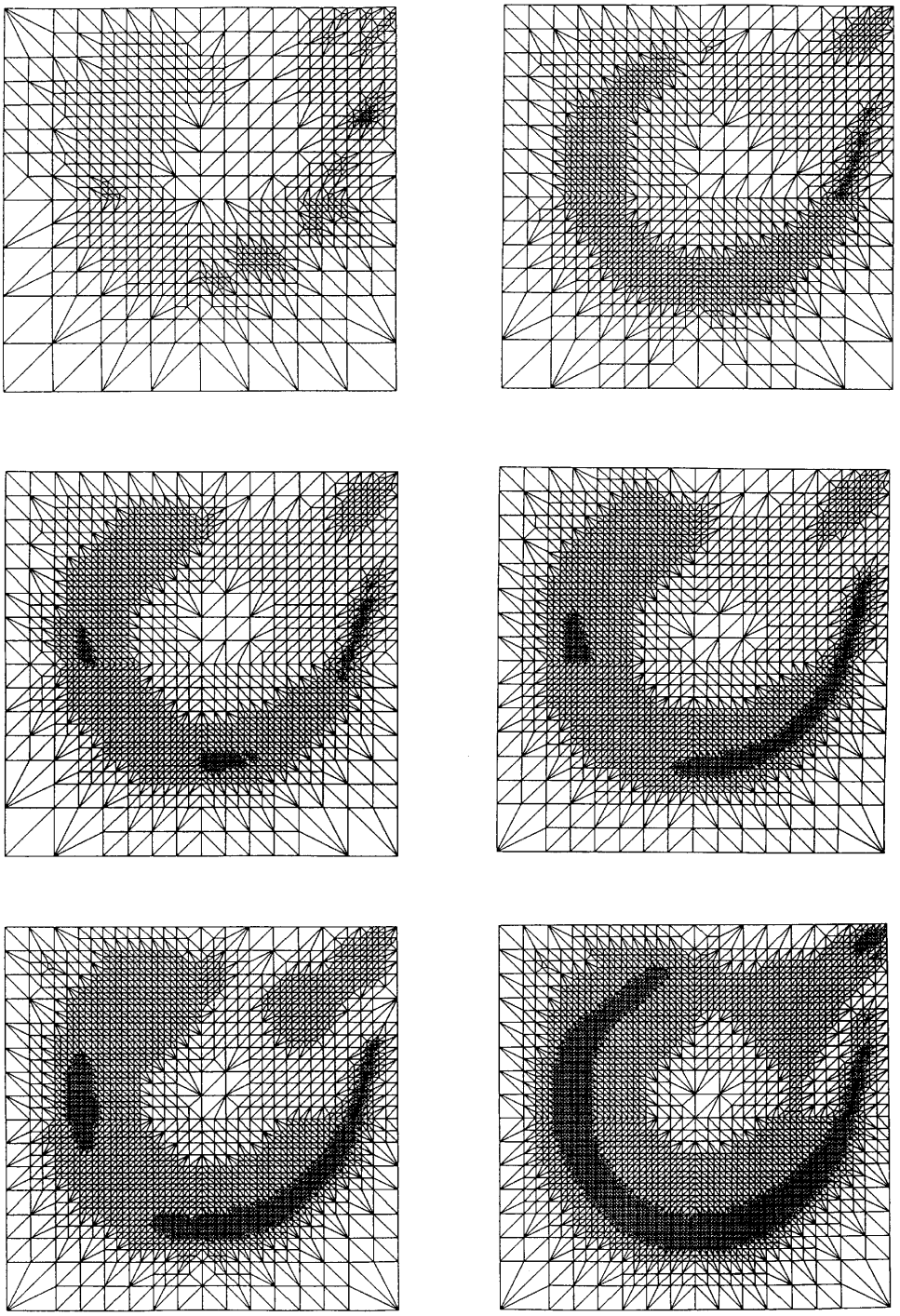

Figure 3. Grids generated by tri-tree algorithm at selected Reynolds numbers. The adaption level is $\epsilon_A = 1{\cdot}0$. The cavity Reynolds numbers from the upper left are $Re = 400$, 480, 560, 640, 720, 800. The Reynolds number between grids is increased by increasing the velocity in steps of 0·00002

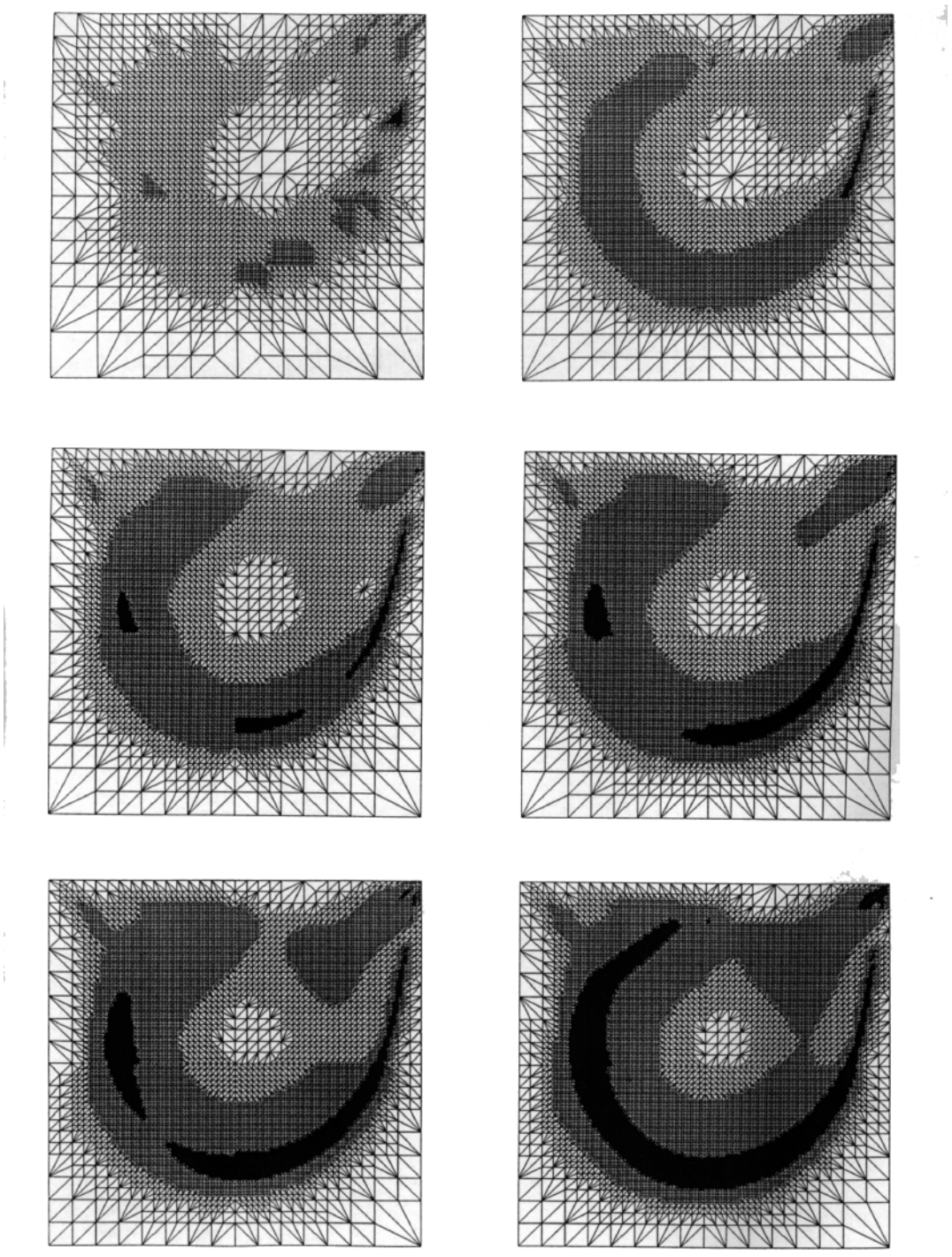

Figure 4. Grids generated by tri-tree lgorithm at selected Reynolds numbers. The adaption level is $\epsilon_A = 0{\cdot}5$. The cavity Reynolds numbers from the upper left are $Re = 400$, 480, 560, 640, 720, 800. The Reynolds number between grids is increased by increasing the velocity in steps of 0·00002

9. DISCUSSION

The Navier–Stokes equations may be considered as a combination of the linear Stokes equations and the non-linear Euler equations. In order to solve the non-linear Navier–Stokes equation system, the equations should be linearized in some way. The most common linearization method is the Newton method with known second-order convergence rate. In the present work an additional linearization method is applied in a structured way. The linearization of the equation system is performed by grid adaption at the specific tri-tree elements where the solutions of the equation system indicate high convection compared with diffusion. The present linearization technique therefore consists of a global Newton linearization which is applied to the equation system in the entire computational domain and a grid adaption linearization which is applied locally to high-convection areas.

In previous work[10] it is shown that the element Reynolds number has to be smaller than 10 for all elements in the grid in order to obtain a convergent solution of the Navier–Stokes equations using the CGSTAB linear equation solver preconditioned with incomplete LU factorization with coupled node fill-in.[7] The present work shows that the computer time used by the equation solver is proportional to the number of unknowns and that the degree of non-linearity plays a minor role as long as the element Reynolds number is below a reasonable limit, less than 10.

While linear and non-linear convergence in the solution procedure for differential equations has been established as an important property of the solution method, little attention has been paid to the convergence of the solution with respect to grid adaption. The present work outlines a method of grid adaption to the solution as well as a grid convergence estimator.

The non-linearities of the implicit equation system of the finite element formulation of the Navier–Stokes equations can be reduced by the tri-tree grid adaption algorithm. The magnitude of the non-linear terms is governed by the element Reynolds number. As the non-linearities in the equation matrix are reduced, the equation system becomes more symmetric. The non-linear terms will appear on the right-hand side of the equation system.

ACKNOWLEDGEMENTS

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

REFERENCES

1. G. S. Barussi, W. G. Habashi and J. G. Guevremont, ‘A second-order finite element for the solution of the transonic Euler and Navier–Stokes equations’, *Int. j. numer. methods fluids*, **20**, 671–693 (1995).
2. D. L. Hill and E. A. Baskharone, ‘A multiblock Navier–Stokes algorithm using equal-order quadratic finite elements’, *Int. j. numer. methods fluids*, **20**, 169–185 (1995).
3. Y. Zang and R. L. Street, ‘A composite multigrid method for calculating unsteady incompressible flows in geometrically complex domains’, *Int. j. numer. methods fluids*, **20**, 341–361 (1995).
4. A. A. Lorber and G. F. Carey, ‘A vector–parallel scheme for Navier–Stokes computations at multi-gigaflop performance rates’, *Int. j. number. methods fluids*, **21**, 445–466 (1995).
5. Y. Kallinderis, ‘Adaptive hybrid prismatic/tetrahedral grids’, *Int. j. numer. methods fluids*, **20**, 1023–1037 (1992).
6. 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).
7. 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 2-D and 3-D Navier–Stokes equations’, *Int. j. numer. methods fluids*, **15**, 525–544 (1992).
8. S. Ø. Wille, ‘A non-linear adaptive tri-tree multigrid solver for mixed finite element formulation of the Navier–Stokes equations’, *Int. j. number. methods fluids*, **22**, 1041–1059 (1996).
9. S. Ø. Wille, ‘A local solution adapted tri-tree multigrid generator and equation solver for mixed finite element formulation of the Navier–Stokes equations’, *Comput. Methods Appl. Mech. Eng.*, **131**, 109–132 (1996).

10. 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).
11. S. Ø. Wille, 'A tri-tree recoarsement algorithm for the finite element formulation of the Navier–Stokes equations', *Comput. Methods Appl. Mech. Eng.*, **135**, 129–142 (1996).
12. C. Taylor and P. Hood, 'A numerical solution of the Navier–Stokes equations using the finite element technique', *Comput. Fluids*, **1**, 73–100 (1973).