The influence of the order of fill-in on the convergence rate for ILU preconditioned iterative solvers

S.Ø. Wille
Faculty of Engineering, Oslo University College, Oslo, Norway,
Laboratório Nacional de Computação Cientifica, Petropolis, Brazil,
ASE-EM, TICAM, The University of Texas at Austin, USA
Ø. Staff
Faculty of Engineering, Oslo University College, Oslo, Norway
A.F.D. Loula
Laboratório Nacional de Computação Cientifica, Petropolis, Brazil
G.F. Carey
ASE-EM, TICAM, The University of Texas at Austin, USA

Received October 2002
Revised January 2003
Accepted April 2003

Keywords *Variational techniques, Operations management, Iterative methods, Finite element analysis*

Abstract *The indefinite nature of the mixed finite element formulation of the Navier-Stokes equations is treated by segregation of the variables. The segregation algorithm assembles the coefficients which correspond to the velocity variables in the upper part of the equation matrix and the coefficients which corresponds to the pressure variables in the lower part of the equation matrix. During the incomplete; elimination of the velocity matrix, fill-in will occur in the pressure matrix, hence, divisions with zero are avoided. The fill-in rule applied here is related to the location of the node in the finite element mesh, rather than the magnitude of the fill-in or the magnitude of the coefficient at the location of the fill-in. Different orders of fill-in are explored for ILU preconditioning of the mixed finite element formulation of the Navier-Stokes equations in two dimensions.*

1. Introduction

The use of direct equation solvers for large scale finite element simulation has been limited owing to the requirement of large memory storage capacity. Consequently, the efforts have been directed towards the development of fast iterative equation solvers requiring less memory storage and CPU time (Meijerink and van der Vorst, 1977; Sonneveld, 1987). In some algorithms,

This research was supported in part by the CNPq (Conselho Nacional de Desenvolvimento Cientúfico e Tecnológico, Brazil), Grant No. 300520/98-0, the Norwegian Research Council, Grant No. 131 523/410 and the National Science Foundation, NSF Contract # ACI-0081791.

International Journal of Numerical Methods for Heat & Fluid Flow
Vol. 14 No. 3, 2004
pp. 325-340

0961-5539
DOI 10.1108/09615530410517986

equation matrix need not be stored at all as in element-by-element formulations (Wille, 1986, 1987). However, iterative schemes may not be robust, and good preconditioners may be needed.

Hence, the preconditioning of iterative solvers has been the subject of extensive investigation in recent times (Carey *et al.*, 1989; Dahl and Wille, 1992; van der Vorst, 1992; Young and Yea, 1980). The most popular preconditioner has been incomplete Gaussian factorization (ILU). As demonstrated in our numerical experiments, the efficiency of ILU preconditioning for a finite element equation is strongly related to the node numbering of the finite element mesh. There are several efficient ways of node numbering. The nodes may be numbered from the boundary of the element mesh towards the center of the mesh, from one side of the mesh to the opposite one or from the center of the mesh towards the boundary, to minimize bandwidth or frontwidth, in accordance with nested dissection criteria. In the present context, the influence of ordering on fill-in for ILU preconditioning is of particular interest.

In the solution of algebraic equations, the amount of fill-in is governed by a predefined limiting value for the coefficients of the magnitude of the fill-in. The greater the magnitude of this limit is set, the higher order of fill-in is accepted. This fill-in rule is not very suitable in practice, especially when the system of finite element equations is indefinite as, for example, the Navier-Stokes system.

In the present work, a fill-in rule based on the location of the nodes in the element mesh is developed. First-order fill-in is defined as follows: if nodes belong to the same element, fill-in is confined to the corresponding locations in the equation matrix. If nodes belong to an element or adjacent elements, second-order fill-in is accomplished.

The finite element equations for incompressible flow with mixed interpolation are indefinite in nature. Therefore, care has to be taken when assembling the equation matrix to avoid division by zero during the factorization of the equation matrix. In the present algorithm, the segregation of variable method is applied to the Navier-Stokes system. The pressure equations are assembled in the last diagonal block of the matrix. During the elimination process, fill-in will occur on the diagonal and in other locations, which were initially zero due to indefiniteness.

In the present work the fill-in algorithm is implemented and numerical experiments are performed for an ILU preconditioned conjugate gradient algorithm for solving the global system of equations. The experiments are performed for increasing the order of fill-in and various Reynolds numbers with meshes are adapted to the solution (Greaves and Borthwick, 1998, 1999; Kallinderis, 1992; Wille, 1992, 1996).

Dahl and Wille (1992) compared the present fill-in rule with the first-order fill-in described in the work by Carey *et al.* (1989), where a small number was

added to the zero diagonal entries. This comparison was favorable with respect to the present fill-in algorithm.

The effect of adapted meshes for the the first-order fill-in algorithm in solving the Navier-Stokes equations has been studied by Wille (1992). The adapted mesh algorithm has also been applied to other problems, such as surface flow in the North-Sea (Wille, 1998).

2. The equations

As a model problem, we consider the stationary Navier-Stokes system in two dimensions

$$\rho \mathbf{u} \cdot \nabla \mathbf{u} - \mu \nabla^2 \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega \subset R^2 \tag{1}$$

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

with homogeneous Dirichlet boundary conditions

$$\mathbf{u} = 0 \quad \text{on } \Gamma = \partial\Omega \tag{3}$$

where $\mathbf{u}$ is the velocity vector, p the pressure, ρ the density and μ the viscosity coefficient.

For small $|\mathbf{u}|$ this can be approximated by

$$-\mu \nabla^2 \mathbf{u} + \nabla p = \mathbf{f} \tag{4}$$

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

as a particular case of the stationary Navier-Stokes system without convection.

A variational formulation of the Navier-Stokes system: find the velocity $\mathbf{u} \in U = H_0^1(\Omega) \times H_0^1(\Omega)$ and the pressure $p \in Q = L^2(\Omega)/R$ such that

$$\tilde{a}(\mathbf{u}, \mathbf{u}, \mathbf{v}) + b(p, \mathbf{v}) = f(\mathbf{v}) \quad \forall \mathbf{v} \in U \tag{6}$$

$$b(q, \mathbf{u}) = 0 \quad \forall q \in Q \tag{7}$$

with

$$a(\mathbf{u}, \mathbf{u}, \mathbf{v}) = \int_\Omega [\mathbf{u} \cdot \nabla \mathbf{u} \cdot \mathbf{v} + \mu \nabla \mathbf{u} \cdot \nabla \mathbf{v} - p \nabla \cdot \mathbf{v}] \, \mathrm{d}\Omega \tag{8}$$

$$b(q, \mathbf{u}) = -\int_\Omega \nabla \cdot \mathbf{u} q \, \mathrm{d}\Omega \tag{9}$$

$$f(\mathbf{v}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, \mathrm{d}\Omega \tag{10}$$

Let us consider the standard mixed finite element formulation with both velocities and pressure approximated using C^0 Lagrangian interpolations with quadratic basis functions for velocities and linear basis functions for pressure on triangles (Taylor and Hood, 1973). Denoting $U_h \subset U$ and $Q_h \subset Q$, as the corresponding C^0 Lagragian finite element spaces for velocity and pressure fields, the Galerkin approximation for the Navier-Stokes equation system becomes

$$a(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) + b(p_h, \mathbf{v}_h) = f(\mathbf{v}_h) \quad \forall \mathbf{v}_h \in U_h \tag{11}$$

$$b(q_h, \mathbf{u}_h) = 0 \quad \forall q_h \in Q_h \tag{12}$$

The Stokes approximation is recovered by omitting the convective term in equation (11), yielding

$$a(\mathbf{u}_h, \mathbf{v}_h) + b(p, \mathbf{v}_h) = f(\mathbf{v}_h) \quad \forall \mathbf{v}_h \in U_h \tag{13}$$

$$b(q_h, \mathbf{u}_h) = 0 \quad \forall q_h \in Q_h \tag{14}$$

with

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} [\mu \nabla \mathbf{u} \cdot \nabla \mathbf{v} - p \nabla \cdot \mathbf{v}] \, \mathrm{d}\Omega \tag{15}$$

3. Pivoting of nodes

A typical mesh and node ordering as delivered by a standard mesh generator is shown in Figure 1 (left). This mesh does not have an optimal node ordering and

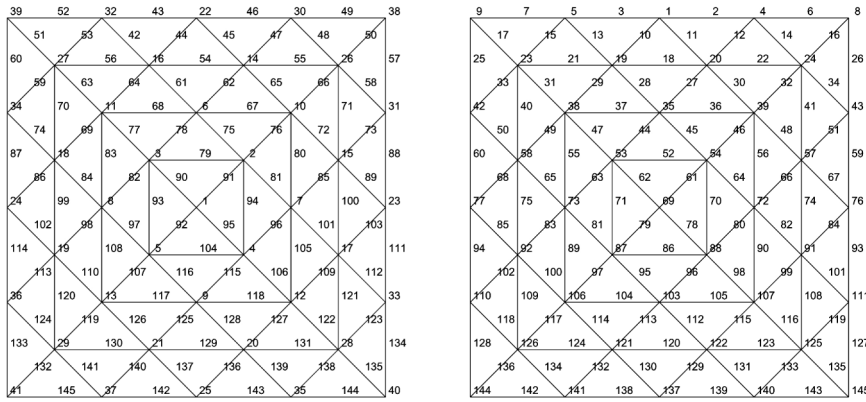


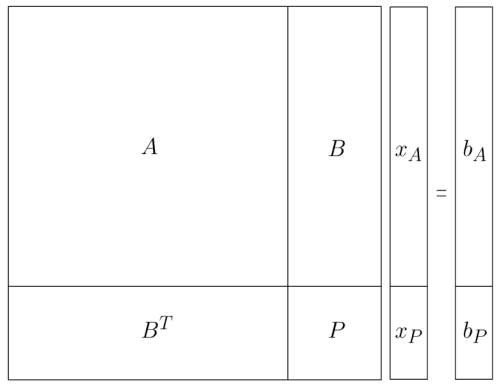
Figure 1.
Node ordering given by a standard generator, for example, oct tree mesh generators and Delauney triangulation (left). The node ordering after sorting the nodes with respect to distance to a point far away in the y-direction (right)

results in a less efficient ILU preconditioner (Dahl and Wille, 1992). The node ordering using the numbering strategy used in the work of Wille and Loula (2002), is shown in Figure 1 (right). The nodes in this mesh are sorted with respect to the distance to a point far away in the y-direction. Although not proved theoretically, numerous experiments have shown that this is one of the best node ordering schemes to obtain an efficient preconditioner (Wille and Loula, 2002). The ILU factorization in this order ensures that contributions of fill-in are propagated progressively from one end of the mesh to the other.

4. Segregation of variables

The segregation of variables method has been described earlier by Dahl and Willie (1992) and Willie and Loula (2002). Applying this method to the indefinite systems of finite element equations, the variables associated with the zero diagonal entries are assembled as the last unknowns in the equation system. Thus, when using a direct Gaussian solver or an incomplete Gaussian factorization, the elimination of the variables prior to the indefinite variables will cause fill-in at the pivot locations in the matrix which were initially zero due to indefiniteness. In the viscous flow problem, the absence of pressure in the continuity equation implies zero diagonal block in the assembled system.

The block structure of the assembled finite element matrix under this ordering of degrees of freedom is shown in Figure 2. The coefficients for the velocities are assembled in matrix **A**. The coupling between the velocities and pressures appear in matrix **B** and the continuity equation is assembled in $\mathbf{B}^{\mathrm{T}}$. The matrix for the pressure degrees of freedom **P** is initially zero. However, during the factorization of the velocity matrix, **A**, fill-in will appear in the pressure matrix **P**. When the factorization reaches **P**, the fill-in evolving from the factorization of **A**, will prevent division by zero in the factorization of **P**.



Note: The matrix corresponding to the velocity degrees of freedom is contained in **A**. The coupling between velocities and pressure is via the **B** matrix. The continuity equation is assembled in $\mathbf{B}^{\mathbf{T}}$. The pressure diagonal block sub matrix **P** is initially zero

Figure 2.
Block partitioning of the matrix with the segregation of variables technique

5. Fill-in rules

The fill-in rule for ILU preconditioners is accepted at locations in the equation matrix where the magnitude of the coefficients is above a predefined limit. The order of fill-in is determined by the magnitude of this limit. However, this fill-in rule is not well suited for indefinite finite element equations. Considering the Navier-Stokes equations with mixed interpolation, this fill-in rule would never allow fill-in at the location corresponding to the zeros due to the absence of pressure in the continuity equation. A new fill-in rule for finite element equations has to be considered. The fill-in rules for finite elements considered in the present investigation are as follows.

- *First-order fill-in*. Fill-in is accepted only at locations in the global matrix where the nodes belong to the same element.
- *Second-order fill-in*. Fill-in is accepted at locations for the first-order fill-in and for locations in the global matrix corresponding to couplings to nodes in adjacent elements which caused first-order fill-in.
- *N-order fill-in*. Fill-in is accepted at locations for the $(N-1)$ order fill-in and for locations in the global matrix corresponding to couplings to nodes in adjacent elements which caused $(N-1)$ order fill-in.

By applying the above-mentioned fill-in rules, the desired fill-in for the pressure block diagonal submatrix will take place. Thus, if nodal numbering and thereby also the Gaussian elimination order is chosen carefully, the zeros at the pressure locations in the continuity equation will not introduce problems during the incomplete elimination.

Figures 3-5 show the coupling between nodes for the node at the lower left corner, the mid side node on the upper edge and the center node, respectively. The fill-in will take place at the locations of the corresponding degrees of freedom in the equation matrix. Full coupling with all nodes for the lower left corner node appear for eighth-order fill-in (Figure 3). The mid side node at the upper edge achieves full coupling for the sixth-order fill-in (Figure 4), while full coupling for the center node occur for the fourth-order fill-in (Figure 5).

6. Numerical methods

To solve the non-linear Navier-Stokes finite element equation system (11) we use Newton's method, which is known to have a second-order convergence rate.

Defining,

$$\mathbf{u}_h^{n+1} = \mathbf{u}_h^n + \Delta\mathbf{u} \tag{16}$$

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

yields the following linear incremental system to be solved at each Newton step

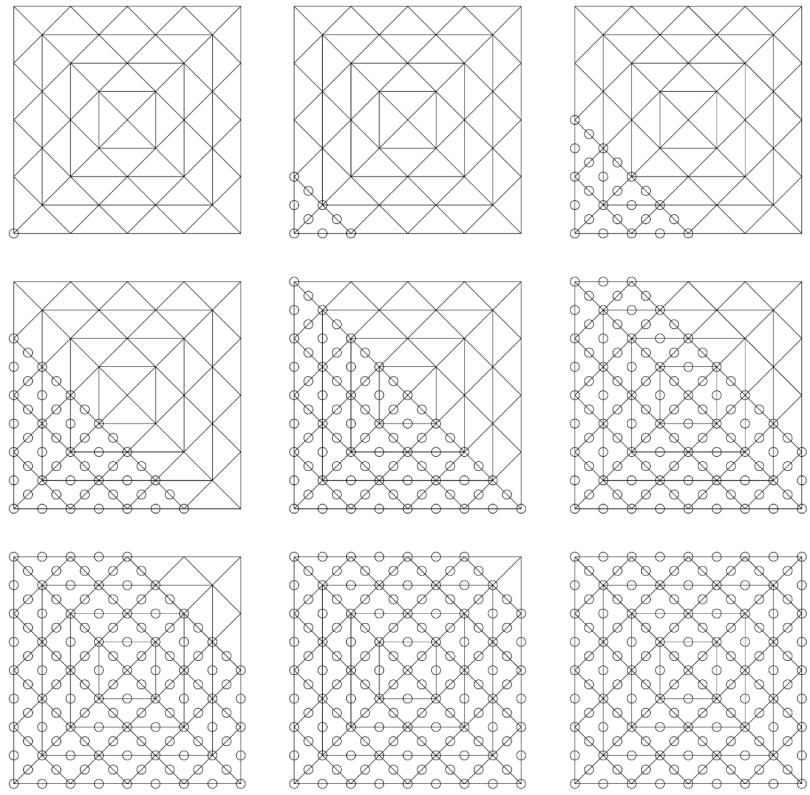


Figure 3.
Coupling of the node at the lower left corner for different orders of fill-in. The first figure to the upper left has zero-order fill-in. The last one has eight-order fill-in

$$a\left(\mathbf{u}_h^n, \Delta\mathbf{u}_h, \mathbf{v}_h\right) + a\left(\Delta\mathbf{u}_h, \mathbf{u}_h^n, \mathbf{v}_h\right) + b\left(p_h^n, \mathbf{v}_h\right) = f(\mathbf{v}_h) \quad \forall \mathbf{v}_h \in U_h \tag{18}$$

$$b(q_h, \Delta\mathbf{u}_h) = b\left(q_{\mathrm{h}}, \mathbf{u}_h^n\right) \quad \forall q_h \in Q_h \tag{19}$$

The linear nonsymmetric set of equations is solved by the ILU preconditioned Bi-CGStab algorithm (van der Vorst, 1992).

7. Numerical experiments

The influence of the fill-in order for the ILU preconditioner is explored for driven cavity flow. We solved both Stokes and Navier-Stokes equations, to investigate different fill-in orders for the symmetric and a nonsymmetric system, respectively.

The storage of the equation matrix and the preconditioning matrix is increasing with the order of fill-in. The limitation of the order of fill-in to be explored is therefore related to the computer memory. The highest order of fill-in explored with the available computer memory is the third-order fill-in.

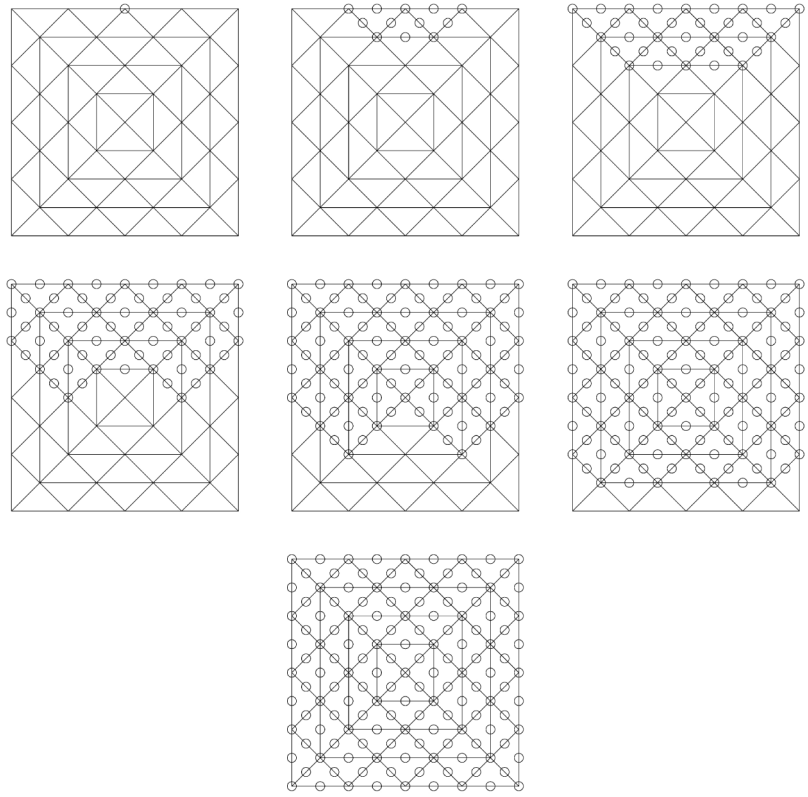


Figure 4.
Coupling of the middle node at the upper edge for different orders of fill-in. The first figure to the upper left has zero-order fill-in. The last one has sixth-order fill-in

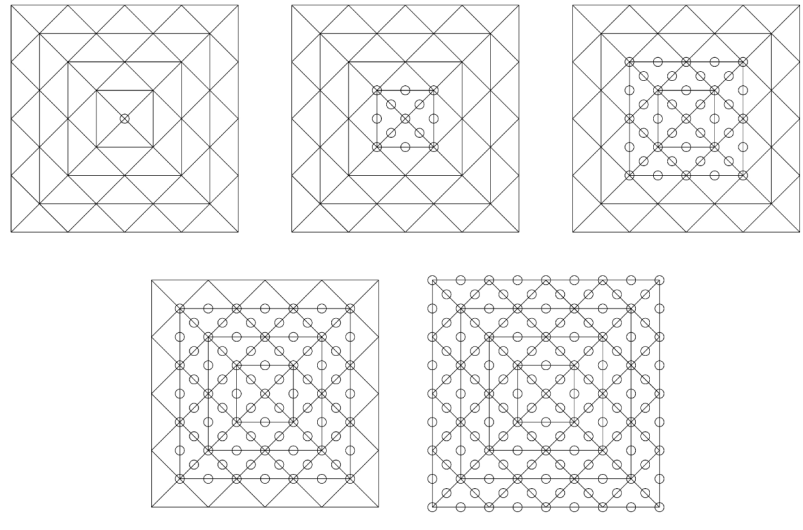


Figure 5.
Coupling of the center of the mesh for different orders of fill-in. The first picture on the upper left has zero-order fill-in. The last one has four-order fill-in

The finite element meshes used in solving the Stokes equations are shown in Figure 6 and the adapted meshes in solving the Navier-Stokes equations are shown in Figure 7. The meshes are adapted to the solution by using the ratio of convection to diffusion as refinement – recoarsening indicator (Wille, 1996).

The solutions of the Stokes equations are shown to the left and the solutions of the Navier-Stokes equations are shown to the right in Figure 8.

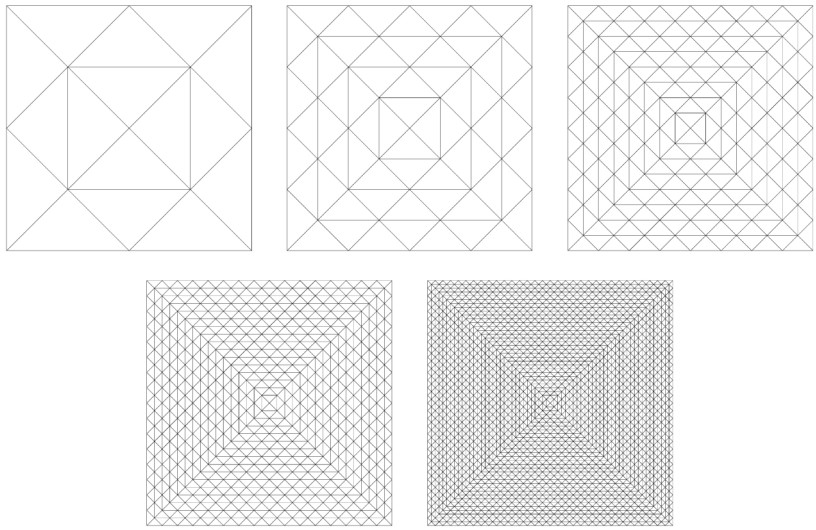


Figure 6.
Meshes used in Stokes simulations

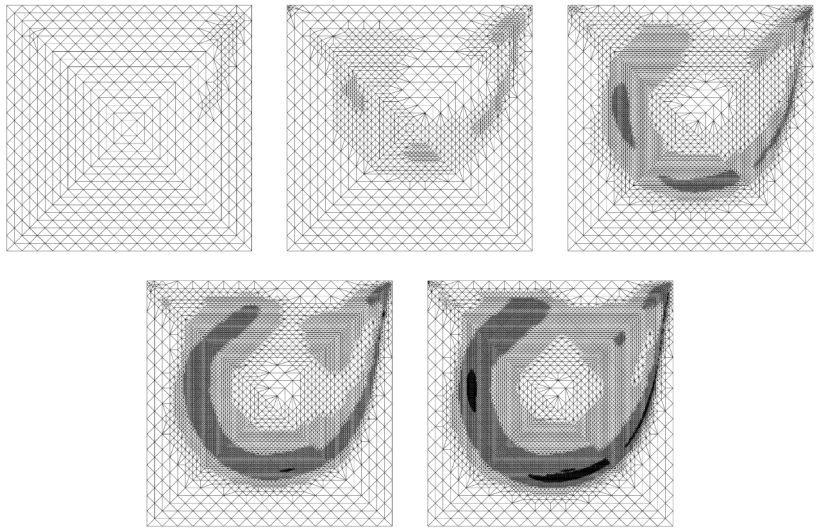


Figure 7.
Meshes for the simulations of the Navier-Stokes equations for the Reynolds number 200, 400, 600, 800, 1,000

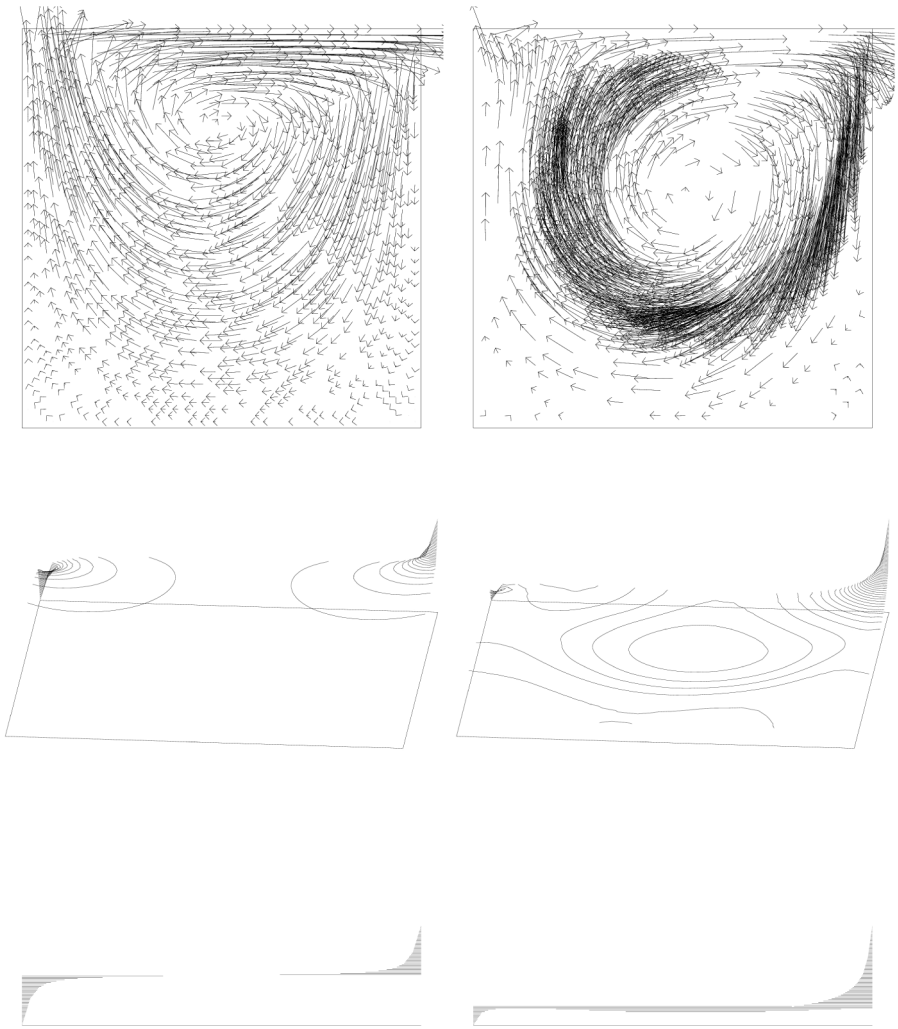


Figure 8.
Solutions of the Stokes equations, velocity vectors and isobars, to the left and the solutions of the Navier-Stokes equations for Reynolds number 500 to the right

Figure 9 shows the amount of memory storage required by the first three orders of fill-in in terms of megabytes in ILU preconditioning of the Stokes equations. The first order fill-in needs much less memory storage than the second- or third-order fill-in. The time for initialization of the different orders of fill-in (Figure 10) is the factorization time for the equation matrix, which is much less for first-order fill-in than the fill-in second- or third-order.

Figure 11 shows the number of iterations of the linear solver required to achieve convergence as a function of the number of degrees of freedom for the different orders of fill-in. The first-order fill-in requires more linear iterations

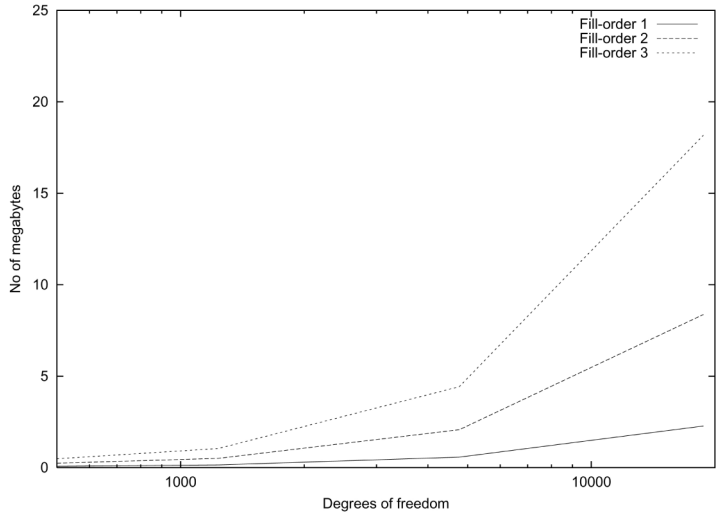


Figure 9.
The memory storage for the preconditioning matrix as a function of the number of degrees of freedom for the Stokes equations

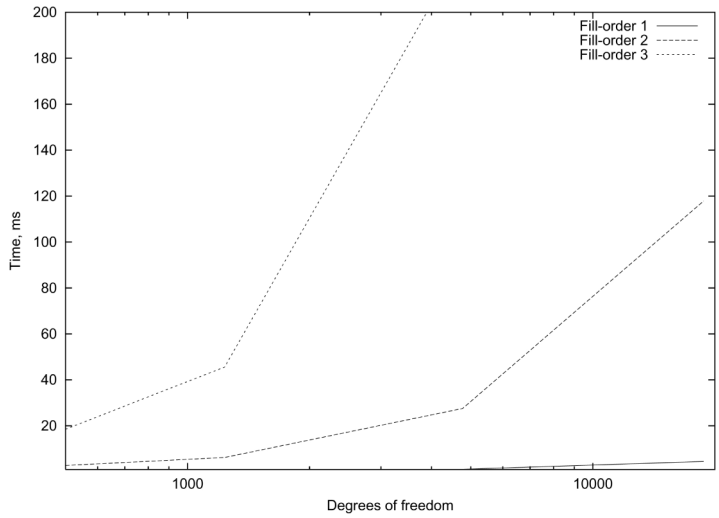


Figure 10.
The factorization time of preconditioning matrix as a function of the number of degrees of freedom for the Stokes equations

for convergence than the second- or third-order fill-in. However, as shown in Figure 12, the first-order fill-in is much faster in terms of CPU time than the second- or third-order fill-in.

The results of similar experiments with the Navier-Stokes equations are shown in Figures 13-16. In the Navier-Stokes simulations, the results are given as a function of Reynolds number. Figure 13 shows that first-order fill-in

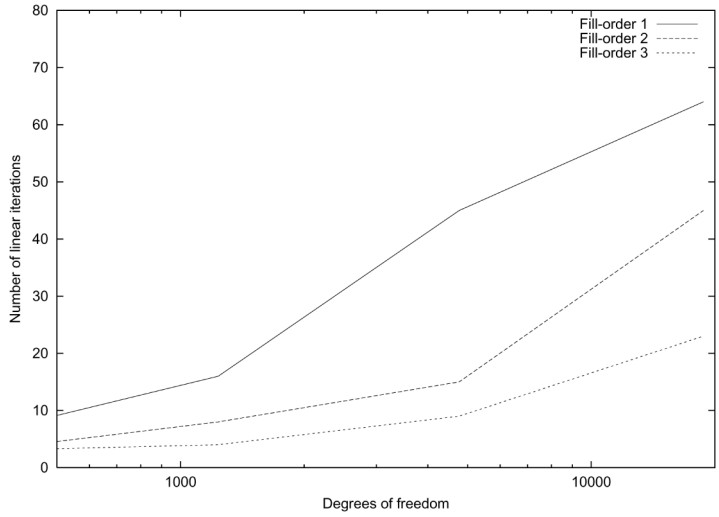


Figure 11.
The number of linear iterations for the iterative solver as a function of the number of degrees of freedom for the Stokes equations

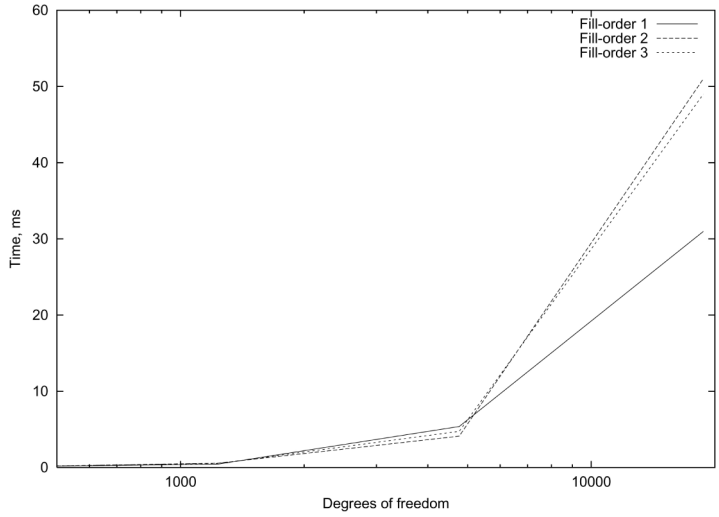


Figure 12.
The total CPU time of linear iteration as a function of the number of degrees of freedom for the Stokes equations

requires less storage for the preconditioning matrix than the second- or third-order fill-in. The factorization time (Figure 14) for the preconditioning matrix is also much less for the first-order fill-in.

The number of degrees of freedom for the adapted meshes for solving the Navier-Stokes equations, shown in Figure 7, are shown in Figure 17. Figures 15 and 16 show the number of linear iterations and the CPU time corresponding to five Newton iterations of the linear solver. Figure 15 shows that the picture

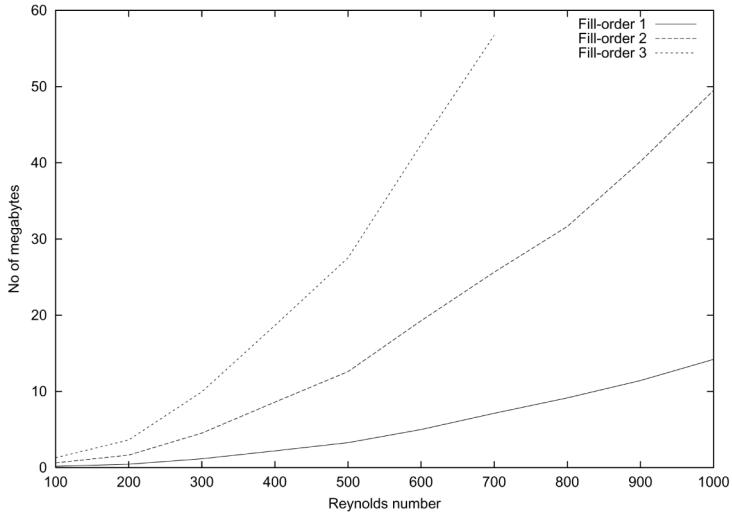


Figure 13.
The memory storage for the preconditioning matrix as a function of the Reynolds number for the Navier-Stokes equations

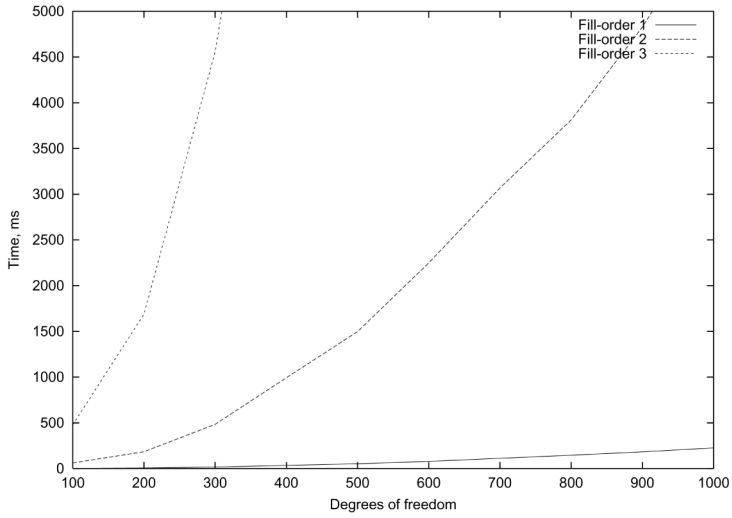


Figure 14.
The factorization time for the preconditioning matrix as a function of the Reynolds number for the Navier-Stokes equations

for the linear iterations is different for the nonlinear and nonsymmetric Navier-Stokes equations compared to the linear and symmetric Stokes equations. For the highest Reynolds numbers (Figure 15), the first-order fill-in scheme requires fewer linear iterations than the higher order fill-in schemes. Figure 16 also shows that the first-order fill-in is much faster than the second- or third-order fill-in.

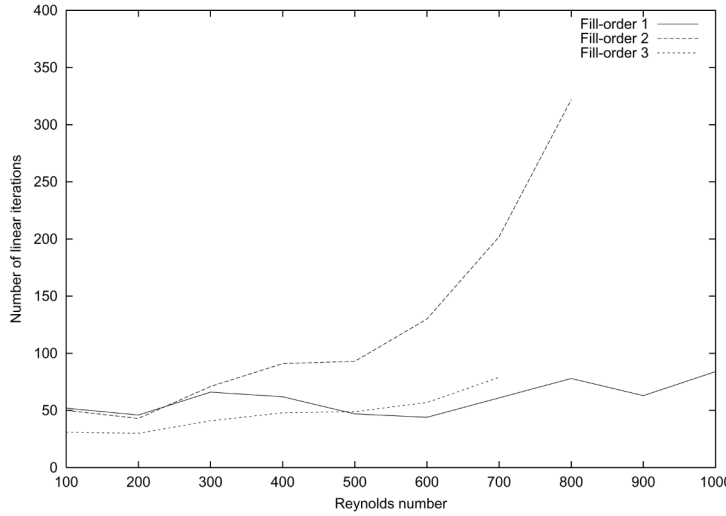


Figure 15.
The number of linear iterations for the iterative solver as a function of the Reynolds number for the Navier-Stokes equations

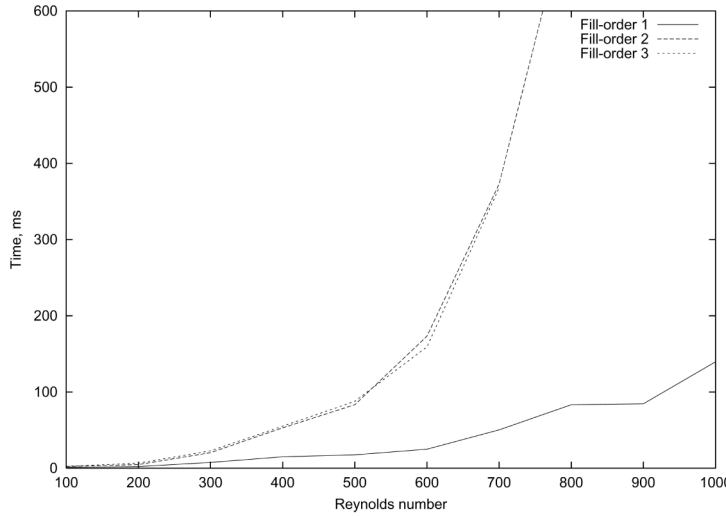


Figure 16.
Time for the solution as a function of the number of degrees of freedom for the Navier-Stokes equations

8. Concluding remarks

A new fill-in rule for ILU preconditioning has been suggested for systems such as those arising in primitive variable Navier-Stokes problems. This fill-in rule is believed to be much more relevant to these equation systems than those based on thresholding.

Different orders of fill-in have been compared for Stokes and Navier-Stokes equations. The conclusion of this work is quite clear. The first-order fill-in

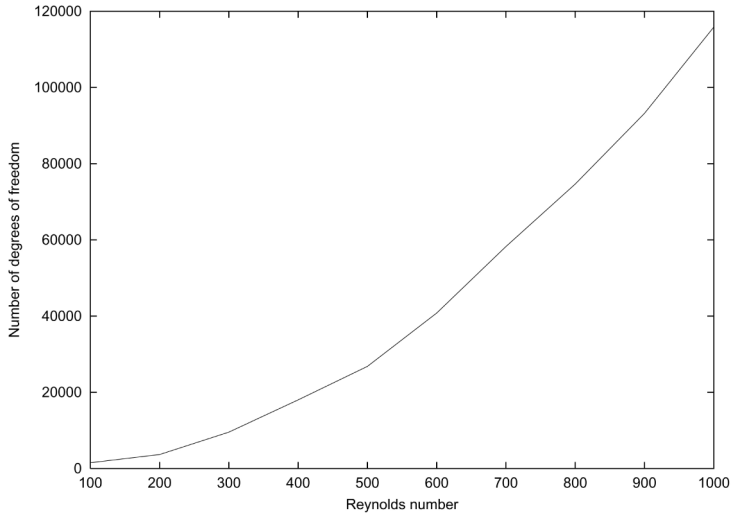


Figure 17.
The number of degrees of freedom as a function of Reynolds number for the meshes shown in Figure 7 for solving the Navier-Stokes equations

algorithm requires less storage and is faster and more robust than higher order fill-in algorithms.

References

Carey, G.F., Wang, K.C. and Joubert, W.D. (1989), "Performance of iterative methods for Newtonian and generalized Newtonian flows", *Int. J. Numer. Methods Fluids*, Vol. 9, pp. 127-50.

Dahl, O. and Wille, S.Ø. (1992), "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*, Vol. 15, pp. 525-44.

Greaves, D.M. and Borthwick, A.G.L. (1998), "On the use of adaptive hierarchical meshes for numerical simulation of separated flows", *Int. J. Numer. Methods Fluids*, Vol. 26, pp. 303-22.

Greaves, D.M. and Borthwick, A.G.L. (1999), "Hierarchical tree-based finite element mesh generation", *Int. J. Numer. Methods Engn*, Vol. 45, pp. 447-71.

Kallinderis, Y. (1992), "Adaptive hybrid prismatic/tetrahedral grids", *Int. J. Numer. Methods Fluids*, Vol. 20, pp. 1023-37.

Meijerink, J.A. and van der Vorst, H.A. (1977), "An Iterative solution method for linear systems of which the coefficient matrix is a symmetric *M*-matrix", *Math. Comp.*, Vol. 31, pp. 148-62.

Sonneveld, P. (1987), "CGS, a fast Lanczos-type solver for nonsymmetric linear systems", *SIAM J. Stat. Comput.*, Vol. 10, pp. 36-52.

Taylor, C. and Hood, P. (1973), "A numerical solution of the Navier-Stokes equations using the finite element technique", *Computers and Fluids*, Vol. 1, Pergamon Press, New York, NY, pp. 73-100.

van der Vorst, H. (1992), "Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear sytems", *SIAM J. Sci. Statist. Comput.*, Vol. 13, pp. 631-44.

Wille, S.Ø. (1986), "An element by element preconditioner for refined finite element grids", *Proceedings of the First International Conference on Numerical Grid Generation in Computational Fluid Dynamics*, Landshut, Germany.

Wille, S.Ø. (1987), "Local gaussian preconditioning of symmetric and nonsymmetric finite element equation solvers", *Proceedings of the First International Conference on Industrial and Applied Mathematics, ICIAM 87*, Paris, France.

Wille, S.Ø. (1992), "A structured tri-tree search method for generation of optimal unstructured finite element grids in two and three dimensions", *Int. J. Numer. Methods Fluids.*, Vol. 14, pp. 861-81.

Wille, S.Ø. (1996), "An adaptive unstructured tri-tree iterative solver for mixed finite element formulation of the stokes equations", *Int. J. Numer. Methods Fluids.*, Vol. 22, pp. 899-913.

Wille, S.Ø. (1998), "Adaptive finite element simulations of the surface currents in the North Sea", *Computer Methods in Applied Mechanics and Engineering*, Vol. 166, pp. 379-90.

Wille, S.Ø. and Loula, A.F.D. (2002), "A priori pivoting in solving the Navier-Stokes equations", *Communications in Numererical Methods in Engineering*, Vol. 18, pp. 691-8.

Young, D.M. and Yea, K.C. (1980), "Generalized conjugate-gradient acceleration of non-symmetrizable iterative methods", *Lin. Alg. and its Applications*, Vol. 34, pp. 159-94.