Linear and Non-linear Iterative Methods for the Incompressible Navier-Stokes Equations*

Simon S. Clift and Peter A. Forsyth
Department of Computer Science
University of Waterloo, Waterloo, Ontario, N2L 3G1

Technical Report CS-93-02

August 5, 1993

Abstract

In this study, the discretized finite volume form of the two dimensional, incompressible Navier Stokes equations is solved using both a frozen coefficient and a full Newton nonlinear iteration. The optimal method is a combination of these two techniques. The linearized equations are solved using a conjugate-gradient-like method (CGSTAB). Various different types of preconditioning are developed. Completely general sparse matrix methods are used. Investigations are carried out to determine the effect of finite volume cell anisotropy on the preconditioner. Numerical results are given for several test problems.

Keywords: Navier-Stokes, nonlinear iteration, preconditioned conjugate gradient

Running Title: Iterative methods for N-S equations

AMS Subject Classification 65F10, 76D05

*This work was supported by the Natural Sciences and Engineering Research Council of Canada, and by the Information Technology Research Center, which is funded by the Province of Ontario.

1 Introduction

Finite volume or finite element discretizations of primitive variable formulations of the incompressible Navier-Stokes equations result in a large system of non-linear algebraic equations. These algebraic equations can be solved in a sequential, decoupled manner (as for example in the SIMPLE algorithm [1]), or more fully coupled methods may be used [2, 3, 4, 5].

There are two main approaches for fully coupled solution methods. One popular technique is to simply use full Newton iteration [3, 4, 6, 7, 8, 9]. Newton iteration has the advantage that convergence is quadratic provided an initial guess is close enough to the solution. Consequently, it is usually possible to obtain solutions of the discrete equations which have a very small non-linear residual, at the expense of a relatively small number of non-linear iterations [10, 3]. On the other hand, it is often the case that arbitrary initial solution estimates may cause the Newton iteration to diverge. In practice, this problem is avoided by using pseudo-timestepping [3, 6], or continuation in the Reynolds number [4]. Frequently, direct methods are used to solve full Newton Jacobians [11], which are very expensive for three dimensional problems. Iterative methods have recently been used for solution of full Newton Jacobians [3, 7], but care must be taken with the ordering of the unknowns [3, 12], and the type of preconditioning used [13].

Another fully coupled solution method is based on "frozen coefficient" iteration. In this approach, non-linear terms are linearized by "freezing" some the unknowns at old iteration values. For example, if v_i^k is the value of the discrete velocity at node i, nonlinear iteration k, then a term in the discrete equations such as

$$v_{i+1/2} v_i$$

would be linearized as

$$v_{i+1/2}^k v_i^{k+1}$$

This frozen coefficient matrix is usually more diagonally dominant than the Jacobian matrix, and hence easier to solve with an iterative method. Direct methods [14, 15], have been used to solve the frozen coefficient matrix. Multigrid methods typically iterate on a variation of the frozen coefficient matrix [16, 17, 18, 19]. Frozen coefficient nonlinear iteration also appears to be a very stable method [14] and convergence can often be obtained with initial solution estimates that would cause Newton iteration to diverge [14]. A disadvantage of frozen coefficient iteration is that convergence of the nonlinear iteration can be very slow, if a small nonlinear residual is required.

Note that both of these nonlinear methods (frozen coefficient and full Newton) require few, if any, iteration parameters. This is a distinct advantage over the more decoupled methods.

The objective of this paper is to compare both of these fully coupled nonlinear iteration methods, while using an iterative method to solve the resulting large sparse matrices. In fact, it will be demonstrated that the best method uses a combination of frozen coefficient and full Newton iteration, in order to utilize the best features of both techniques. Note that some comparisons of full Newton and frozen coefficient nonlinear iteration were carried out in [14], however, a direct method was used for the full Newton iteration.

In this work, the matrices are solved using a preconditioned conjugate gradient method (PCG) with CGSTAB [20, 21] acceleration. An incomplete LU (ILU) type preconditioning is used [22].

Poor results can be obtained with ILU preconditioning unless careful attention is paid to the ordering of the unknowns in the matrix [12, 3, 23, 24, 25], and even to the discretization used in the preconditioning matrix [3], which may be different, in general, from the discretization used in the actual Jacobian. Another level of sophistication is introduced in this article, by noting that an ILU factorization of the frozen coefficient matrix may be used to precondition the Jacobian. Completely general sparse matrix methods are used, and no special properties of the discretization are required. Consequently, we believe that these same methods can be used with little or no modification for finite element or finite volume discretizations on unstructured meshes.

The most efficient methods developed in this work have very few parameters (i.e. no underrelaxation is used), which is convenient for non-expert users of software.

As model problems, we consider the primitive variable formulation of the incompressible Navier-Stokes equations on a variety of two dimensional regions. A standard finite volume discretization on a staggered grid is used. Results will be reported in terms of total CPU time for solution of the nonlinear algebraic equations for a specified convergence tolerance. Total times will include matrix construction, (incomplete) factor and solve.

Comparisons will be made using full Newton iteration with pseudo-timestepping, frozen coefficient iteration, and a combination of frozen coefficient iteration and full Newton iteration. Various preconditioning techniques and ordering methods for solution of the linear equations will also be tested, and compared to direct solution methods. The effect of cell aspect ratio on the performance of the iterative methods will also be demonstrated.

For the convenience of the reader, a nomenclature is provided in Appendix A.

2 The Governing Equations and Their Discretization

The equations governing two-dimensional incompressible fluid flow are those for the conservation of momentum (the Navier-Stokes equations)

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(uu) + \frac{\partial}{\partial y}(vu) + \frac{\partial p}{\partial x} - \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0, \tag{1}$$

$$\frac{\partial v}{\partial t} + \frac{\partial}{\partial x}(uv) + \frac{\partial}{\partial y}(vv) + \frac{\partial p}{\partial y} - \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) = 0, \tag{2}$$

and the conservation of mass equation

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \tag{3}$$

Here u and v are the velocities in the x- and y-directions respectively, and p is the pressure. Equations (1–3) are in dimensionless form with a single parameter, the Reynolds number Re. If the terms $\partial u/\partial t$ and $\partial v/\partial t$ are dropped from equations (1) and (2) respectively, we are left with the elliptic (steady state) flow equations.

2.1 Discretization and Weighting Techniques

The equations (1–3) are discretized using a finite volume approach over a staggered grid as described fully in [1]. The region is divided into rectangular cells, with the pressure unknowns placed in the

centres of the cells, and the velocity unknowns at the faces. The mass conservation equation (the M equation) is integrated over each cell (Figure 1a) with dimensions $\Delta x \times \Delta y$ to give

$$(u_{i+1,j} - u_{i,j})\Delta y + (v_{i,j+1} - v_{i,j})\Delta x = 0. \tag{4}$$

The equations (1) and (2) (the U and V equation respectively) are integrated over 'staggered' cells which have u and v at their centres (Figures 1b and 1c). Using the notation of Reference [1], the two equations can be written more generally as

$$\frac{\partial \phi}{\partial t} + \frac{\partial \mathcal{F}_x}{\partial x} + \frac{\partial \mathcal{F}_y}{\partial y} = S, \tag{5}$$

where

$$\mathcal{F}_x = u\phi - \frac{1}{\mathrm{Re}}\frac{\partial \phi}{\partial x}, \tag{6}$$

$$\mathcal{F}_y = v\phi - \frac{1}{\mathrm{Re}}\frac{\partial \phi}{\partial y}. \tag{7}$$

The terms $\mathcal{F}_x$ and $\mathcal{F}_y$ represent flux per unit volume in the x and y directions respectively. The variable ϕ represents u or v, and S represents the source term (in this case, the pressure differential). Integrating (5) over a cell of dimensions $\Delta x \times \Delta y$ with ϕ at the centre gives

$$\frac{\phi_{i,j}^{n+1} - \phi_{i,j}^{n}}{\Delta t}\Delta x \Delta y + \left(\mathcal{F}_{i+1/2,j} - \mathcal{F}_{i-1/2,j}\right)\Delta y + \left(\mathcal{F}_{i,j+1/2} - \mathcal{F}_{i,j-1/2}\right)\Delta x = S_{i,j} \tag{8}$$

$$S_{i,j} = \begin{cases} (p_{i,j} - p_{i-1,j})\,\Delta y & \text{where } \phi = u, \\ (p_{i,j} - p_{i,j-1})\,\Delta x & \text{where } \phi = v. \end{cases}$$

The terms $\mathcal{F}_{i+1/2,j}$ and $\mathcal{F}_{i-1/2,j}$ represent the values of $\mathcal{F}_x$ at the left and right cell interfaces, and $\mathcal{F}_{i,j+1/2}$ and $\mathcal{F}_{i,j-1/2}$ represent the values of $\mathcal{F}_y$ at the top and bottom cell interfaces respectively. Note that the equations are fully implicit; all variables except the $\phi_{i,j}^{n}$ of the time derivative term are solved at the new time (thus $\phi_{i,j} \equiv \phi_{i,j}^{n+1}$).

Finally, we must discretize the flux terms at the cell faces. Taking, for example, the $\mathcal{F}_x$ term at the interface between cells centred at $\phi_{i,j}$ and $\phi_{i+1,j}$, we may write $\mathcal{F}_{i+1/2,j}$ as

$$\mathcal{F}_{i+1/2,j} = u_{\mathrm{avg}}\phi_{i,j} + (\mathcal{A} + \mathcal{B})\left(\frac{\phi_{i,j} - \phi_{i+1,j}}{\mathrm{Re}\; h}\right) \tag{9}$$

$$\mathcal{B} = \begin{cases} 0 & \text{if } u_{\mathrm{avg}} \geq 0 \\ -\mathrm{Re}_c & \text{if } u_{\mathrm{avg}} < 0 \end{cases}$$

$$u_{\mathrm{avg}} = \frac{(u_{i,j} + u_{i+1,j})}{2}$$

where h is the distance between the two grid points, u_{avg} is the average x direction velocity through the interface between the cells, and $\mathrm{Re}_c = \mathrm{Re}\; u_{\mathrm{avg}}\; h$ is the cell Reynolds number. In concert with $\mathcal{B}$, $\mathcal{A}$ can be chosen to implement a number of weighting strategies. It is not the objective of this

paper is to examine the various possibilities, nor to consider the effects of weighting methods on the final solution. Most of the tests reported in this paper will be carried out using the *power-law* weighting method described in [1]. This is a popular method, and is easily implemented by setting

$$\mathcal{A} = \max\left[0, (1 - 0.1 \mid \mathrm{Re}_c \mid)^5\right]. \tag{10}$$

Central weighting will be used for one test case.

3 Solution Strategy Components

With the equations discretized, we are left with a large non-linear system that must be solved. We have chosen to solve the system in its fully coupled form. Decoupling one set of equations, such as is done with the conservation of mass equations in the SIMPLE family of algorithms, may require more non-linear iterations as compared to the simultaneously solved set of equations [5]. Although the work per non-linear iteration is less for SIMPLE type methods compared to fully coupled approaches, the experimentally determined computational complexity of SIMPLE appears to be $\mathcal{O}(N^2)$ [5], which compares to $\mathcal{O}(N^{3/2})$ of the fully coupled methods used in this work (where N is the number of cells in the discretization).

Both the full Newton (FN) and frozen coefficient (FC) non-linear methods iterate according to the equation

$$\mathbf{A}^k(x^{k+1} - x^k) = -r^k. \tag{11}$$

where $\mathbf{A}^k$ is the linearized equation matrix (LEM) formed from values determined in the k^{th} non-linear iteration, x^{k+1} represents a vector of u, v, and p variables from iterations k and $k+1$, and r^k represents the non-linear residual vector formed by evaluating the u and v momentum equations, and the conservation of mass equation at the k^{th} non-linear iteration. For FN iteration, the matrix $\mathbf{A}^k$ is the full Jacobian. In the case of FC iteration, some derivatives in $\mathbf{A}^k$ are ignored.

An important point to note is that the FN and FC methods differ only in the construction of the LEM. The evaluation of the residual is the same for both methods, and hence, both methods can be used to solve the same problem. Both methods, when they converge (i.e. when the residual approaches zero), arrive at the same answer to a particular problem, although they may approach that answer differently.

3.1 Full Newton Iteration

Full Newton iteration is performed by constructing the LEM with all the u, v and p unknowns of the discretized equations evaluated at the current iteration point. To illustrate, consider equation (9) for the U equation, and its partial derivative with respect to u^k. The Re_c term is fully expanded, and when $0 < \mathrm{Re}_c < 10$

$$\mathcal{F}^{FN}_{i+1/2,j} = \left(\frac{u^k_{i,j} + u^k_{i+1,j}}{2}\right) u^k_{i,j} + \left(1 - \frac{\mathrm{Re}\, h}{10}\frac{\left(u^k_{i,j} + u^k_{i+1,j}\right)}{2}\right)^5 \frac{u^k_{i,j} - u^k_{i+1,j}}{\mathrm{Re}\, h}. \tag{12}$$

Thus

$$\frac{\partial \mathcal{F}^{FN}_{i+1/2,j}}{\partial u^k_{i,j}} = u^k_{i,j} + \frac{u^k_{i+1,j}}{2} + \frac{1}{\mathrm{Re}\, h}\left[1 - \frac{\mathrm{Re}\, h\left(u^k_{i,j} + u^k_{i+1,j}\right)}{20}\right]^5 - \frac{u^k_{i,j} - u^k_{i+1,j}}{4}\left[1 - \frac{\mathrm{Re}\, h\left(u^k_{i,j} + u^k_{i+1,j}\right)}{20}\right]^4 . \tag{13}$$

The full expansion of the U, and V equations is similar, and hence will not be given here. The M equation is, of course, linear in the momenta.

3.2 Frozen Coefficient Iteration

Frozen coefficient iteration uses a simplified form of the LEM. The M equation is expanded as in the full Newton matrix. In the U and V equations, we replace u^k_{avg} with $u^0_{\mathrm{avg}} = u^k_{\mathrm{avg}}$. This term is not expanded when constructing the partial derivatives, but is "frozen" as a value for the coefficients and nothing more. Consider equation (9) from the U equation again, and its partial derivative with respect to u^k, with the same conditions as specified for equation (12).

$$\mathcal{F}^{FC}_{i+1/2,j} = u^0_{\mathrm{avg}} u^k_{i,j} + \left(1 - \frac{\mathrm{Re}\, h\, u^0_{\mathrm{avg}}}{10}\right)^5 \frac{u^k_{i,j} - u^k_{i+1,j}}{\mathrm{Re}\, h} . \tag{14}$$

Thus we obtain

$$\frac{\partial \mathcal{F}^{FC}_{i+1/2,j}}{\partial u^k_{i,j}} = u^0_{\mathrm{avg}} + \frac{1}{\mathrm{Re}\, h}\left[1 - \frac{\mathrm{Re}\, h\, u^0_{\mathrm{avg}}}{10}\right]^5 . \tag{15}$$

Not only are the partial derivatives less complicated, but certain other terms which appear in the full Newton matrix disappear entirely under this scheme. The FC matrix, therefore, has fewer nonzeros than the FN matrix to store. To recapitulate, the FC matrix is can be regarded as a FN matrix with some of the derivative terms set to zero, and other terms slightly modified.

3.3 Notes on the Resulting Matrices

Several matrix characteristics, important to PCG iterative matrix solvers, are evident from the given expanded sections of the two types of LEM's. For example, the rows of the FC matrix corresponding to the momentum equations have the property that the diagonal is positive, and the off-diagonal terms corresponding to neighbouring momenta are negative.

However, for full Newton Jacobian (FN) linear equations, for cells where $\mid \mathrm{Re}_c \mid < 10$, derivatives of $\mathcal{A}$ (Equation 10) appear. These may cause off-diagonal momentum derivatives to appear which have the opposite sign to the FC momentum terms. This tends to decrease the diagonal dominance of the matrix and hence may cause problems for an iterative solver.

The M equation contains no pressure terms. Thus $\partial M_{ij}/\partial p_{ij} = 0$ for both FC and FN LEM's, and hence the diagonal entries for the M equations are zero. This can be the cause of problems when the matrix is factored directly, as well as when it is partially factored to produce a preconditioner

for a PCG type matrix solver. Zeros on the diagonal will cause a non-pivoting matrix factorization to fail, so precautions to prevent this in the partial factorization must be taken. Since pivoting during factorization would require a more complicated data structure, and greatly slow the process, it is not considered.

For direct methods, this zero pivot problem can be avoided by realigning the equations and unknowns [11, 26] or preprocessing the matrix [27]. In the case of iterative methods, either of the previous two approaches may be used, or care must be taken with the ordering of the unknowns [24, 3].

3.4 Matrix Solution Methods

Iterative, PCG-type matrix solvers have been found to be effective in solving the matrices arising from fluid flow problems. We use CGSTAB acceleration [21], with right preconditioning, which was chosen over a number of other available methods based on previous experiments (see [28]). As a preconditioner, we use an incomplete LU factorization, keeping the first few levels of fill-in (referred to ILU(n) where n is the highest level of fill-in kept) [29, 3]. It is possible to use a drop tolerance preconditioning, but tests have shown that this method is sometimes unreliable for high Reynolds number problems [30], and hence will not be considered here.

3.5 Pre-elimination

Although special ordering techniques can be used to ensure that an incomplete factorization does not produce a zero pivot [3], this method does not necessarily produce a small amount of fill in the incomplete (ILU) factorization. For direct methods, realignment of equations and unknowns has been used successfully [11]. For example, $\partial U_{ij}/\partial p_{ij} \neq 0$, and $\partial M_{ij}/\partial u_{ij} \neq 0$. Consequently, nonzero diagonals can be obtained by interchanging the rows of the matrix corresponding to the U and M equations, as described in [11]. Although this method is successful if a direct method is used for the matrix solve, our tests of this realignment (or row interchange) procedure produced poor results for iterative methods.

Alternatively, *pre-elimination* can be carried out on the rows of the LEM matrix corresponding to the mass conservation M equation (4). Each pressure term in the staggered grid has from two to four adjacent velocity terms, all of which appear in the discretized M equation. Gaussian elimination is performed using the U and V equations corresponding to those adjacent velocity terms, which eliminates the neighbouring velocity terms from the M equation. This also introduces non-zero terms in the diagonal entry of the M equation. We perform no non-symmetric row or column reorderings such as was done in [11]. Experiments showed that selecting only one U or V equation to pre-eliminate against the M equation caused poor convergence of the PCG methods. The best results were obtained when all adjacent equations were used. Note that the resulting pressure terms in the pre-eliminated M equation are similar to the SIMPLE pressure equation. Of course, this pre-eliminated equation has additional momentum terms as well.

More formally, let $\{M\}$ be the set of all rows of the LEM matrix corresponding to mass conservation equations. Let k be the row of the LEM matrix which corresponds to the mass conservation equation at cell (i, j), $M_{i,j}$, and let $\{A\}_{i,j} = A_{i,j}$ be the elements of the LEM equation (11). Then the pre-elimination algorithm is shown in Figure 2. The operation of pre-elimination is $\mathcal{O}(N)$, where

N is the number of pressure-centred cells in the grid. It is quick to perform, but does somewhat increase the matrix storage requirements.

This pre-elimination step results in a preprocessed matrix $(A)^p$ and right hand side vector $(-r)^p$ which are row equivalent to the original system. No approximations are made in this pre-elimination step. Note that in [27], some of the terms in the pre-elimination are lagged an iteration (placed in the right hand side vector), and hence the matrix in [27] is not row equivalent to the original FC matrix.

To avoid a profusion of superscripts, the superscript p, indicating pre-eliminated matrix A and right hand side $-r$ will be dropped in the following. It will be clear from the context that a pre-eliminated or non-pre-eliminated matrix is being used.

This pre-elimination method can be used for both complete or incomplete factorization. Note that the realignment procedure of [11] will require modification in the presence of internal boundaries, while the pre-elimination method will always produce non-zeros on all diagonals, regardless of the solution domain. It should also be noted that the pre-eliminated matrix does not, in general, have a symmetric structure. However, this does not pose any particular difficulties for our matrix solution methods.

3.6 Ordering Methods

The ordering of the unknowns can have a large effect on the convergence rate of PCG type iterative methods [30, 31]. The Minimum Discarded Fill (MDF) ordering method attempts to determine a good ordering by minimizing the size of the discarded fill terms which are ignored in the incomplete factorization. More specifically, at each stage of the incomplete elimination, the next pivot element is selected (amongst the remaining uneliminated elements) which minimizes the discarded fill. For matrices having a large number (on average) of nonzeros per row, MDF can be costly to compute. The Minimum Updating Matrix (MUM) ordering method is an approximation to MDF ordering which is less costly to compute. For more details concerning these ordering methods, the reader is referred to [3, 30, 31]. Since at each stage, MUM ordering attempts to minimize the size of discarded fill terms, MUM ordering will attempt to determine a pivot sequence which tends to avoid small elements on the diagonal. This is because a pivot row with a zero on the diagonal would have an infinite discarded fill.

In the case of a PCG type iterative method, MUM [3] was shown to be effective without pre-elimination. Since it uses information from the numerical entries of the matrix, and not just the graph, MUM ordering usually produces an effective ordering. Because of this numerical entry sensitivity, re-ordering is occasionally required during the solution of the problem if the ordering is to remain optimal. MUM is fairly robust, and in all our experiments has rarely produced an ordering that caused the iterative matrix solver to fail. However, since MUM ordering is actually an approximation to the MDF ordering described in [25], some information is lost using MUM ordering. While MDF ordering can determine a pivot sequence which produces rapid convergence for anisotropic problems [30, 31], MUM ordering can sometimes produce poor orderings for anisotropic problems. In the context of Navier-Stokes problems, anisotropies are generated by discretizations having large cell aspect ratios. However, in practice, MDF ordering is too time consuming for Navier-Stokes type matrices.

It is also more costly to perform MUM ordering, both in terms of storage space for its data struc-

tures, and the time it consumes, than the purely graph-based alternatives tested in the following experiments.

Minimizing matrix bandwidth also tends to improve the quality of ILU factorizations. Briefly, this is because for a given number of nonzeros in the ILU factorization, bandwidth minimizing orderings tend to retain higher level fill terms compared with other orderings [12]. To this end, a Reverse Cuthill McKee [32] (RCM) ordering was also used in conjunction with pre-elimination. First the matrix was pre-eliminated. For the purposes of generating an RCM ordering, the data structure of the pre-eliminated matrix was symmetrized, adding non-zero storage to the data structure as required. (Of course, these non-zeros were removed in the actual symbolic incomplete factorization.) RCM ordering was then performed on this new matrix graph. This heuristic proved effective, when combined with pre-elimination, for FC LEM's. It has the advantage of being quick to perform, and requires very little storage for intermediate work space. It was expected that with pre-elimination, no special treatment of the ordering, (one based on matrix values,) would be required. RCM with pre-elimination will be referred to as Pre+RCM.

Note that in [13] the zero pivot problem was avoided by ordering the pressure unknowns last. While this is a robust method, tests in [3] indicated that pressure last orderings were poor in terms of convergence of the iterative solver.

3.7 Preconditioning the Full Newton Jacobian

Computational experiments indicated that the frozen coefficient matrix (LEM) was relatively easy to solve compared to the full Newton Jacobian (FN) matrix. Based on the observation in [3] that the performance of iterative methods is sometimes improved when preconditioning with an upstream weighted matrix, or equivalently, a preconditioning matrix with additional artificial viscosity [33, 30, 34], we have also tested the use of a frozen coefficient matrix (FC) as a preconditioner for a pre-eliminated Full Newton (FN) matrix. With the M equations pre-eliminated, the full Newton LEM's still produced unsatisfactory ILU preconditioners when ordered with pure matrix graph methods. They were prone to very small diagonal entries after incomplete factorization, which in turn led to numerical instability in the CGSTAB acceleration. The FC LEM proved to be an effective FN LEM preconditioner.

It is important to note that the solution to the LEM remains the same, regardless of the preconditioner chosen. More precisely, if right preconditioning is used, then the CGSTAB algorithm is applied to the equivalent system

$$\begin{aligned} \left(\mathbf{A}^k\right)'\left(x^{k+1}-x^k\right)' &= -r^k \\ \left(\mathbf{A}^k\right)' &= \mathbf{A}^k\mathbf{P}^{-1} \\ \left(x^{k+1}-x^k\right)' &= \mathbf{P}\left(x^{k+1}-x^k\right) \end{aligned}$$

where $\mathbf{P}$ is the incompletely factored FC matrix.

In the following, when pre-elimination is used with the full Newton (FN) matrix, frozen coefficient preconditioning will be used. To be more precise, the FC matrix is constructed, and then pre-eliminated. This pre-eliminated FC matrix is then incompletely factored and used as a preconditioner. The FN matrix and right hand side are pre-eliminated as usual, and are used in the CGSTAB algorithm.

4 Test Cases

The solution techniques were tested on over 30 two-dimensional geometries. In the interests of brevity, five representative problems are presented in this paper, ranging from the standard Driven Cavity problem, to the more difficult Backward Step [4]. We have found that the standard test problems appear to belong to two categories, those with roughly square physical dimensions (e.g. the driven cavity) and those having anisotropic physical dimensions (the Backward Step). All walls in these problems are set to no-slip boundary conditions ($u = 0, v = 0$ at the boundary).

4.1 Driven Cavity (DC)

This test is over a square region, with a non-dimensional width of 1.0, with a lid-driven flow. See [35] for the details of this common test.

4.2 Two In, One Out, Symmetric Flow Chamber (Symm)

This test involves a more complicated geometry, and accelerating flow, and is specified in Figure 3 (without the interior blocks A and B in the middle of the chamber). The flow speed reaches the maximum of 1.0 at the upper outlet. The inlets and outlet have parabolic inflow and outflow conditions. For reasonable Reynolds numbers it was expected that we would observe symmetric flow patterns.

4.3 Two In, One Out, Asymmetrically Blocked Chamber (Asym)

This geometry is fully specified in Figure 3. As with the Symm problem, flow speed reaches the maximum of 1.0 at the upper outlet, and inlets and outlet have parabolic inflow and outflow conditions. This problem was created to demonstrate that flow symmetry, and interior boundaries do not affect our method. As we shall see later, patterns of flow in the chamber are fairly complex.

4.4 Three Chamber Problem (3Cham)

The dimensions of this problem are given in Figure 4. The walls of the chamber are specified to be 1 grid cell thick. The maximum fluid speed is attained in the gaps between the chambers. Parabolic inflow and outflow conditions have been normalized so that the maximum speed in the chamber is 1.0. This problem involves a larger pressure differential than the others, and also experiences accelerating flow due to the gaps in the wall being smaller than the inlet and outlet.

4.5 Backwards Facing Step (BFS)

The dimensions of this problem are given in [4]. This problem, proposed as a standard test case, has parabolic inflow conditions. We imposed parabolic outflow conditions since in [4] and [36] this was given as within 1% of being correct. This problem is distinguished by physical dimensions that greatly exceed the characteristic length used to set the Reynolds number. Maximum flow speed was set at 1.5 at the inlet, to adhere to the published setup.

5 Comparing Non-Linear Methods

Given the non-linear methods outlined above, three approaches were studied. All three solved the steady state U, V and M equations beginning from the initial guess of a zero flow field ($u = 0, v = 0$ over the entire region). To arrive at the steady state, all these methods solve the time-dependent equations at $t = 10^6$, which in effect causes the time-dependent term to disappear. Previous tests [3] have shown that for flows at Re$\simeq$ 1000 this time condition produces the steady-state flow to four digits accuracy. The approaches used were:

1. Frozen coefficient iteration from the beginning until convergence (AllFC).
2. Full Newton iteration from the beginning until convergence (AllFN).
3. Frozen coefficient iteration until a certain non-linear residual reduction is observed, then full Newton iteration (FC+FN).

The AllFC method is robust, allowing the use of a single, very large pseudo-timestep of 10^6. No underrelaxation is required for convergence, even starting from a zero flow initial guess, but convergence tends to be slow.

Because a zero flow field generally does not appear to lie within the radius of convergence of Newton's method when it is applied to the elliptic form of the Navier-Stokes equations, pseudo-timestepping is required, and under-relaxation used to improve the efficiency of the solve. The method we used is fully described in [3]. Note that a very aggressive timestepping strategy is used. Typically, $10 - 15$ pseudo timesteps are required to reach the steady state, from an initial state of zero velocity. The final time step is typically of the order of 10^4, and results in a rapid, large reduction of the residual of the elliptic equations. Also of note is that this method uses an incompletely factored FN LEM as a preconditioner.

As with the AllFC method, the FC+FN method uses a single time step of 10^6. Early experiments showed that for small problems (where all problem dimensions, in dimensionless units, were $O(1)$, e.g. DC, Symm, Asym, 3Cham) the switch from FC to FN could occur at a 10^{-2} non-linear residual reduction. Problems with a larger dimension (i.e., in dimensionless units, one of the problem dimensions was $\gg 1$), such as BFS, required continuing with FC iteration until the non-linear residual was reduced to between 10^{-3} and 0.7×10^{-3}. Note that this method uses the FC LEM as a preconditioner at all stages, for reasons that will be explained in section 6.3.

A minor point to note is that the internal data structure for all methods was constructed for the FN LEM. Thus, during any FC iterations, some zero entry overhead was introduced. However, this also meant that the ILU factorization of the FC LEM contained more entries, and was therefore more complete. The reason for this in the FC+FN method was to avoid reworking the data structure after the change to the FN method. Although these extra non-zeros could be easily eliminated for the All FC method, they were left in the FC data structure to avoid skewing the tests by changing the number of zeros in the ILU factorization. In fact, some tests showed that an ILU factorization of the FC matrix using a symbolic ILU based on the FN data structure was slightly superior to an ILU based solely on the FC data structure.

It remains unclear how one can determine an appropriate point at which the FC+FN method should switch from the first to the second phases. This would require determining whether the

intermediate solution is within the radius of convergence of Newton's method, which is not an easy task. We suggest that at the point where the method attempts to switch to FN, the intermediate solution be saved. Our experience suggests that if Newton's method is going to diverge, it will tend to diverge on both the first and second FN iterations. If this is the case, then the saved intermediate solution can be restored and more FC iterations performed. Presumably, a point will eventually be reached where the Newton iteration will converge.

5.1 Convergence Criteria

The solution for a particular time step was considered converged (and thus the solution for the All FC and FC+FN methods) when a non-linear iteration made no change to the solution greater than 10^{-4} in any of the u, v, or p variables. The final time step of the All FN method was considered converged when an entire time step made a such a small change. (Recall that a very aggressive timestepping method was used, so that typically this last timestep was of the size $O(10^5)$ in dimensionless time). As pointed out below, this convergence criteria is not necessarily the best, but it is commonly used.

This convergence criterion had some interesting side-effects. The All FC strategy finished after having reduced the non-linear residual by a much smaller factor than the strategies that finished with Newton iteration. FN iteration reduced the non-linear residual, in general, by a factor of 10^{-6} or more. The FC methods tended to terminate with less than a 10^{-5} residual reduction. Convergence criteria based on an absolute reduction in non-linear residual would guarantee that the solution is accurate to a given degree. However, the FC method converges so slowly that for a number of problems (BFS in particular) the cost of a large residual reduction would be prohibitive. Other experiments with tighter tolerances demonstrated that more extreme residual reductions were easily obtained, but we decided, given the non-linear residual reduction typical in most publications, that the 10^{-9} reduction typical of our FC+FN method was ample.

Mixed tolerance convergence criteria was used for the iterative linear solver. We define $\|r_0^L\|$ as the l_2 linear residual at the start of an iterative matrix solve (which the reader will note is equal to the non-linear l_2 residual at that point), and $\|r_m^L\|_2$ as the linear residual after the m^{th} linear solver iteration. The linear residual of each LEM had to be either reduced by a relative precision factor of 10^{-6}, or the change in the updates in the linear iteration for all variables had to be less than 10^{-8}. More precisely

$$\frac{\|r_m^L\|_2}{\|r_0^L\|_2} < 10^{-6} \tag{16}$$

or

$$\max_{i,j} \left[|u_{ij}^{m+1} - u_{ij}^m|, |v_{ij}^{m+1} - v_{ij}^m|, |p_{ij}^{m+1} - p_{ij}^m| \right] < 10^{-8}. \tag{17}$$

The reasoning behind the tight relative residual reduction is based on an estimate of how much the flow will vary from grid cell to grid cell. The large dimension problems have flow speeds and pressures that vary by a factor of up to 30 times less than those of the small problems because of their extreme length. Hence a tighter convergence tolerance was required in the linear solve in the early stages. Early experiments showed that a less accurate a linear solve led to convergence problems in the early steps of the solution.

The absolute tolerance criterion (17) is simply a time saver which rescues the matrix solver from having to reduce the linear residual to tolerances far beyond those required by the non-linear convergence tolerances during the later non-linear iterations.

The CGSTAB acceleration was generally allowed to continue for up to 300 iterations. We encountered a number of cases (typically when the switchover from FC to FN iteration in the FC+FN method was done too early), when CGSTAB would "stall" (i.e. many iterations with little or no residual reduction). If the linear residual remained in a small range ($\pm 3\%$) for 30 iterations, we considered CGSTAB to be in this state. Restarting CGSTAB by simply calling the routine again, with the initial guess equal to the one attained in the stalled state, generally caused the acceleration to continue reducing the linear residual. Occasionally the restart had to be performed more than once, but was permitted no more than 4 times. There is still the possibility, however, that the criterion (17) could cause CGSTAB to return earlier with a poor solution if the CGSTAB acceleration is "stalling" [21], and the above restart condition was not triggered. We did not observe this problem in any of our tests.

Few of the runs presented in this paper required this restart. The restart was generally only needed in geometries not presented in this paper where the solver converged to one of two or more possible solutions to the flow (i.e. the flow was bifurcating and not steady state) or when the preconditioning was inadequate due to ordering or aspect ratio problems (see Section 7.4). If the four restarts failed, the entire solution process was stopped, but this was only encountered when the non-linear Newton process was diverging.

In practice, these criteria worked well on virtually all the problems, provided a good preconditioner was selected. The reader will note in the results that follow, that when used with FN iteration, these convergence criteria produce an overall non-linear residual reduction that is indeed quite large.

5.2 Test Results

Table 1 compares the three non-linear methods over five test problems. All CPU times given in this paper are for a Sun 4/670, which is nominally rated at 4 MFLOPS. All arithmetic is Fortran double precision. The five test problems were DC, Asym, and 3Cham on an 80x80 grid at Re=1000, and BFS on a 400x20 grid, with Re=800. The value Re=800 for BFS was chosen to match the tests run in [4]. The FC+FN method switched to FN at a 10^{-2} residual reduction in all but the BFS test, where the switchover point was 0.7×10^{-3}. These switchover points were experimentally determined and all matrix ordering was done with MUM at ILU(4).

From Table 1, we note that the FC+FN method was consistently superior to both the FC and FN method (note that the FC and the FN methods did not converge within the maximum CPU time limit for the BFS problem). The FC+FN also produced a considerably smaller nonlinear l_2 residual reduction than the FC method.

Table 1 does not, however, show the manner in which the convergence occurred. In Figure 5 the non-linear l_2 residual reduction is plotted against CPU time for the DC problem, which shows the rapid convergence of FC+FN. Figure 6 shows similar results for the BFS problem. Note that in both cases, the line for the FC+FN method becomes suddenly steeper. This is a result of switching from FC to the FN method.

The rapid, and large l_2 residual reduction of the FC+FN method, and the general pattern of

residual reduction of all the methods shown in Figure 5 was typical for all problems of roughly square geometry. In particular, note that both FC+FN and All FN show quadratic convergence as the nonlinear residual becomes small. Figure 6 was also typical of the methods' residual reduction patterns for problems with a longer physical domain. In this case, the All FN method did not converge within the CPU time limit, and the region of quadratic convergence was never reached. However, if the time axis of Figure 6 is extended sufficiently far to the right, then the All FN method does eventually show quadratic convergence behaviour, similar to Figure 5.

Note that the full Newton LEMs can take longer to solve than the FC LEMs. The question also arises of what parameters are best to use when performing the linear solve. This leads us to our next section, which covers the linear methods.

Figures 7, 8 and 9 display the streamline plots for the Symm, Asym and 3Cham problems, solved on a 160×160 grid, at Re=1000, with power-law weighting.

6 Comparing Linear Methods

6.1 The Ineffectiveness of Direct Methods

As noted in the introduction, direct methods are often used to solve LEM matrices. Table 2 shows why this, even in two dimensions, is not advised. The direct method shown (MD + Direct) in the table uses minimum-degree ordering [37], a popular and generally accepted method. The iterative method uses CGSTAB, with both pre-elimination with RCM ordering, and MUM ordering. The problem being solved is the Driven Cavity, Re=1000, on a set of grids of increasing size. From these experiments we see that the direct method is slower. In these tests we see the direct method is $\mathcal{O}(N^{1.8})$, while the iterative methods are about $\mathcal{O}(N^{1.3})$, where N is the number of grid cells. (This is the complexity of the solution of the entire non-linear problem). For model second order elliptic problems, the complexity of the linear solve for a PCG method is $\mathcal{O}(N^{3/2})$ [38]. Direct methods also tend to take up more storage space than iterative methods [38, 5]. Thus we dispense with considering direct methods further.

6.2 Level of ILU Factorization

The matrix computations performed in our solver are done using a static data structure. In determining the appropriate ILU to use for the linear solve, we considered overall performance over the entire nonlinear solution. In [3] (see also [39]) it was determined that ILU(2) or ILU(3) was best for the AllFN process.

The current experiments showed that for the Pre+RCM method, ILU(5) was the quickest (balancing the time for incomplete factorization of the preconditioner with the time for the iterative solve). With the MUM ordering method, ILU(3) was best for all problems up to a grid size of 80×80 for square problems and 400×20 for rectangular problems. With larger grid sizes, ILU(4) was required; lower levels of ILU demonstrated erratic convergence and a higher overall time complexity order. Thus for the main body of tests to follow, ILU(4) is used with MUM ordering.

If these experiments were to be extended to three dimensions this issue would have to be re-evaluated. These relatively high levels of incomplete factorization would likely lead to unreasonable

amounts of fill-in because of the larger number of non-zeros per row in the three dimensional case.

6.3 Preconditioning the FN LEM with MUM Ordering

As already noted, attempting pre-elimination and RCM ordering on the FN LEM's produced ILU factorizations with unreasonably small diagonals, which in turn led to the failure of the iterative matrix solver. The FC LEM proved to be a good preconditioner for Pre+RCM.

The question remained as to what the best preconditioner for the FN stage with MUM ordering was (recall that no pre-elimination step is necessary with MUM ordering). Somewhat surprisingly, the FC LEM turned out to be a better preconditioner for the FN stage even with MUM ordering. Table 3 shows tests run on the 80×80 DC and 400×20 BFS problems that illustrate the point. Thus from this point on, all tests with MUM ordering for the FC+FN method use FC preconditioning at all stages.

7 Grid and Reynolds Number Dependence of FC+FN

In order to calculate the dependence of the solution time for the FC+FN method (and its linear solution strategies) on the size of the grid, and the Reynolds number the following series of tests was performed. The Reynolds number was kept within the range 100 to 1000, where one can expect a steady state flow to exist.

7.1 Grid Size Dependence

For the square problems (3Cham, Asym and DC) tests were run at Re=1000 for grid sizes of 40×40, 80×80, and 160×160. For the rectangular problems tests were run at Re=800 for grid sizes of 200×10, 400×20 and 600×32. The three grid sizes for each group will be referred to as coarse, medium and fine respectively. The raw results for FC+FN with Pre+RCM are compiled in Tables 4 and 5. The results for FC+FN with MUM ordering are compiled in Tables 7 and 8. Tables 6 and 9 list the time complexity exponent for the method over the test regions. This complexity exponent was taken from the medium and fine grids, and was measured in terms of the grid size.

Of note is that although MUM ordering required less CPU time over the range of grid sizes tested, it produced, in DC, Symm, Asym, and 3Cham, a higher time complexity exponent. Pre+RCM took between 40% and 60% more time at the medium grid, but only between 2% and 23% more time for the fine grid for these square-domain tests.

A remarkable result is that for square-domain problems, the Pre+RCM time complexity is considerably below the theoretical $\mathcal{O}(N^{3/2})$ that would be expected for a second order linear elliptic problem. This may be due to fact that we are solving a non-linear problem, with linearized equations considerably different from the model problems used for the usual analysis. As well, the convergence criteria (16 - 17) may have an effect. MUM ordering produced no surprising results for square domains, but for the BFS problem, (our long-dimensioned example), it produced a time complexity of $\mathcal{O}(N^{1.34})$, again faster than would be expected for a model problem [38].

The number of non-linear iterations for both the FC and FN phases remained roughly the same for square-domain problems, and only became substantially larger for the BFS test. Other experiments on the BFS region have determined that the boundary conditions are not to blame

for the difficulty in obtaining a solution. When the region was shortened to length 1.0, and the boundary conditions kept the same (although this is highly non-physical) the solution converged in roughly the same time as the DC test. Our tests seemed to indicate that the extreme length relative to the characteristic length used to set the Reynolds number was at the root of the generally slower convergence. In any case, the convergence for the Pre+RCM method on the BFS problem was roughly the expected $\mathcal{O}(N^{3/2})$, which should reduce to $\mathcal{O}(N^{4/3})$ in three-dimensional cases [38]. The reader should note, however, that the matrix-value sensitive MUM ordering did much better on the same test.

7.2 Reynolds Number Dependence

For the square problems, tests were run at Re=100, 500 and 1000. The rectangular problems were run at Re=800 instead of 1000. The timing and iteration results are given in Tables 10 and 11 for Pre+RCM and MUM orderings. We have also listed the time complexity exponent measured in terms of the Reynolds number for the last two tests (i.e. time = $\mathcal{O}(Re^{Ord})$, for a fixed grid size but varying Re).

The MUM ordered tests show a less sharp increase in time with Reynolds number than the Pre+RCM ordered tests. This would seem to indicate that sensitivity to the contents of the matrix, and not just the graph, figure more prominently in the solution time as the Reynolds number increases.

Table 11 shows that the behaviour of the BFS problem is somewhat anomalous compared to the other test problems. As the Reynolds number increases, the other problems show only a small increase in the number of frozen coefficient iterations required to obtain a solution which is within the radius of convergence for Newton's method. However, the BFS problem shows a big jump in the number of frozen coefficient iterations between Re=500 and Re=800. This increase in solution time is similar to that reported in [40]. Examination of the streamline plots of the solution after each nonlinear frozen coefficient iteration showed an interesting behaviour. The primary eddy at the step corner and the secondary eddy at the upper wall form after only a few iterations. Smaller eddies also form and disappear. However, the two major separation zones near the step move very slowly (in terms of iterations) to their final position. It is this very slow movement to the final position which causes the large increase in frozen coefficient iterations between Re=500 and Re=800. The frozen coefficient iteration is thus qualitatively similar to the behaviour of a transient approach to the steady state solution, as described in [41]. Consequently, for problems with long, thin domains, it may be that the frozen coefficient iteration is inefficient for obtaining a solution that is within the radius of convergence of Newton's method. Nevertheless, this approach is still extremely robust.

7.3 Solution Accuracy

As already noted, with the FC+FN method, the discrete equations were solved to a small non-linear residual. Although we are mainly concerned in this paper with efficient techniques for solution of the discretized equations, and not the accuracy of any particular type of discretization, it is worthwhile to compare our solutions for the DC and BFS problems with previously published computations.

The center vortex of the driven cavity at Re=1000 on a 200×200 grid had a maximum negative

streamfunction value of -0.1154 for power law weighting. For reference, computations were also made for the same problem with central and hybrid weighting (using the FC+FN method), where the values were -0.1183 and -0.1182. Table 12 lists the maximum streamfunction value, and the maximum negative x-direction velocity on the vertical centerline of the cavity, for all our tests, and those found in [16, 35, 42, 43, 40, 44]. Our measurements fell within the ranges given, and closely matched when the upwinding techniques were the same. Taking into account the variation that typically arises with different grid sizes, and discretization techniques, we conclude our results are comparable to previous computations.

The features of the flow of BFS on the 600×32 grid closely matched those given in [43, 4]. The length of the recirculation region below the step was somewhat shorter in our study (roughly 4.5 vs. 6.10 in [4]), but U-direction flow speeds at the $x = 7$ and $x = 15$ points differed by less than 7% of the maximum flow speed. However, it should be noted that Gartling used a finer, adaptive finite element mesh, and central weighting.

The variation between our results and those in the cited studies can be accounted for by differences in the grid size, discretization (finite elements on a variable grid versus our use of the finite volume formulation), and in upwind weighting techniques (or absence thereof).

7.4 Notes on Aspect Ratios

The BFS problem provides a case in point whereby we can emphasize the importance of the ordering of the unknowns for successful application of preconditioned-conjugate-gradient methods to Navier-Stokes problems. The BFS problem has the most extreme physical aspect ratio of the tests at hand, and previous studies have shown that when anisotropies arise in a problem (for example, from a large difference in x and y direction coefficients, or from large control-volume aspect ratios) more attention needs to be paid to the matrix ordering [12, 3, 25, 24].

For this section, two new orderings are introduced. The first is "natural" in the x, then y direction (NatX). This orders the equations by pressure centered cell along the x-axis first, grouping the u, v, and p unknowns together in a small block for each cell. The second ordering, NatY, follows the idea of NatX, only ordering in the y direction first. More details on these orderings can be found in [3, 25, 24]. NatX and NatY allow us to compare graph-based orderings that follow or run against the anisotropy of the BFS problem. The NatX and NatY orderings are combined with pre-elimination during the solution, since there is no mechanism inherent in NatX and NatY that prevents a zero pivot on the diagonal of the matrices.

Table 13 lists the results for the BFS run on two grids, with the four different orderings. When the control volume aspect ratio was favourable (1.5:1), MUM ordering produced the best timing results. For problems with finite volume cells with large x-dimensions compared to the y-dimension, this creates a strong (discrete) coupling in the y-direction. The work of [25, 24] indicates that an effective ordering for this situation is produced by first ordering along the x-direction first (which is somewhat counter intuitive) and then in the y-direction, i.e. NatX ordering. When the control volume aspect ratio became more extreme (24:1), NatX ordering, which follows the anisotropy of the problem, produced by far the best solution time. This confirms the findings of [25, 24]. Note that as pointed out in [24], MUM ordering is unable to detect anisotropies (compared to MDF ordering [25]).

When the Pre+RCM ordering failed, it did so due to a small diagonal pivot produced in the

ILU factorization. (Note that the diagonal pivots measured for Table 13 were normalized using the maximum absolute value in the row of the pivot.) Indeed, when any ordering method completely failed to produce a solution, a small pivot had been encountered. A small pivot causes a rapid numerical growth in the matrix entries, leading to the failure of the iterative method.

Further investigation into the challenges posed by aspect ratios in the solution of PDE's is under way. Suffice to say for the time being that proper matrix ordering appears to be the solution to the problems produced by anisotropies induced by finite volume aspect ratio problems.

It is worthwhile to point out, as indicated in Table 13 that the aspect ratio problem affects the linear iterative equation solution. The number of nonlinear iterations is actually smaller for the unfavourable aspect ratio problem.

8 Conclusion

The FC+FN approach to the solution of the non-linear problem presented by the steady-state, incompressible, Navier-Stokes equations is designed to take advantage of the best aspects of both the frozen coefficient and full Newton iteration schemes. In general, the frozen coefficient linear equations matrices are easy to solve. The frozen coefficient method alone shows adequate initial convergence, which rapidly becomes slow. Provided that the actual solution to the discrete problem was not bifurcating, and that the linearized FC equations were solved to sufficiently small tolerances, the FC method was an extremely robust method for solution of the nonlinear equations. The full Newton method cannot, in general, be used from a initial zero flow field without pseudo-timestepping, but can be used on the steady-state equations after the frozen coefficient method has partially resolved the solution. The full Newton linear equation matrices are generally somewhat more costly to solve, but fewer solutions are required since the method, once started within its radius of convergence, converges extremely rapidly.

Consequently, it would appear that the FC+FN method is superior to a frozen coefficient (FC) method alone, if large non-linear residual reductions are required. If only a small non-linear residual reduction is necessary, there are some situations where the FC technique alone might be appropriate. The full Newton (FN) method, with pseudo-timestepping to ensure convergence, was always a poor third in all our tests compared to FC or FC+FN.

We have also presented a robust approach for solving the linear equations. Of particular note is that the incompletely factored frozen coefficient formulation of the linear equations is a good preconditioner for the full Newton Jacobian. The frozen coefficient matrix is a more easily (incompletely) factored preconditioner, and has been shown to produce more rapid convergence for the full Newton iteration stage of the FC+FN method.

MUM ordering is the best of the tested matrix orderings if the problem domain is particularly long in one direction, relative to the characteristic length used to set the Reynolds number of the problem. Otherwise, a graph based ordering, coupled with the pre-elimination of the conservation of mass equation to eliminate zero pivots on the diagonal of the matrix, proved most effective. We have noted that ordering methods can overcome the problem of small diagonal pivots induced by a high control volume aspect ratio. Note that high aspect ratio appears to affect the linear iteration much more than the nonlinear iteration. More research on this problem will be presented in future papers. We emphasize once again that the question of the ordering of the unknowns is crucial for

application of PCG methods to Navier-Stokes equations.

We have presented five of over thirty problem geometries on which this method was tested. The reader will note that the FC+FN solution method coupled with a PCG solver for the linear equations is entirely independent of the problem geometry (with the exception of the above comment on ordering of the unknowns), and that internal boundaries and fine flow details (the Asym problem, for example) are resolved as easily as geometries with coarser flow features (such as the driven cavity).

Performance of the FC+FN non-linear and linear methods together exceeds the previously expected limits. Instead of the $\mathcal{O}(N^{3/2})$ generally expected in two-dimensional elliptic problems, our method obtained performance between $\mathcal{O}(N^{1.20})$ and $\mathcal{O}(N^{1.38})$ for all problems presented (assuming that the best ordering was selected). Note that other studies ([24]) have indicated that these methods should apply to finite element discretizations on unstructured grids with equal effectiveness. Since the generally expected performance of iterative methods on three-dimensional domains is $\mathcal{O}(N^{4/3})$, we expect the performance of this method to improve in three dimensions. This method has been tested with the other upwind weighting techniques presented in [1], and was equally as effective.

Convergence is rapidly obtained to an arbitrary precision through the use of Newton iteration at the final stage, and the solutions obtained have been shown to be accurate. The only problem dependent parameters are the residual reduction required for switching from FC to FN iteration, and the ordering for the unknowns for the linear solve. Any estimate for the former parameter can be used, since the algorithm can recover (by continuing FC iteration) if the FN iteration begins to diverge. The matrix ordering question is more difficult, but a completely automatic method (MUM ordering) can be used, which is very robust. However, it is certainly possible to use orderings which outperform MUM ordering in some circumstances.

Further research is anticipated to extend the FC+FN approach and the accompanying linear methods to three dimensional flows, and to irregular grids.

A Nomenclature

3Cham	Three Chamber problem
AllFC	Non-linear method using only frozen coefficient iteration
AllFN	Non-linear method using only Newton iteration
Asym	Asymmetric Flow Chamber problem
BFS	Backwards Facing Step problem
DC	Driven Cavity problem
FC	Frozen Coefficient
FC+FN	Non-linear method using frozen coefficient, then Newton iteration
FN	Full Newton
ILU(n)	Incomplete Lower/Upper factorization keeping n levels fill
LEM	Linearized Equation Matrix
MDF	Minimum Discarded Fill (matrix ordering)
MUM	Minimum Update Matrix (matrix ordering)
NatX	Natural, grid-wise matrix ordering (X direction first)
NatY	Natural, grid-wise matrix ordering (Y direction first)
PCG	Preconditioned Conjugate Gradients
Pre+	Pre-elimination performed with the given ordering
RCM	Reverse Cuthill-McKee (matrix ordering)
Symm	Symmetric Flow Chamber problem

References

[1] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow.* Hemisphere Publishing Corporation, 1980.

[2] D. Howard, W. M. Connolley, and J. S. Rollett. Unsymmetric conjugate gradient methods and sparse direct methods in finite element flow simulation. *International Journal for Numerical Methods in Fluids*, 10:925–945, 1990.

[3] P. Chin, E. F. D'Azevedo, P. A. Forsyth, and W.-P. Tang. Preconditioned conjugate gradient methods for the incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 15:273–295, 1992.

[4] D. K. Gartling. A test problem for outflow boundary conditions–flow over a backward facing step. *Internat. J. Numer. Methods Fluids*, 11:953–967, 1990.

[5] M. E. Braaten and S. V. Patankar. A block-corrected subdomain solution procedure for recirculating flow calculations. *Num. Heat Transfer, Part B*, 15:1–20, 1989.

[6] W. G. Habashi and J. Strigberger. Application of AF-preconditioned conjugate gradient-like methods to the computation of unsteady incompressible viscous flow. In *Intern. Conf. in Num. Meth. in Laminar and Turbulent Flow*, pages 1537–1547, Swansea, 1991. Pineridge Press.

[7] M. P. Robichaud and P. A. Tanguy. Finite element solution of three-dimensional incompressible fluid flow problems by a preconditioned residual method. *International Journal for Numerical Methods in Engineering*, 24:447–457, 1987.

[8] S. A. Jordan. An iterative scheme for numerical solution of steady incompressible viscous flows. *Computers and Fluids*, 21(4):503–517, 1992.

[9] R. Hunt. The numerical solution of the laminar flow in a constricted channel at moderately high reynolds number using newton iteration. *International Journal for Numerical Methods in Fluids*, 11:247–259, 1990.

[10] G. F. Carey, K. C. Wang, and W. D. Joubert. Performance of iterative methods for newtonian and generalized newtonian flows. *International Journal for Numerical Methods in Fluids*, 9:127–150, 1989.

[11] S. P. Vanka. Block-implicit calculation of steady turbulent recirculating flows. *Int. J. Heat Mass Transfer*, 28(11):2093–2103, 1985.

[12] I. S. Duff and G. A. Meurant. The effect of ordering on preconditioned conjugate gradients. *BIT: Nordisk Tijdskrift for Informationbehandling*, 29:635–657, 1989.

[13] O. Dahl and S. O. 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. *International Journal for Numerical Methods in Fluids*, 15:525–544, 1992.

[14] P. F. Galpin and G. D. Raithby. Treatment of non-linearities in the numerical solution of the incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 6:409–426, 1986.

[15] J. W. MacArthur and S. V. Patankar. Robust semidirect finite difference methods for solving the Navier-Stokes and energy equations. *International Journal for Numerical Methods in Fluids*, 9:325–340, 1989.

[16] S. P. Vanka. Block-implicit multigrid solution of Navier-Stokes in primitive variables. *Journal of Computational Physics*, 65:138–158, 1986.

[17] E. Dick. A multigrid method for steady incompressible Navier-Stokes equations based on vector-flux splitting. In S. F. McCormick, editor, *Multigrid Methods*. SIAM, Philadelphia, 1987.

[18] R. D. Lonsdale. An algebraic multigrid scheme for solving the Navier-Stokes equations on unstructured meshes. In *Intern. Conf. in Num. Meth. in Laminar and Turbulent Flow*, pages 1432–1442, Swansea, 1991. Pineridge Press.

[19] C. P. Thompson and G. K. Leaf. Application of a multigrid method to a bouyancy-induced flow problem. In S. F. McCormick, editor, *Multigrid Methods*. SIAM, Philadelphia, 1987.

[20] P. Sonneveld. CGS, a fast Lanczos-type solver for nonsymmetric systems. *SIAM J. Sci. Statist. Comput.*, 10:36–52, 1989.

[21] H. A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13(2):631–644, March 1992.

[22] M. Buffat. Simulation of two- and three-dimensional internal subsonic flows using a finite-element method. *International Journal for Numerical Methods in Fluids*, 12:683–704, 1991.

[23] Laura C. Dutto. The effect of ordering on preconditioned GMRES algorithm, for solving the compressible Navier-Stokes equations. Research report, Université de Montréal, CRM, February 1992. to appear in International Journal for Numerical Methods in Engineering.

[24] E. F. D'Azevedo, P. A. Forsyth, and W. P. Tang. Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems. *SIAM J. Matrix Anal. Appl.*, 13(3):944–961, July 1992.

[25] E. F. D'Azevedo, P. A. Forsyth, and W.-P. Tang. Towards a cost-effective ILU preconditioner with high level fill. *BIT*, 32:442–463, 1992.

[26] E. O. Einset and K. F. Jensen. A finite element solution of three-dimensional mixed convection gas flows in horizontal channels using preconditioned iterative matrix methods. *International Journal for Numerical Methods in Fluids*, 14:817–841, 1992.

[27] M. Zedan and G. E. Schneider. A coupled strongly implicit procedure for velocity and pressure computation in fluid flow problems. *Num. Heat Transfer*, 8:537–557, 1985.

[28] P. Chin and P. A. Forsyth. A comparison of GMRES and CGSTAB preconditioning for incompressible viscous flow. *J. Comp. Appl. Math*, 46:415–426, 1993.

[29] O. Axelsson. *Solution of Linear Systems of Equations: Iterative Methods.* Springer-Verlag, Berlin, Heidelber, New York, 1977.

[30] E. F. D'Azevedo, P. A. Forsyth, and W.-P. Tang. Drop tolerance preconditioning for incompressible viscous flow. *Intern. J. Computer Math.*, 44:301–312, 1992.

[31] E. F. D'Azevedo, P. A. Forsyth, and W.-P. Tang. Two variants of minimum discarded fill ordering. In R. Beauwens and P. de Groen, editors, *Proc. IMACS Intern. Symp. on Iterative Methods in Linear Algebra*, pages 603–612. North-Holland, 1992.

[32] A. George and J. W. H. Liu. *Computer Solutions of large sparse positive-definite systems,.* Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

[33] M. F. Peeters, W. G. Habashi, and B. Q. Nguyen. Finite element solution of the incompressible Navier-Stokes equations by a helmholtz velocity decomposition. *International Journal for Numerical Methods in Fluids*, 13:135–144, 1991.

[34] M. P. Robichaud and W. G. Habashi. Iterative methods for the steady-state compressible Navier-Stokes equations. Canadian Appl. Math Annual Meeting, Ottawa, May 1991.

[35] U. Ghia, K. N. Ghia, and C. T. Shin. High-re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 48:387–411, 1982.

[36] B. F. Armaly, F. Durst, J. C. F. Pereira, and B. Schönung. Experimental and theoretical investigation of backward-facing step flow. *J. Fluid Mech.*, 127:473–496, 1983.

[37] W. F. Tinney and J. W. Walker. Direct solutions of sparse network equations by optimally ordered triangular factorization. *Proc. IEEE*, 55:1801–1809, 1967.

[38] O. Axelsson and V. A. Barker. *Finite Element Solution of Boundary Value Problems, Theory and Computation.* Academic Press, 1984.

[39] H. P. Langtangen. Conjugate gradient methods and ILU preconditioning of non-symmetric matrix systems with arbitrary sparsity patterns. *International Journal for Numerical Methods in Fluids*, 9:213–233, 1989.

[40] M. C. Thompson and J. H. Ferziger. An adaptive multigrid technique for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 82:94–121, 1989.

[41] P. M. Gresho, D. K. Gartling, K. A. Cliff, T. J. Garratt, A. Spence, K. H. Winters, J. W. Goodrich, and J. R. Torczynski. Is the steady viscous incompressible 2D flow over a backward-facing step at Re=800 stable? Preprint UCRL-JC-113824, Lawrence Livermore National Laboratory, February 1993. To appear in Intl. J. Num. Meth. Fluids.

[42] P. M. Gresho, S. T. Chan, R. L. Lee, and C. D. Upson. A modified finite element method for solving the time-dependent incompressible Navier-stokes equations. Part 2: Applications. *International Journal for Numerical Methods in Fluids*, 4:619–640, 1984.

[43] J. L. Sohn. Evaluation of FIDAP on some classical laminar and turbulent benchmarks. *International Journal for Numerical Methods in Fluids*, 8:1469–1490, 1988.

[44] C.-H. Bruneau and C. Jouron. An efficient scheme for solving steady incompressible Navier-stokes Equations. *Journal of Computational Physics*, 89:389–413, 1990.

Table 1: Non-linear methods compared over five test problems

	Non-linear Method								
	All FC			FN+FC			All FN		
Test	NLI	Time	NLRed	NLI	Time	NLRed	NLI	Time	NLRed
DC	15	18.57	1.135e-05	8	11.40	7.865e-09	41	50.20	6.292e-08
Symm	13	7.59	4.970e-06	8	5.22	2.571e-09	30	23.55	2.302e-09
Asym	15	7.96	1.240e-05	9	5.26	2.491e-09	27	18.71	2.296e-09
3Cham	20	19.11	1.366e-05	11	11.75	5.882e-09	38	61.42	1.801e-07
BFS	120	153.71	3.942e-05†	68	93.86	3.342e-09	40	156.27	6.667e-03†

NLI *is number of non-linear iterations.*
Time *is the CPU time in minutes.*
NLRed *is the non-linear residual reduction at convergence.*
† *Failed to converge within the 150 minute CPU time limit.*

The DC, Asym, and 3Cham tests were performed over an 80×80 *grid at Re=1000, whereas the BFS test was done over a* 400×20 *grid with Re=800. All tests were performed with the MUM ordering method.*

Table 2: A comparison between direct and iterative matrix solvers on the Driven Cavity problem

	MD + Direct		Iterative (Pre+RCM)		Iterative (MUM)	
Grid Size	Total Time	Avg. Per Matrix	Total Time	Avg. Per Matrix	Total Time	Avg. Per Matrix
20×20	0.57	0.05	0.55	0.05	0.41	0.03
30×30	2.33	0.21	1.46	0.13	1.08	0.10
40×40	7.04	0.64	3.20	0.29	2.42	0.22
60×60	32.06	2.91	9.47	0.86	6.61	0.60

The DC problem is solved here at Re=1000. Note that these times are in CPU minutes. In this series of tests, the direct method is $\mathcal{O}(N^{1.84})$.

Table 3: FC and FN as Preconditioners for the FN stage of FC+FN

	Time for FN Stage with Preconditioner Type			
	FC		FN	
Test	Total	per Matrix	Total	per Matrix
DC	3.69	1.23	4.37	1.46
BFS	10.47	2.62	20.99	5.25

Times are in CPU minutes. The DC problem was run on an 80×80 *grid at Re=1000. The BFS problem was run on a* 400×20 *grid at Re=800. In all tests, MUM ordering was used at ILU(3).*

Table 4: Times by phase and grid size for Pre+RCM ordered tests with FC+FN method

	Grid Size, with Time by Phase of FC+FN with Pre+RCM								
	Coarse			Medium			Fine		
Test	Total	FC	FN	Total	FC	FN	Total	FC	FN
DC	2.67	1.85	0.82	16.00	8.85	7.16	93.69	54.86	38.83
Symm	1.18	0.75	0.43	7.80	4.59	3.21	53.06	34.23	18.83
Asym	1.37	0.88	0.49	8.13	5.11	3.02	43.06	26.24	16.83
3Cham	3.26	2.10	1.15	19.55	12.98	6.57	118.48	85.10	33.38
BFS	13.72	11.12	2.60	158.18	139.33	18.85	630.30	556.87	73.43

Times are in CPU minutes. Tests DC, Asym, and 3Cham were run with Re=1000, while the BFS test was run with Re=800. ILU(5) was used.

Table 5: Iterations by phase and grid size for Pre+RCM ordered tests with FC+FN method

Test	Grid Size, with Iterations by Phase of FC+FN with Pre+RCM											
	Coarse				Medium				Fine			
	FC		FN		FC		FN		FC		FN	
	NLI	Avg.	NLI	Avg.	NLI	Avg.	NLI	Avg.	NLI	Avg.	NLI	Avg.
DC	6	14.8	2	27.0	5	24.8	3	38.3	5	49.6	3	62.3
Symm	5	10.0	2	21.0	5	17.8	3	24.3	5	40.0	3	37.3
Asym	6	10.5	2	24.0	6	17.2	3	20.7	5	32.0	3	36.7
3Cham	8	16.0	3	26.3	8	24.8	3	37.3	9	43.3	3	53.7
BFS	36	15.9	3	61.7	64	31.5	4	80.8	71	51.0	4	133.2

NLI *denotes number of non-linear iterations that phase.*
Avg *denotes the average number of linear iterations per non-linear iteration for that phase.*

Times are in CPU minutes. Tests DC, Asym, and 3Cham were run with Re=1000, while the BFS test was run with Re=800. ILU(5) was used.

Table 6: Time Complexity by phase and grid size for Pre+RCM ordered tests with FC+FN method

Test	Time complexity exponent		
	Overall	FC Phase	FN Phase
DC	1.27	1.32	1.22
Symm	1.38	1.45	1.28
Asym	1.20	1.18	1.24
3Cham	1.30	1.36	1.17
BFS	1.58	1.58	1.55

The figures in this table are the exponent α of the order expression $\mathcal{O}(N^\alpha)$ where N is the number of grid pressure cells. The measurement is taken from the medium and fine grids listed in table 4.

Table 7: Times by phase and grid size for MUM ordered tests with FC+FN method

	Grid Size, with Time by Phase of FC+FN with MUM								
	Coarse			Medium			Fine		
Test	Total	FC	FN	Total	FC	FN	Total	FC	FN
DC	1.53	1.19	0.34	11.40	6.98	4.42	91.99	59.99	32.00
Symm	0.84	0.60	0.24	5.22	3.42	1.79	43.91	28.91	15.00
Asym	0.82	0.60	0.22	5.26	3.65	1.60	37.27	24.04	13.23
3Cham	1.94	1.42	0.52	11.75	8.27	3.48	106.87	82.78	24.09
BFS	7.30	6.63	0.67	93.86	85.13	8.73	303.93	269.14	34.79

Times are in CPU minutes. Tests DC, Asym, and 3Cham were run with Re=1000, while the BFS test was run with Re=800. ILU(4) was used.

Table 8: Iterations by phase and grid size for MUM ordered tests with FC+FN method

	Grid Size, with Iterations by Phase of FC+FN with MUM											
	Coarse				Medium				Fine			
Test	FC		FN		FC		FN		FC		FN	
	NLI	Avg.	NLI	Avg.	NLI	Avg.	NLI	Avg.	NLI	Avg.	NLI	Avg.
DC	6	18.8	2	19.5	5	30.0	3	39.0	5	65.2	3	64.0
Symm	5	14.2	2	19.5	5	17.2	3	18.3	5	39.0	3	39.0
Asym	6	12.0	2	18.0	6	15.8	3	17.3	5	30.8	3	35.0
3Cham	8	18.1	3	19.7	8	22.2	3	29.7	9	51.6	3	48.0
BFS	36	14.1	3	19.0	64	28.6	4	52.5	71	36.0	4	93.8

NLI *denotes number of non-linear iterations that phase.*
Avg *denotes the average number of linear iterations per non-linear iteration for that phase.*

Times are in CPU minutes. Tests DC, Asym, and 3Cham were run with Re=1000, while the BFS test was run with Re=800. ILU(4) was used.

Table 9: Time Complexity by phase and grid size for MUM ordered tests with FC+FN method

Test	Time complexity exponent Overall	FC Phase	FN Phase
DC	1.51	1.55	1.43
Symm	1.54	1.54	1.53
Asym	1.41	1.36	1.52
3Cham	1.59	1.66	1.39
BFS	1.34	1.31	1.58

The figures in this table are the exponent α of the order expression $\mathcal{O}(N^{\alpha})$ where N is the number of grid pressure cells. The measurement is taken from the medium and fine grids listed in table 7.

Table 10: Solution time, and iteration count for Pre+RCM ordered tests over varying Reynolds numbers

	Reynolds Number												
	100			500			800			1000			
Tests	Time	NLI	Avg	Time	NLI	Avg	Time	NLI	Avg	Time	NLI	Avg	Ord
DC	7.40	5	21.0	11.01	7	23.6	-	-	-	16.00	8	29.9	0.54
Symm	4.52	5	17.4	6.63	7	19.4	-	-	-	7.80	8	20.2	0.23
Asym	3.70	5	14.0	5.53	7	16.6	-	-	-	8.13	9	18.3	0.56
3Cham	7.83	6	19.5	14.31	10	23.5	-	-	-	19.55	11	28.2	0.45
BFS	39.89	15	39.3	40.12	15	39.3	158.18	68	34.4	-	-	-	2.92

\- *(dashes) indicate that the test was not run at that Reynolds number.*
Time *is total solution time in CPU minutes for FC+FN solve.*
NLI *is total non-linear iterations, FC and FN.*
Avg *is average linear iterations per non-linear iteration, FC and FN.*
Ord *is the exponent (time complexity) of the change in time for the last two tests with respect to Reynolds number.*
All solutions are for the medium grid size (80×80 *or* 400×20*) using the FC+FN method with Pre+RCM ordering, and ILU(5).*

Table 11: Solution time, and iteration count for MUM ordered tests over varying Reynolds numbers

	Reynolds Number												
	100			500			800			1000			
Tests	Time	NLI	Avg	Time	NLI	Avg	Time	NLI	Avg	Time	NLI	Avg	Ord
DC	10.26	5	30.6	11.08	7	30.0	-	-	-	11.40	8	33.4	0.04
Symm	5.64	5	19.8	6.31	7	20.9	-	-	-	5.22	8	17.6	-0.28
Asym	5.30	5	19.0	5.23	7	16.7	-	-	-	5.26	9	16.3	0.01
3Cham	12.42	6	29.7	13.86	10	25.5	-	-	-	11.75	11	24.3	-0.24
BFS	32.45	15	36.7	30.56	15	36.7	93.86	68	30.0	-	-	-	2.39

\- *(dashes) indicate that the test was not run at that Reynolds number.*
Time *is total solution time in CPU minutes for FC+FN solve.*
NLI *is total non-linear iterations, FC and FN.*
Avg *is average linear iterations per non-linear iteration, FC and FN.*
Ord *is the exponent (time complexity) of the change in time for the last two tests.*
All solutions are for the medium grid size (80×80 *or* 400×20*) using the FC+FN method with MUM ordering, and ILU(5).*

Table 12: Comparison of two features of the Driven Cavity at Re=1000 with other studies

Reference	Weighting	Grid Size	ψ_{min}	u_{min} on CL	y-location
Ghia *et. al.* [35]	†	257×257	-0.1179	-0.3829	0.1719
Gresho *et. al.* [42]	STU	129×129	-0.114	-0.375	0.160
Vanka [16]	Hybrid	321×321	-0.1173	-0.387	0.1734
Sohn [43]	STU	129×129	-0.0799	‡	‡
	Central	129×129	-0.1151	‡	‡
Thompson & Ferziger [40]	Power Law	256×256	-0.1167	‡	‡
	Central	128×128	-0.1178	‡	‡
Bruneau & Jouron [44]	§	256×256	-0.1163	-0.3764	0.1602
This study	Central	200×200	-0.1183	-0.3861	-0.1750
	Hybrid	200×200	-0.1182	-0.3852	-0.1750
	Power Law	200×200	-0.1154	-0.3726	-0.1750

†*indicates study used ψ-ω formulation of the Incompressible Navier-Stokes Equations.*
‡*indicates values not given explicitly.*
§*— see [44].*
Weighting *is the upwind differencing scheme. Power law, hybrid and central are explained in [1]. For other schemes, such as streamwise upwinding (STU), please refer to the cited papers.*
ψ_{min} *is the minimum value of the streamfunction at the center of the primary vortex of the driven cavity.*
u_{min} **on CL** *refers to the greatest negative x-direction velocity on the vertical centerline.*
y-**location** *is the vertical location of u_{min} on CL.*

Table 13: The BFS problem at various aspect ratios and grids

Grid Size		Ordering Method			
		Pre+RCM	MUM	Pre+NatX	Pre+NatY
Medium	Time	†	362.97	78.03	185.72
100x80	NLI	†	35	35	35
24:1	Avg	†	95.1	19.1	61.7
	Fill	1149652	2356908	1660495	1645753
	Min.	6.6E-06	4.6E-03	1.4E-02	1.5E-02
Medium	Time	158.18	93.86	304.13	257.55
400x20	NLI	68	68	68	68
1.5:1	Avg	34.4	30.0	54.0	49.5
	Fill	1084012	674754	1617175	1476613
	Min.	7.5E-03	7.3E-03	7.4E-03	9.5E-03

† *Indicates failed to converge due to small normalized diagonal.*
Grid Size *also gives the number of x and y direction pressure cells, and the ratio of pressure-centered control volume width to height.*
Time *is total solution time in CPU minutes.*
NLI *is total non-linear iterations.*
Avg *is the average number of linear (inner) iterations per non-linear iteration.*
Fill *is the total number of non-zero fill terms generated during the ILU factorization. ILU(4) was used for the MUM ordering, and ILU(5) used for the others.*
Min. *is the minimum diagonal encountered in the ILU factorization. All diagonals were normalized by the max. absolute value in their respective rows.*
All tests were on the BFS problem at Re=800, at the given grid sizes.

List of Figures

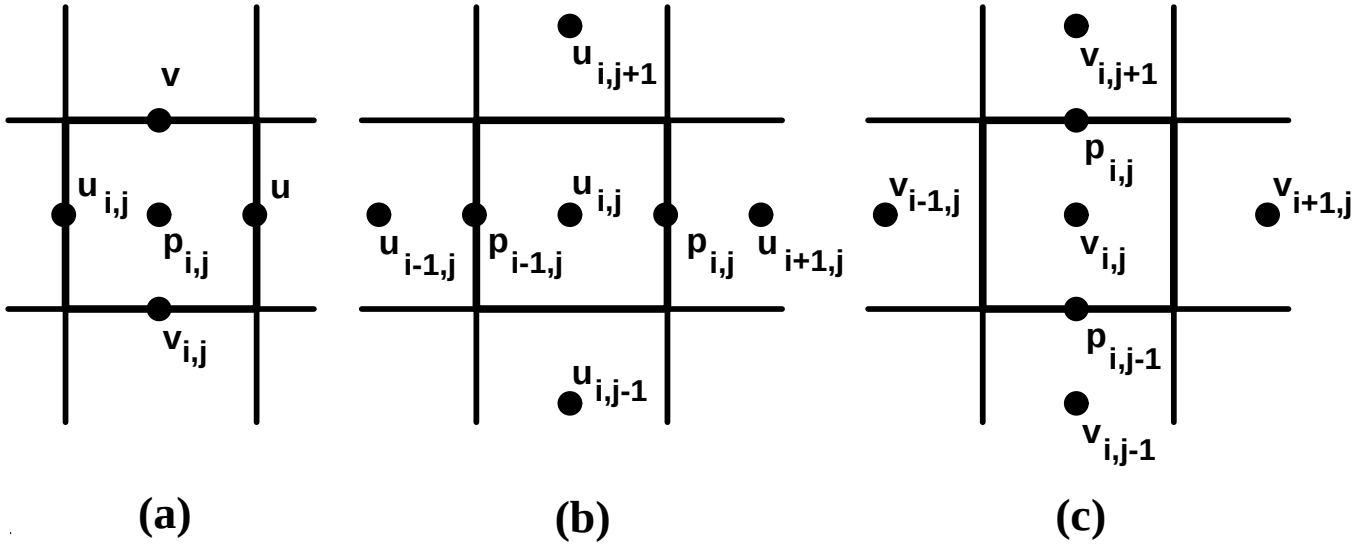

Figure 1: Finite Volumes containing p, u, and v

```
For all rows k in A
    (r_k)^p = r_k
Endfor

For all rows k ∉ {M}
    For all nonzero columns l in row k, A_{k,l}
        (A_{k,l})^p = A_{k,l}
    Endfor
Endfor

For all rows k ∈ {M}
    For all nonzero columns l in row k, A_{k,l}
        -(r_k)^p := -(r_k)^p + (A_{k,l}/A_{l,l}) (r_l)^p
        For all nonzero columns q in row l
            (A_{k,q})^p = A_{k,q} - (A_{k,l}/A_{l,l}) A_{l,q}
        Endfor
    Endfor
Endfor
```

Figure 2: Pre-elimination algorithm

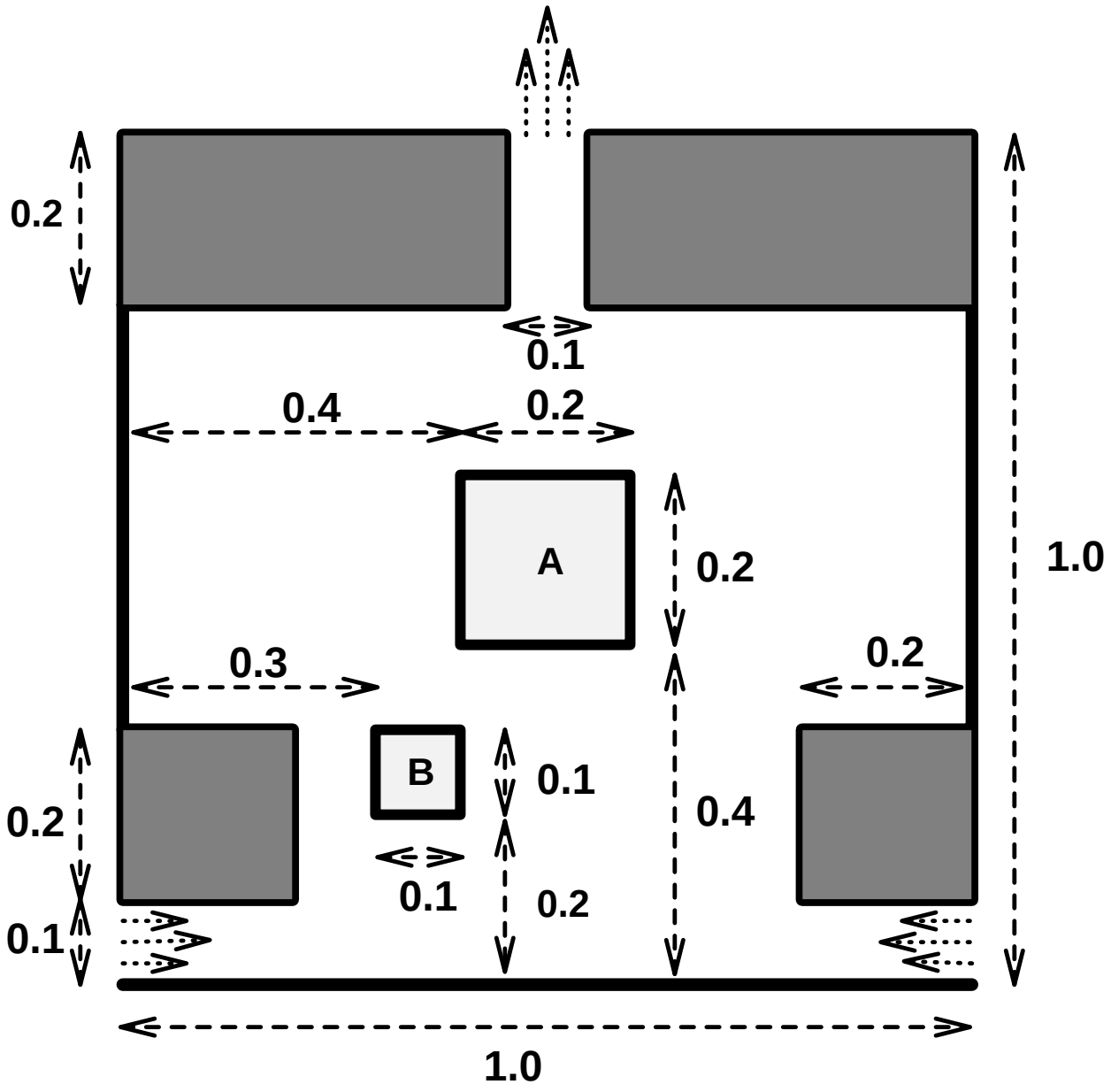

Figure 3: 2-In, 1-Out, Symmetric Flow Chamber and Asymmetrically Blocked Chamber. The Symmetric version of the problem omits the interior obstructions labelled A and B.

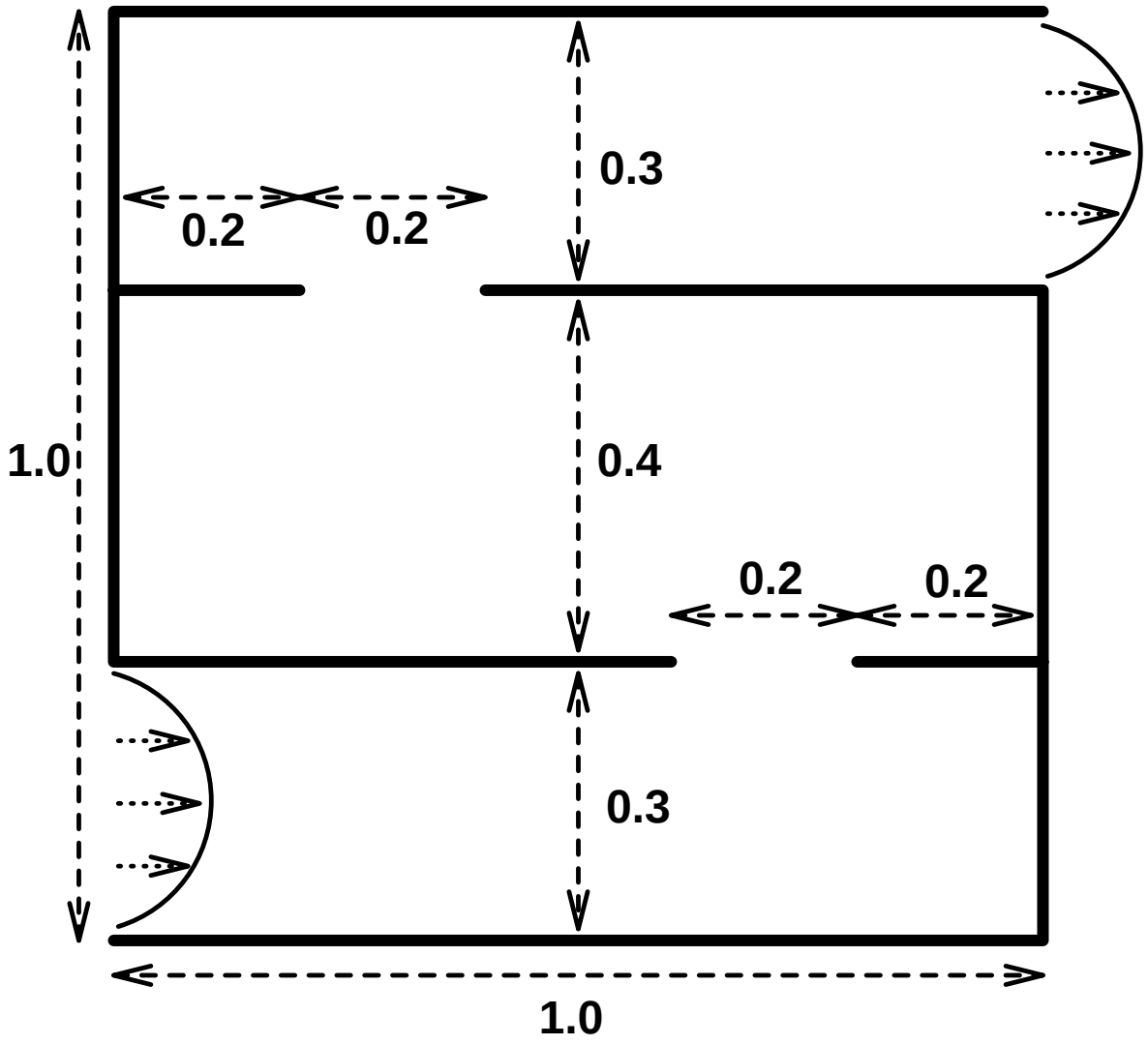

Figure 4: Three Chamber Problem

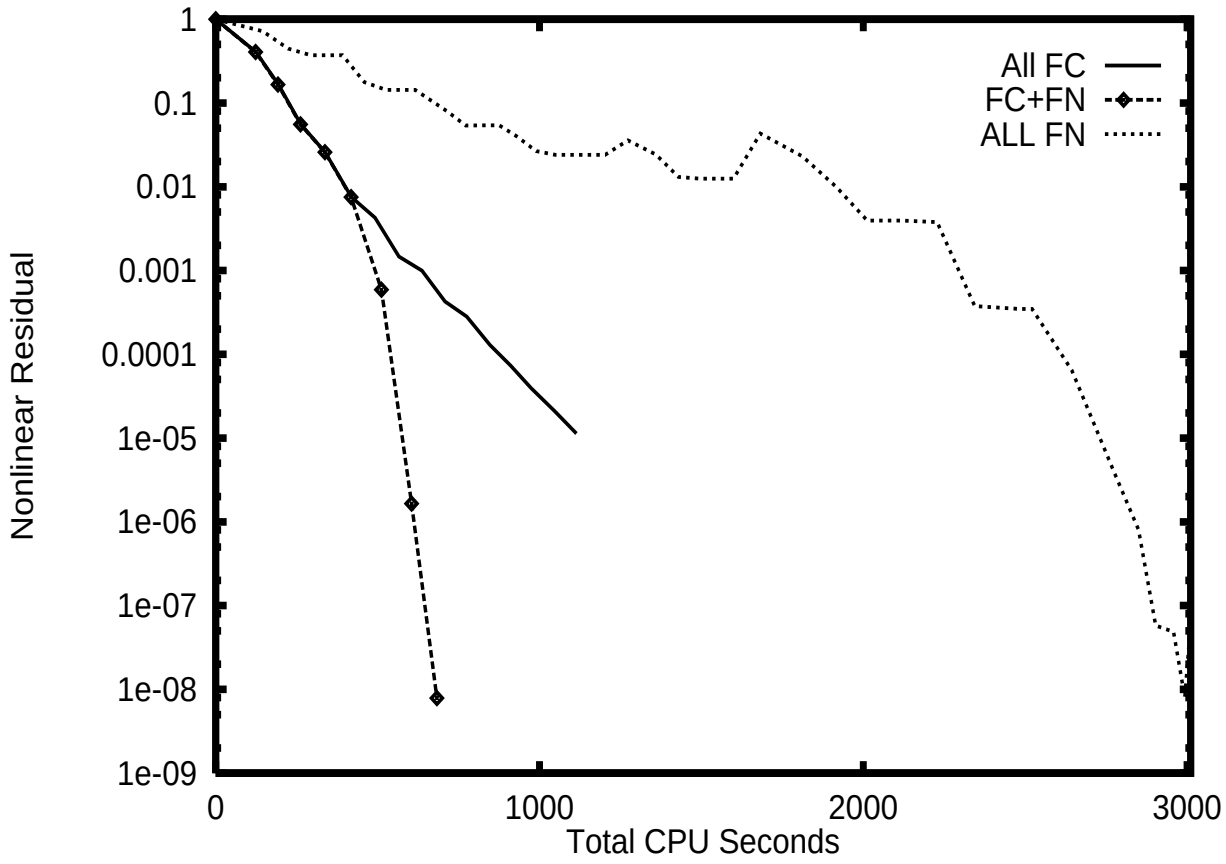

Figure 5: Non-linear convergence graph for problem DC

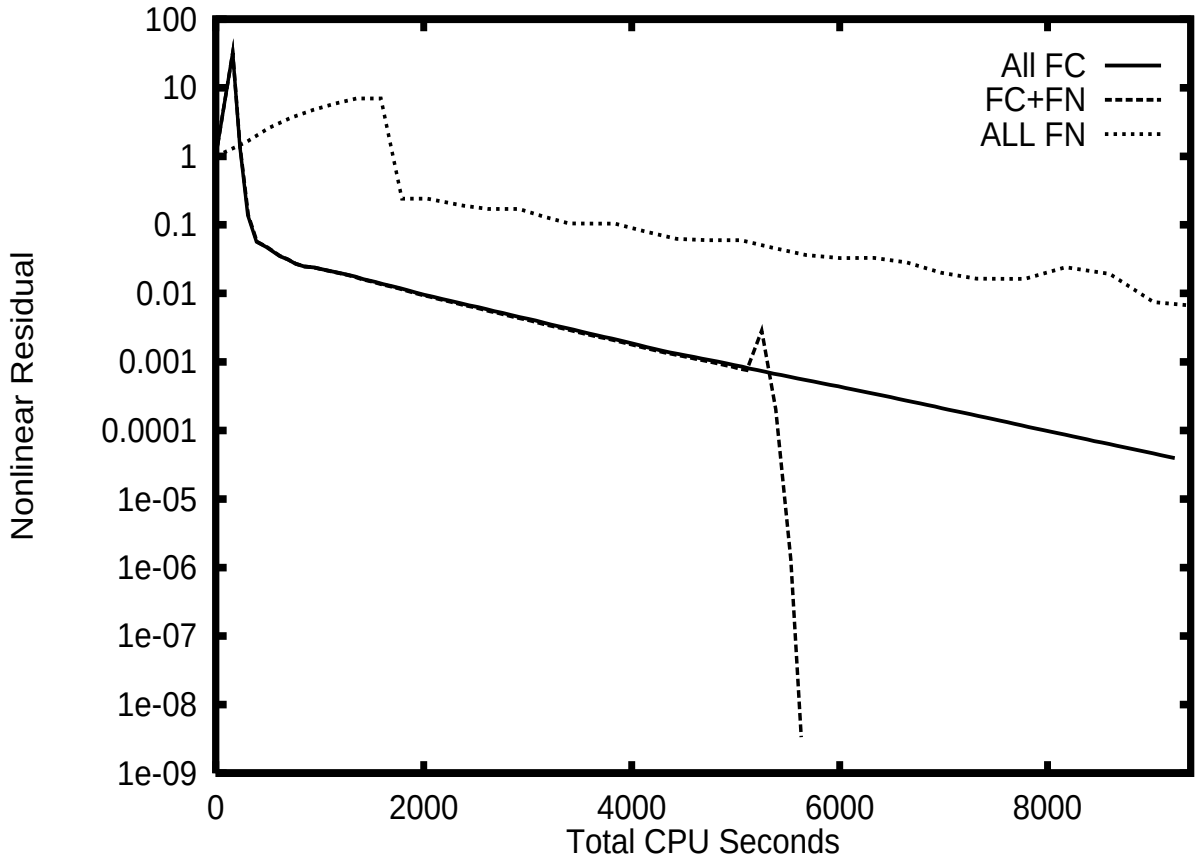

Figure 6: Non-linear convergence graph for problem BFS

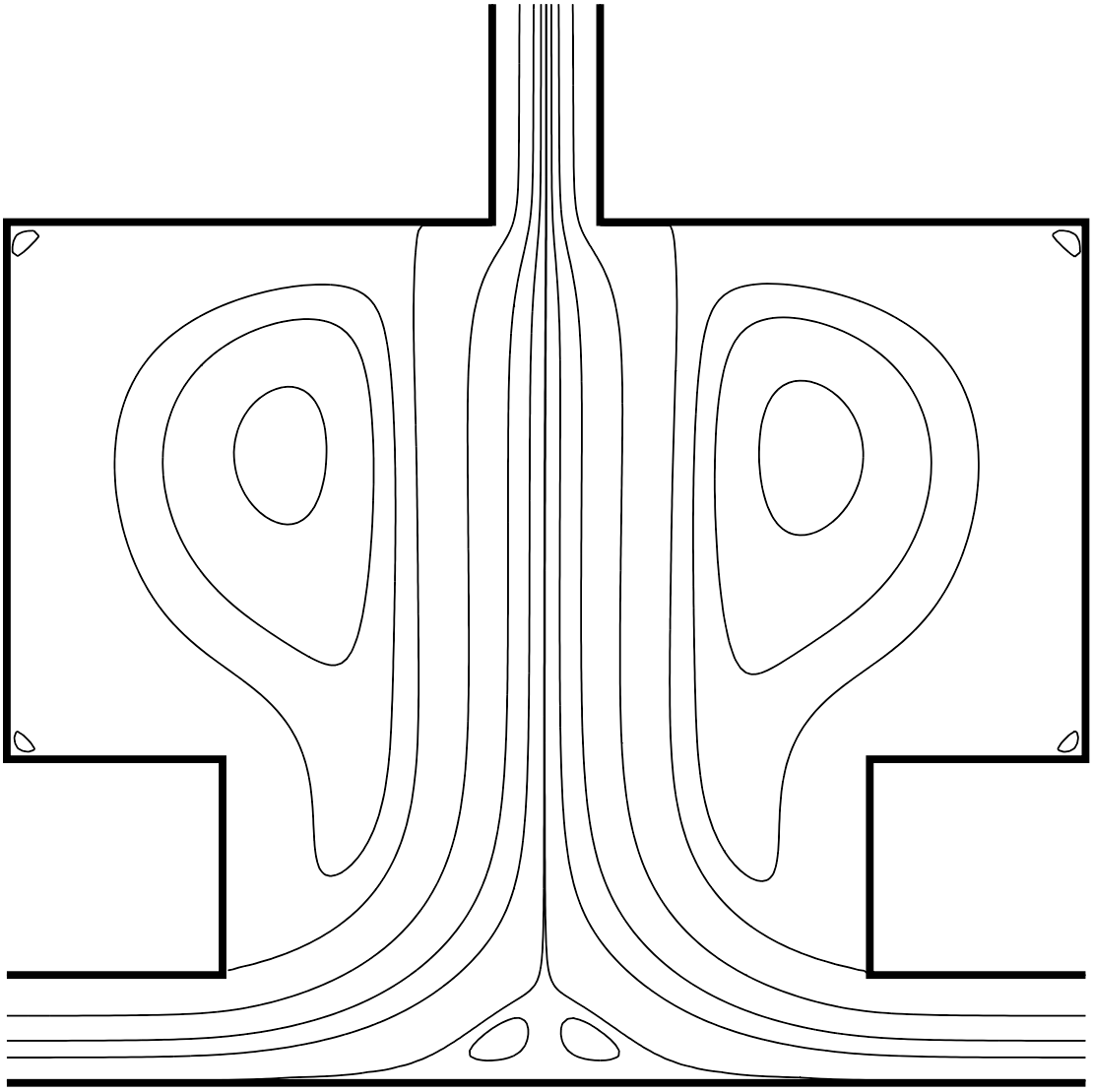

Figure 7: Streamfunction contours of Symm problem at Re=1000, on a 200×200 grid. Levels are ± 0.0001, ± 0.005, ± 0.012, ± 0.024, ± 0.0353773, ± 0.0386, ± 0.041, and ± 0.044.

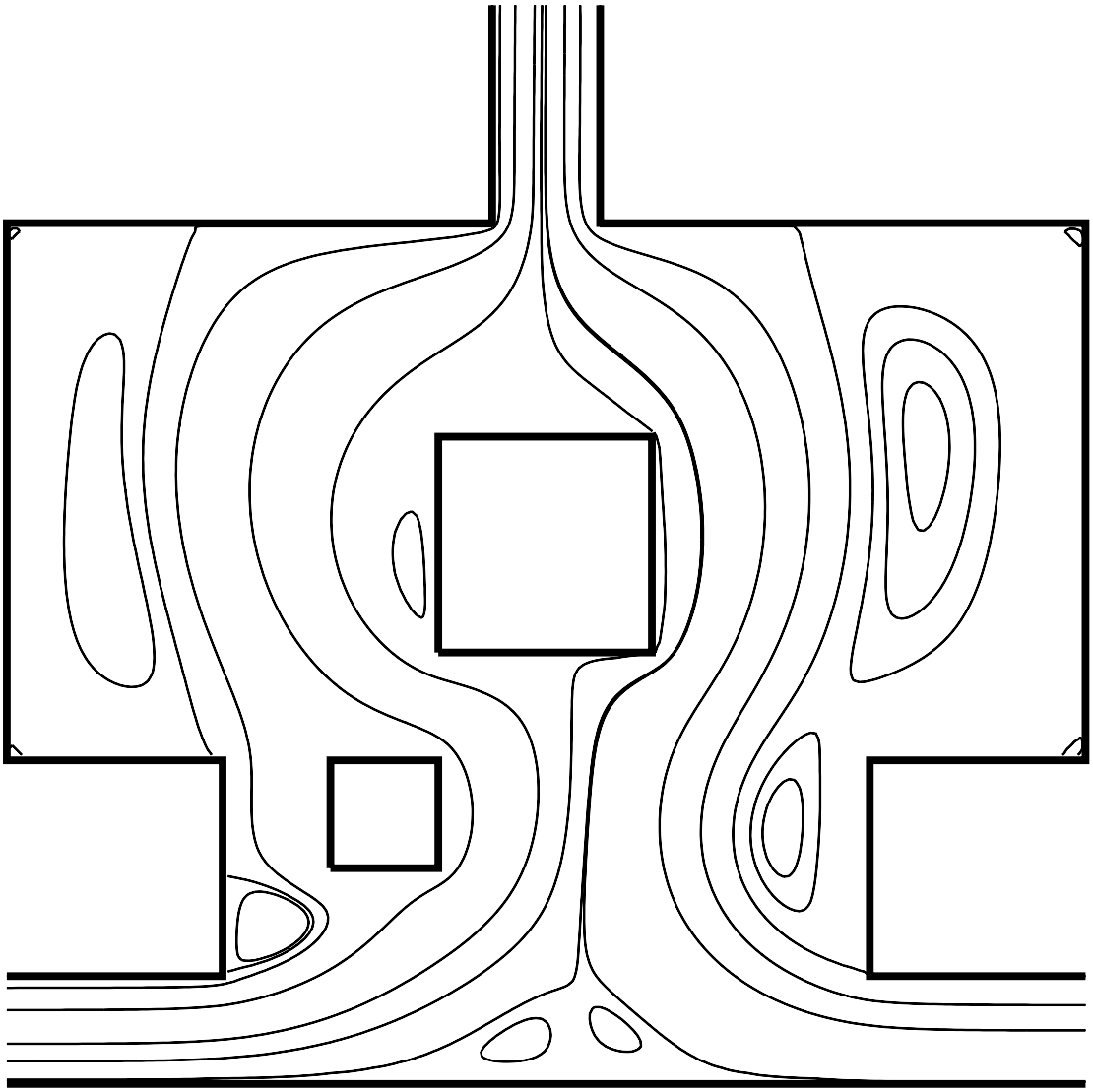

Figure 8: Streamfunction contours of Asym problem at Re=1000, on a 200×200 grid. Levels are -0.0001, -0.0175, -0.0290, -0.03537745, 0.03537749, -0.0374, -0.0382, -0.0388, 0.0002, 0.0041, 0.0111, 0.0270, 0.0342, and 0.0357.

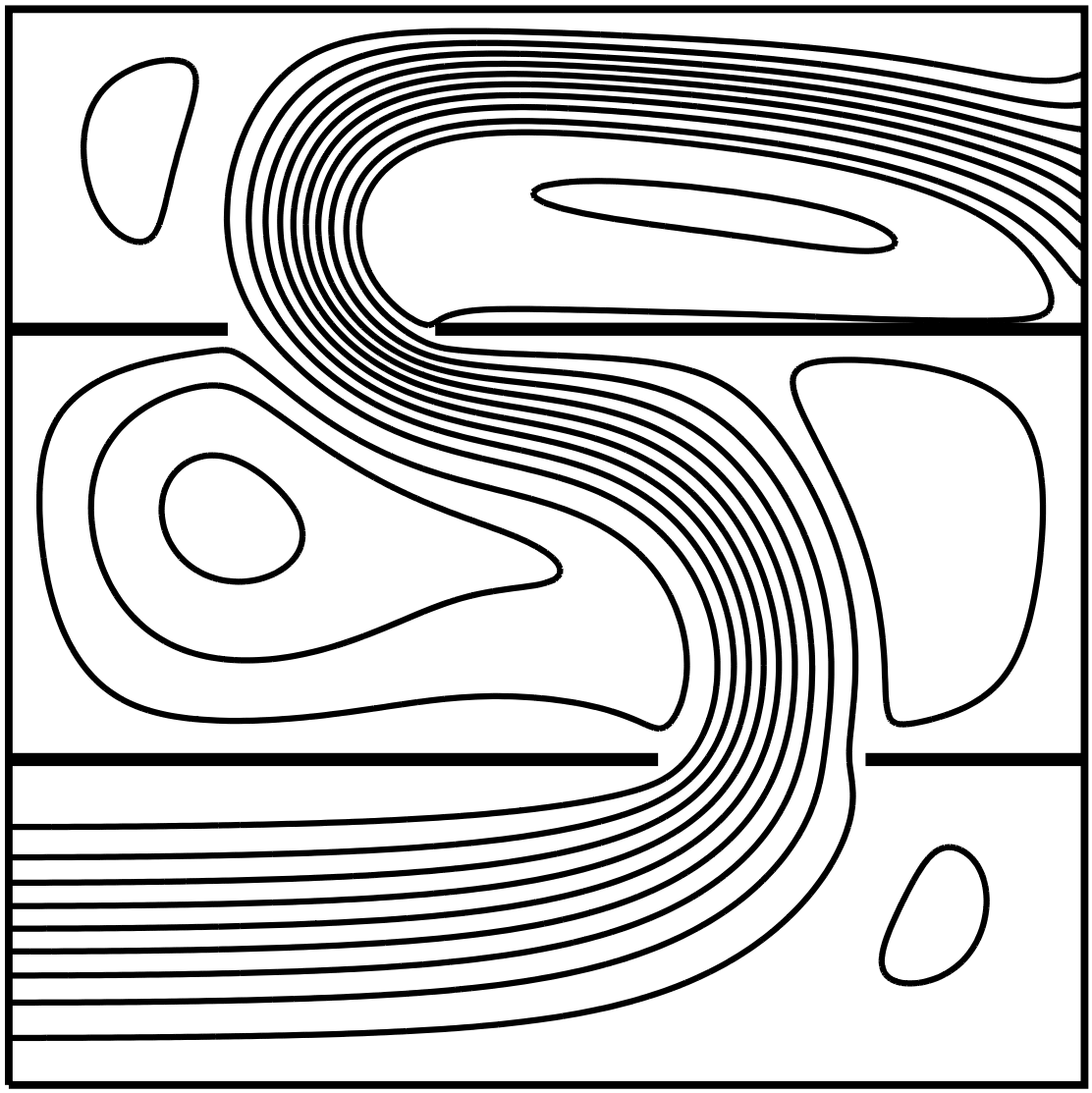

Figure 9: Streamfunction contours of 3Cham problem at Re=1000, on a 200×200 grid. Levels are -0.0100, -0.0200, -0.0300, -0.0400, -0.0500, -0.0600, -0.0700, -0.0800, -0.0900, -0.0965, -0.1100, 0.0020, 0.0100, and 0.0200.