ITERATIVE METHODS AND FORMULATIONS FOR INCOMPRESSIBLE FEM FLOW SOLVERS

R. Bramley* D. Pelletier†

Abstract

The solution of discretized incompressible flow problems involves block 2×2 linear systems for the velocity and pressure fields. This paper examines the suitability of four different formulations of these linear systems for iterative methods. The emphasis is not on the particular iterative algorithm but rather on the interrelationships between the different formulations, and their consequences for modern iterative solution methods.

Introduction

The solution of large scale incompressible problems from FEM flow solvers often requires implicitly or explicitly solving block 2×2 linear systems of the form

$$Gx = \begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} = b \qquad (1)$$

where $B \in \Re^{n \times q}$ and $A \in \Re^{n \times n}$ are large, sparse matrices resulting from a discretization of the computational domain. Here u is the discretized velocity field of the flow and p is the scalar pressure field. The size and nonzero structure of the matrices A B in this generalized Stokes problem make direct methods such as Gaussian elimination impractical. This paper presents four *formulations* of the generalized Stokes problem, discusses their suitability for iterative linear system solvers, and compares the performance of iterative solvers on the four formulations in the context of a CFD code. The comparisons center on solving the linear systems of the form (1), and address problems which are among the most challenging: the steady state solution of flows with large Reynold's numbers and unstructured, highly adapted meshes. By contrast, much of the available detailed numerical linear algebra analysis in the literature addresses linear systems which occur in time-dependent problems with quasi-uniform meshes.

Our emphasis is less on the particular choice of iterative solver used than on the interrelationships between the different formulations, and the consequences for existing iterative algorithms.

Modeling and Solver

The model problem used here is the primitive variable form of the steady-state Navier–Stokes equations

$$\begin{aligned} u \cdot \nabla u + \nabla p - \frac{1}{Re} \triangle u &= f \\ \nabla \cdot u &= 0 \end{aligned} \qquad (2)$$

where Re is the Reynold's number. Using a successive substitution or Newton's method for (2) gives a sequence of subproblems: given w with $\nabla \cdot w = 0$, solve

$$\begin{aligned} w \cdot \nabla u + \nabla p - (Re)^{-1} \triangle u &= f \\ \nabla \cdot u &= 0 \end{aligned} \qquad (3)$$

Here $A = (Re)^{-1} A_K + A_N$, where A_N approximates the convection operator $w \cdot \nabla$. A is nonsymmetric, and for large values of Re is dominated by the singular matrix A_N. These features cause difficulty for iterative linear system solvers.

The FEM solver used here adaptively refines the mesh,[13,14,18] placing more elements in regions

*CSci Dept, Indiana University, Bloomington IN 47401; email bramley@cs.indiana.edu. Work supported by NSF grant CDA 93-03189

†Senior Member AIAA, Mech.Eng. Dept, Ecole Polytechnique, Case postale 6079, succ. A, Montreal, Canada H3C 3A7

that require higher resolution as the computations proceed. The final meshes can have elements with area ratios of order 10^5 or more, a feature which also makes (1) difficult to solve iteratively.

Iterative Methods for Linear Systems

Modern iterative methods for linear systems of equations $Hx = b$ (see[2,3,20]) share important features. Generally, all have similar computational kernels, all require some form of preconditioning to be effective for practical problems, and a single property serves as the most practical sufficient condition for convergence. In particular, the biconjugate gradients (BiCG) algorithm,[10] quasi-minimal residual (QMR) methods,[12] the Orthomin(m) algorithm,[22] the restarted generalized minimal residual (GMRES(m)) algorithm,[21] the Chebyshev method,[16] and the biconjugate gradient stabilized (Bi-CGSTAB) algorithm[7] all

- have similar computational kernels: dotproducts ($\alpha = x^T y$), vector updates ($y = y + \alpha x$), matrix–vector products: $w = H * d$, $\tilde{w} = H^T * \tilde{d}$, and for robust preconditioning the solution of sparse upper and lower triangular systems $Uw = v$ and $Lu = v$.
- converge in theory if the symmetric part $\frac{1}{2}(H + H^T)$ of H is positive definite (see[8] for proofs of this for many of the cited methods).

The majority of computational time and data movement is spent in forming matrix-vector products and solving sparse triangular systems.

The key issue here is the suitability of *formulations* of (1) for iterative methods, and not the choice of iterative solver. For steady-state problems with irregular, highly adapted meshes, the particular choice of iterative solver makes surprisingly little difference. Generally, for a given problem and preconditioner they all work equally well or equally poorly. Because of this we use the Bi-CGSTAB algorithm for uniformity in the testing, but our testing has shown that the conclusions do not change when other iterative solvers are used.

Sufficient Convergence Conditions: Although having $\frac{1}{2}(H + H^T)$ positive definite assures convergence for the methods, the slightly weaker requirement that H is *positive real*, that is, all the eigenvalues of H lie in the right half of the complex plane, usually suffices for convergence. Furthermore, the usual verifiable sufficiency condition for convergence of older linear stationary methods such as Jacobi, Gauss–Seidel, and SOR is that H be diagonally dominant. With the Gerschgorin Disk Theorem,[15] diagonal dominance implies that H is positive real. Although iterative solvers may converge in other cases, in practice the only assurance is when H is positive real. For example, GMRES(m) may converge when a few eigenvalues of H have negative real parts, but it also frequently fails in that case.

Some methods not mentioned here do not require that H be positive real. For example, the conjugate gradient method can be applied to the normal equations $H^T H x = H^T b$, since $H^T H$ is symmetric and positive definite. However, the eigenvalue distribution of $H^T H$ is much worse than that of H for iterative methods, and in particular for the problems examined here the normal equations methods fail. So in practice positive real systems are more suitable for iterative methods than other systems. Furthermore, their convergence is better if the eigenvalues are *clustered*, lying in a few small regions of the complex plane, and if few of the eigenvalues are close to the origin relative to the eigenvalue largest in magnitude. These ideas are used here as criteria for the suitability of formulations of (1).

Preconditioning: *Preconditioning* means transforming a system $Hx = b$ into one of the form $M_1^{-1} H M_2^{-1} M_2 x = M_1^{-1} b$, so that the new coefficient matrix $M_1^{-1} H M_2^{-1}$ is more suitable for an iterative method. For simplicity only *left* preconditioning is discussed here, with $M_1 = M$ and $M_2 = I$, where I is the identity matrix. The goal of preconditioning is to reduce the computer solution time within allowed storage limitations. From previous discussion this often means mapping the eigenvalues of $M^{-1} H$ to the right half plane, and away from the origin.

Many preconditioning approaches have been proposed, but the most robust ones are based on *incomplete factorization* of the coefficient matrix (although some versions create and store a direct approximation of H^{-1} itself, rather than the incomplete factors of H). The simplest, ILU(0),[17] performs Gaussian elimination on H, but with any fill-in entries created in the factors discarded. This gives a factorization $H = LU + E$, where L is unit lower triangular and U is upper triangular, and the nonzeros of $L + U$ occur in the same locations as those of H. Other incomplete factorizations are based on orthogonal factorizations such as QR.[5,23] Incomplete factorizations are usually parameterized to allow some fill-in to occur, controlled by the original nonzero structure of H, the size of the created fill entries, or a combination of the two (see[20] for an excellent discussion of this and practical implementation details). Incomplete factorizations are robust because allowing all fill-in gives a complete factorization of the matrix, and the system is solved in one iteration. In practice allowing only a few "levels" of fill-in often suffices to achieve convergence. A key shortcoming of robust preconditioners is that they need to directly access the entries of the coefficient matrix H; for linear systems where H is only defined implicitly this is usually impractical.

Four Formulations of the Generalized Stokes Problems

The generalized Stokes problem commonly has four formulations for iterative solvers: mixed, augmented Lagrangian, Uzawa, and projection. The simplest case where A is symmetric and positive definite is used to illustrate the eigenvalue distributions of the coefficient matrices for these formulations, but the difficulties examined still occur when A is nonsymmetric.

Mixed Formulation: The first formulation consists of directly applying iterative methods to (1). One problem occurs because even if A is symmetric and positive definite, the mixed matrix G will be indefinite, i.e., will have positive and negative eigenvalues.[19] Furthermore, when the eigenvalues of G are distinct, q eigenvalues are negative and n are positive. Since q is the number of pressure unknowns, this means large numbers of eigenvalues are on both sides of the imaginary axis. Another problem is that the trailing block of zeros in G can cause difficulties for factorizations which assume the diagonal is nonzero. This is less of a problem than it is for direct methods, because preconditioners are approximations and replacing zero pivots encountered with an arbitrary value can allow the factorization to complete, but with a potential loss of effectiveness of the preconditioner.

Finally, note that the system is of order $n+q$, so the vector operations in the iterative methods will be of that length.

Augmented Lagrangian: The augmented Lagrangian formulation can be derived from a shifted penalty method (see[11]), giving the algorithm

Initialize: $p \;=\; p_0$ *is given*
for $k \;=\; 1,\; 2,\; \ldots$
    Solve $A_\rho \; u \;=\; f \;-\; Bp$
    Set $R \;=\; B^T u$
    Set $p \;=\; p \;+\; \rho \; R$
endfor
Solve $A_\rho \; u \;=\; f \;-\; Bp$

Here $A_\rho = A + \rho BB^T$ and ρ is the penalty parameter, typically of size $\mathcal{O}(10^5)$. In practice only a few solves with A_ρ are needed, and the final linear system solve $A_\rho u = f - Bp$ can be omitted.

The problem with the augmented Lagrangian formulation is the underlying eigenvalue distribution of A_ρ. Again suppose that A is symmetric and positive definite. The Wilkinson Interlock Theorem[24] implies that as ρ increases, eigenvalues of A_ρ separate into two groups: one bounded above by $\lambda_n(A)$, the other bounded below by $\rho\lambda_{n-q+1}(BB^T)$. When A is symmetric and positive definite the matrix A_ρ is also, and the system being solved is of order n instead of $n+q$. However, the separation of eigenvalues into two widely separated groups with a large number of distinct eigenvalues in each group causes extreme difficulties for iterative methods. Because the singular penalty term ρBB^T tends to dominate A_ρ, commonly used incomplete factorization preconditioners based on numerical tolerances for retaining fill-in elements have difficulty in producing stable preconditioners.

Uzawa Formulation: The Uzawa formulation[1] is possibly the one chosen most often. Solve (1) for $u = A^{-1}(f - Bp)$, substitute into $B^T u = g$ to get

$$B^T A^{-1} B p = B^T A^{-1} f - g \tag{4}$$

This is also called the Schur complement for the original problem. Once p is found from (4), the velocity u is found by solving $Au = f - Bp$. The Uzawa system has $q \ll n$ unknowns, and the matrix-vector products $w = B^T A^{-1} B d$ needed in iterative methods can be formed in three steps:

1. $v = B * d$
2. Solve $A\tilde{w} = v$ for $\tilde{w}$
3. $w = B^T * \tilde{w}$.

This requires *solving* a linear system with coefficient matrix A on every iteration. Also, even if A and B are sparse, $B^T A^{-1} B$ almost never is. This means the matrix $B^T A^{-1} B$ cannot be formed in practice, making incomplete factorization preconditioning impractical.

Projection Formulation: To simplify the presentation, assume $g = 0$ in (1). The projection formulation[5] is derived by solving for p in a least squares sense: $p = B^\dagger(f - Au)$, where $B^\dagger$ is the Moore–Penrose pseudoinverse (see[6] for a detailed presentation of pseudoinverses and their properties.) A pseudoinverse must be used since B is rectangular; $p = B^\dagger(f - Au)$ is the unique vector of minimum norm which solves $Au + Bp = f$ for a given u.

Substituting this p back into the same equation gives

$$\begin{aligned} Au + BB^\dagger(f - Au) &= f \\ (I - BB^\dagger)Au &= (I - BB^\dagger)f \\ PAu &= Pf, \end{aligned}$$

where $P = I - BB^\dagger$ is the orthogonal projector onto null(B^T), the space of discrete null divergence vectors. $B^T u = 0$ is equivalent to $Pu = u$, and substituting Pu for u gives the *projection system*

$$PAPu = Pf \tag{5}$$

As was shown in,[4] (5) and (1) are equivalent systems.

The projection system has a singular coefficient matrix with q eigenvalues equal to zero. However, it is consistent (Pf lies in the range of PAP), so matrix-vector based iterative methods are still applicable. Another difficulty is that the coefficient matrix cannot be explicitly formed because it is dense, so incomplete factorization preconditioners cannot be directly applied. However, since PAP is the orthogonal projection of the operator A into the space null(B^T) the preconditioner $PM^{-1}P$, where M is a preconditioner for the matrix A, is reasonable to try.

Disregarding zero eigenvalues, when A is symmetric and positive definite PAP has a better eigenvalue distribution than A and in particular a condition number no worse than A's.[4] In contrast to the Uzawa system, each iteration requires *multiplying* by the matrix A instead of *solving* a system with coefficient matrix A.

Furthermore, when the initial velocity u_0 has null-divergence, modern Krylov iterative methods maintain that property for all succeeding iterates. This allows reducing the number of projections in forming matrix-vector products from two to one per iteration, and effectively solves the restriction of the problem to the subspace of (discrete) null divergence vectors. Unlike projection methods based on the continous problem, the approach used here is purely linear algebraic and so requires no restrictions on the finite elements used, or changes in the overall CFD solution method.

Summary Comparisons of Formulations: Direct application of iterative solvers to the mixed system is possible; the system is explicitly available (or is readily made so) in most FEM codes, allowing incomplete factorization preconditioners to be used. However, the convex hull of the eigenvalues will contain the origin so there is no assurance that iterative linear solvers will converge. Furthermore, no practical preconditioner is yet known that will give a preconditioned system that has all the eigenvalues of G with negative real parts mapped to have positive real parts.

Augmented Lagrangian systems have explicit coefficient matrices and allow direct computation of robust preconditioners. Furthermore, the augmented Lagrangian formulation has been highly effective when direct solvers are used, especially when using skyline methods. However, from the eigenvalue distribution induced by the penalty term, it is clear that iterative methods are not suitable here. As will be verified in the testing in the testing section the two large groups of widely separated eigenvalues cause iterative solvers to require large numbers of iterations, and without memory intensive preconditioning they fail to converge at all. Note that this computational difficulty is not an inherent problem from the original CFD problem, but is rather an artifact introduced by the solution method in the penalty parameter ρ.

The Uzawa system generally has the best spectral characteristics of the four formulations, and in fact when a quasi-uniform mesh is used the eigenvalues are bounded independently of the mesh.[9] However, the system is only implicitly defined since the matrix $B^T A^{-1} B$ cannot be formed in practice. Furthermore, each matrix-vector product required by the iterative method involves *solving* a linear system with coefficient matrix A, an expensive operation compared to the other formulations which only require multiplying by A. Although the *order* of the system is smaller (q instead of n or $n+q$), when solving the systems with coefficient matrix A vector lengths will be n, making that advantage moot.

The projection system is also implicitly defined, but can be practically preconditioned by using a preconditioner on A. Because the orthgonal projector is $P = I - B(B^T B)^{-1} B^T$, each step involves solving a system with coefficient matrix $B^T B$. However, that subsystem is of order $q \ll n$, and can be handled by direct methods (unlike the solves with the matrix A required by Uzawa, which is of order n). The eigenvalue distribution is no worse than that of the matrix A alone, and can be significantly better.

Figures 1 – 4 show the eigenvalue distributions for a sample problem. The example is from a irregular but non-adapted mesh for a backward facing step problem in 2D, with a symmetric matrix A. In this case the eigenvalues for all four formulations are real, and are plotted against a logarithmic scale. The x-axis is the eigenvalue number when the eigenvalues are sorted in increasing order, with the negative eigenvalues of the mixed formulation negated and plotted against negative integers to distinguish them from the positive eigenvalues. The condition

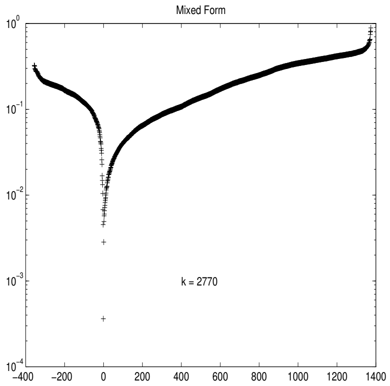

Figure 1: Spectra: Mixed Form

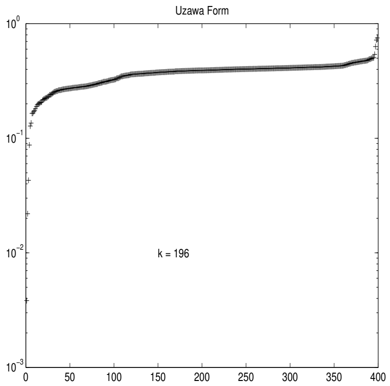

Figure 2: Spectra: Uzawa Form

numbers for these systems is the ratio of largest to smallest magnitudes of the eigenvalues and are displayed on the graphs as k. The Uzawa and projection formulations are much better conditioned than the mixed or augmented Lagrangian systems. However, for modern iterative methods the overall distribution of eigenvalues is more important than the condition number. The presence of a few isolated

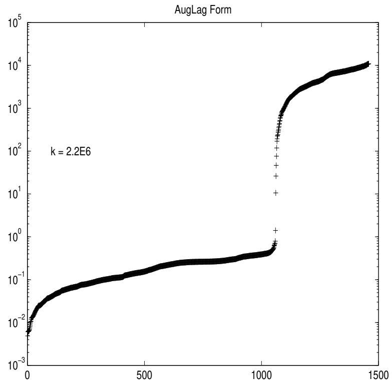

Figure 3: Spectra: AugLag Form

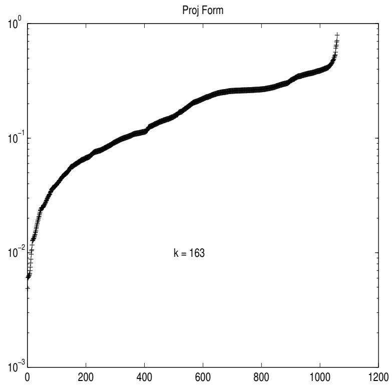

Figure 4: Spectra: Proj Form

small eigenvalues, as occurs for the Uzawa form, is not as deleterious [2, Chapter 13] as having a large group of small eigenvalues, as occurs in the augmented Lagrangian form.

Testing Results

Although the spectra suggest that the mixed or augmented Lagrangian systems will require more iterations than the projection and Uzawa systems, the last two require significantly more work per iteration by having to solve systems with coefficient matrices A and B^TB, respectively, on each iteration. A better comparison involves execution times and memory requirements for iterative solvers applied to the different formulations. This Section presents such a comparison for the backward facing step problem in 2D, with a Reynold's number of 100. Three increasingly adapted meshes are used, as shown in Figure 5 and described in.[18] Crouzeix–Raviart P_2^+–P_1 elements were used with the mid–element bubble node removed analytically. Problems STEP3, STEP4, and STEP5 had 1985, 5019, and 13797 velocity unknowns, resp. For the mixed formulation, GMRES(20) was used as the iterative method, and for the other formulations Bi-CGSTAB was used because it converged faster (but less monotonically) for a given amount of storage. Similar results hold for other nonsymmetric iterative solvers including transpose-free QMR, biconjugage gradients, generalized conjugate residual, and ORTHOMIN, but outright convergence failures were more common with those methods than with Bi-CGSTAB and GMRES(m). For preconditioning, the ILU(s) preconditioner was used where s is the number of allowed fill-in levels. When $s = n$, a complete factorization is performed, so by increasing s the preconditioner can be made more robust. In the testing $s = 0$ and $s = 3$ were used. The preconditioned projection system has the form $P(LU)^{-1}PAPu = P(LU)^{-1}Pf$, where LU are the lower and upper triangular factors produced by ILU(s).

For the Uzawa systems, the same iterative solver is used to solve systems with coefficient matrix A as is used for the overall system with coefficient matrix $B^TA^{-1}B$, giving a so-called "inner-outer" iterative method. Preconditioning is used for solving systems with coefficient matrix A, but no outer preconditioner is used. For these problems performance and convergence was worse when outer preconditioning is used with $(B^TDB)^{-1}$ where D is a diagonal matrix equal to the identity matrix, the diagonal of A, or the row sums of A. The last case corresponds to mass-lumping in many FEM formulations.

The penalty parameter used in the augmented Lagrangian system is $\rho = 10^5$. The orthogonal projections needed in the projection system were computed by forming B^TB and performing a sparse Cholesky factorization to get an upper triangular matrix R with $B^TB = R^TR$. Matrix vector products with P are computed as $w = Pv = v - B(R^TR)^{-1}B^tv$, requiring multiplication by B and B^T, solving two triangular systems, and a vector subtraction.

Figures 6–9 plot the residual norm of (1) against elapsed CPU time for the first grid, STEP3. The three curves shown on each Figure are for GMRES(20) without preconditioning, with ILU(0) preconditioning, and with ILU(3) preconditioning. Figures 10–13 and 14–17 show the same for problems STEP4 and STEP5, respectively. ILU(3) preconditioning is not shown for the mixed formulation on STEP5 because the preconditioner required more memory than was allowed for the solvers. A lack of convergence generally occurs when the system is not preconditioned, which is typical for nonsymmetric linear systems from PDE's. For Step3 only the mixed and projection forms converged without preconditioning, and for none of the other meshes did any unpreconditioned iteration succeed. Using preconditioning is not fail-safe. The augmented Lagrangian formulation in particular had failures occuring because the preconditioner was unstable. This is heuristically sensible; for large penalty parameter ρ, $A + \rho BB^T \approx \rho BB^T$, a singular matrix. In this case the incomplete factorization is especially unstable numerically, as Step5 shows: the residual grows instead of decreasing. Only when the allowed number of levels of fill-in is increased to approach that of a direct method does the preconditioner become effective. However, in that case it is better in terms of storage, CPU time, and accuracy to simply use a direct solver.

On all of the problems the Uzawa formulation performed poorly. Although the Uzawa formulation has certain optimality properties when a quasi-uniform mesh is used for some time-dependent problems,[9] it routinely fails on the adapted meshes encountered here. Part of the problem is the lack of an effective preconditioner for the overall coefficient matrix $B^TA^{-1}B$, which is inherent in the formulation.

The mixed formulation is competitive, surprisingly considering that the broadest known sufficient condition for GMRES(m) convergence is for A to be positive real, which is not the case here. The disadvantage of mixed formulation is in Table 1, which summarizes the memory requirements of the formulations. The memory shown includes all needed for data structure for the arrays, work vectors for the iterative methods, and auxillary matrices needed for the preconditioning and projection operations. For Step5 the mixed formulation failed because of excessive storage requirements. This can be mitigated somewhat by reordering the matrix for sparsity preservation (which tends to also reduce the storage requirements for incomplete factorizations), but all of the formulations can possibly benefit from reordering. Generally, the formulations in order of least memory used are Uzawa, augmented Lagrangian, projection, and then mixed.

The projection formulation was the most robust and accurate one for iterative methods. It only failed once, for Step5 with no preconditioning. Furthermore, in comparing the execution times required for a given tolerance the projection system

		Mixed	AugLag	Uzawa	Proj
Step3	NoPc	0.18	0.17	0.16	0.18
	ILU(0)	0.33	0.29	0.28	0.30
	ILU(3)	0.93	0.46	0.45	0.48
Step4	NoPc	0.45	0.42	0.41	0.47
	ILU(0)	0.87	0.74	0.73	0.79
	ILU(3)	2.70	1.35	1.34	1.40
Step5	NoPc	1.27	1.19	1.14	1.34
	ILU(0)	2.44	2.10	2.06	2.25
	ILU(3)	—	3.31	3.26	3.46

Table 1: Total Memory Requirements in Mwords for Formulations

is clearly the fastest of the four formulations. The price paid is again in increased storage, primarily for the Cholesky factorization of B^TB which corresponds to a pressure Poisson equation. However, B^TB is a sparse matrix of size much less than A. Furthermore, the matrix B^TB is independent of the outer nonlinear iteration number (unlike A) so the storage and computation of B^TB can be done once in a preprocessing phase. During the iterations only the Cholesky factor R needs to be used, and the storage needed for B^TB can be freed up.

Summary

Four formulations of the constrained linear system encountered in incompressible fluid flow computations were presented and analyzed for their suitability for iterative linear solvers. The problems targeted include the more difficult features found in FEM codes, particularly highly adapted meshes. The augmented Lagrangian formulation was found to be lest practical for iterative methods, primarily because the penalized term gives an extremely poor eigenvalue distribution. The reduced or Uzawa formulation was also found to be ineffective because of the impracticality of preconditioning the implicitly defined coefficient matrix $B^TA^{-1}B$.

The mixed formulation of straightforward application of iterative solvers to (1) was more effective than theory would predict, and only suffered in comparison because of the storage needed when using incomplete factorization preconditioning. The projection formulation was the best one in terms of time, robustness, and accuracy. Although the coefficient matrix is only implicitly defined, unlike the Uzawa formulation it can be effectively preconditioned by using a preconditioner based only on the matrix A.

These conclusions are generally independent of the particular iterative method or preconditioner used to solve the linear systems, and demonstrate intrinsic strengths and weaknesses of the *formulation* of the generalized Stokes problem.

References

[1] K. Arrow, L. Hurwicz, and H. Uzawa, *Studies in Nonlinear Programming*, Stanford University Press, Stanford, CA, 1958.

[2] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, Cambridge, UK, 1 ed., 1994.

[3] R. Barrett et al., *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, 1 ed., 1994.

[4] R. Bramley, *An orthogonal projection algorithm for linear Stokes problems*, Tech. Rep. 1190, Center for Supercomputing Research and Development, University of Illinois, Urbana, Illinois, 1992.

[5] R. Bramley, X. Wang, and D. Pelletier, *Orthogonalization based iterative methods for generalized Stokes problems*, in Solution Techniques for Large–Scale CFD Problems, W. G. Habashi, ed., Centre de Recherce en Calcul Applique, 1994.

[6] S. Campbell and J. C. Meyer, *Generalized Inverses of Linear Transformations*, Dover Publications, Inc., New York, 1979.

[7] H. V. der Vorst, *Bi–CGSTAB: A fast and smoothly converging variant of Bi–CG for the solution of nonsymmetric linear systems*, SIAM Journal of Scientific and Statistical Computing, 13 (1992), pp. 631–644.

[8] H. Elman, *Iterative methods for large, sparse, nonsymmetric systems of linear equations*, Tech. Rep. 229, Yale University, 1982.

[9] H. Elman and D. Silvester, *Fast nonsymmetric iterations and preconditioning for Navier-Stokes equations*, Tech. Rep. CS-TR-3283, Dept of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park MD, 1994.

[10] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Lecture Notes in Mathematics, No. 506, Springer-Verlag, Heidelberg, 1976, pp. 73–89.

[11] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary–Value Problems*, North Holland, New York, 1983.

[12] R. Freund and N. Nachtigal, *QMR: a quasi–minimal residual method for non–Hermitian linear systems*, Numer. Math., 60 (1991), pp. 315–339.

[13] J.-F. Hétu and D. Pelletier, *Adaptive remeshing for viscous incompressible flow*, AIAA Journal, 30 (1992), pp. 1986–1992.

[14] ——, *Fast, adaptive finite element scheme for viscous incompressible flow*, AIAA Journal, 30 (1992), pp. 2677–2681.

[15] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1 ed., 1985.

[16] T. Manteuffel, *The Tchebychev iteration for nonsymmetric linear systems*, Numer. Math., 28 (1977), pp. 307–327.

[17] J. Meijerink and H. van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148–162.

[18] D. Pelletier and J.-F. Hétu, *An adaptive finite element methodology for incompressible viscous flow*, in Advances in Finite Element Analysis in Fluid Dynamics II, ASME Winter Annual Meeting, ASME, 1992.

[19] R. Rusten and R. Winther, *A preconditioned iterative method for saddlepoint problems*, SIAM Journal of Matrix Analysis and Applications, 13 (1992), pp. 887–904.

[20] Y. Saad, *Iterative methods for sparse linear systems*, PWS publishing, New York, 1995. to appear.

[21] Y. Saad and M. Schultz, *Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[22] P. Vinsome, *Orthomin, an iterative method for solving sparse sets of simultaneous linear equations*, Proc. Fourth Symp. on Reservoir Simulation, (1976), pp. 149–159.

[23] X. Wang, K. A. Gallivan, and R. Bramley, *CIMGS: A incomplete orthogonalization preconditioner*, Tech. Rep. 393, Indiana University–Bloomington, Bloomington, IN 47405, 1994. Accepted for publication in SIAM J. Sci. Comp.

[24] J. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1 ed., 1965.

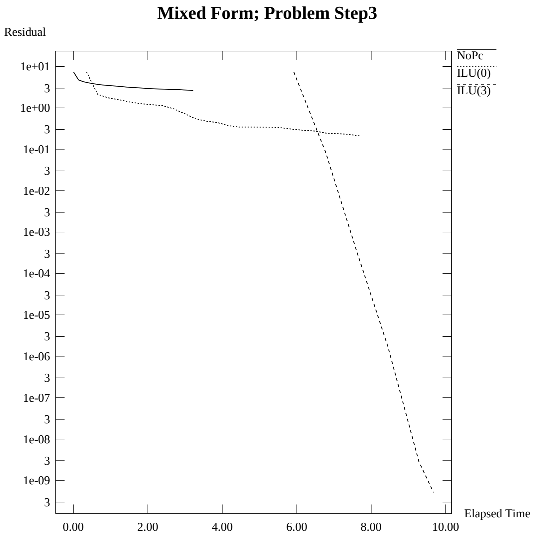

Figure 6: STEP3: Mixed Form

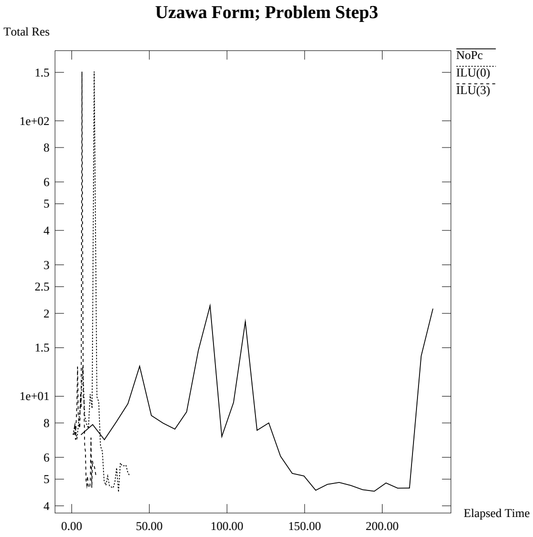

Figure 8: STEP3: Uzawa Form

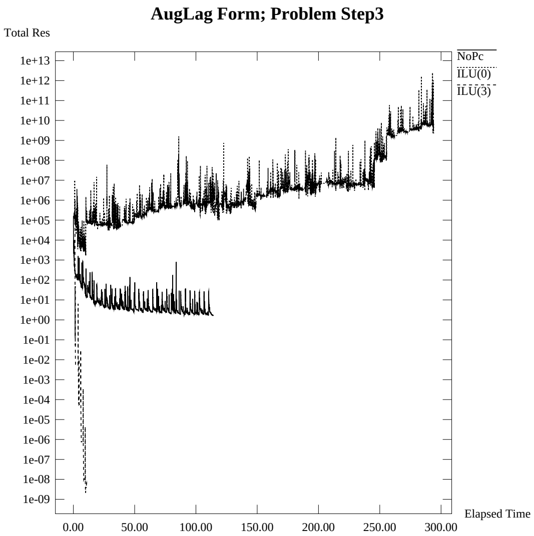

Figure 7: STEP3: AugLag Form

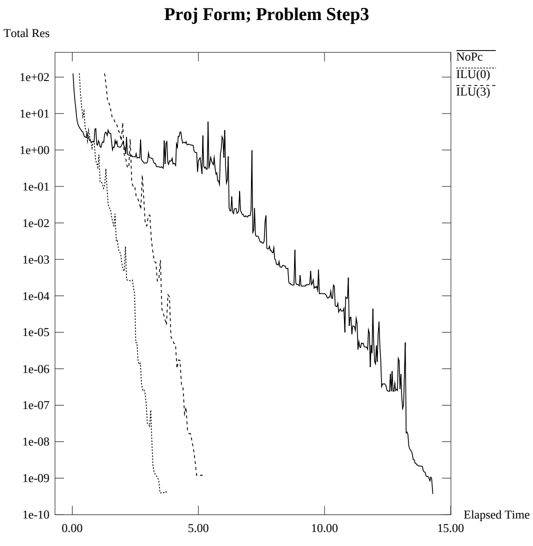

Figure 9: STEP3: Proj Form

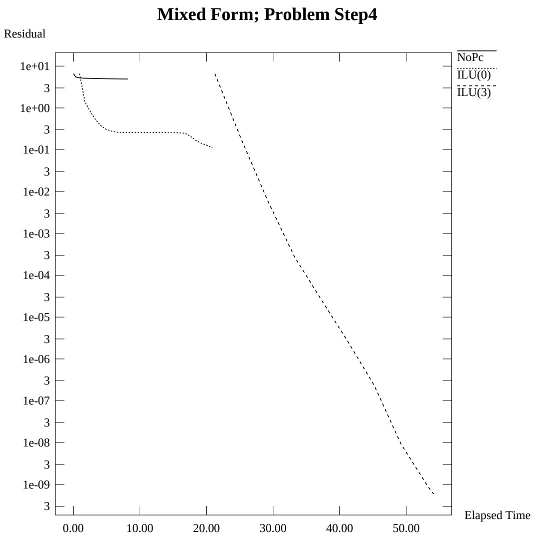

Figure 10: Step4: Mixed Form

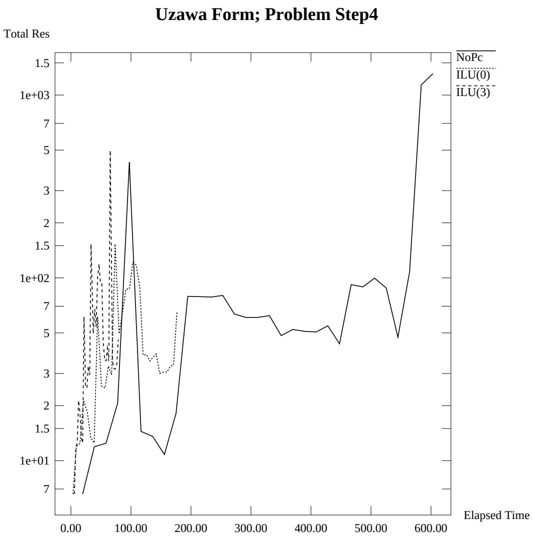

Figure 12: Step4: Uzawa Form

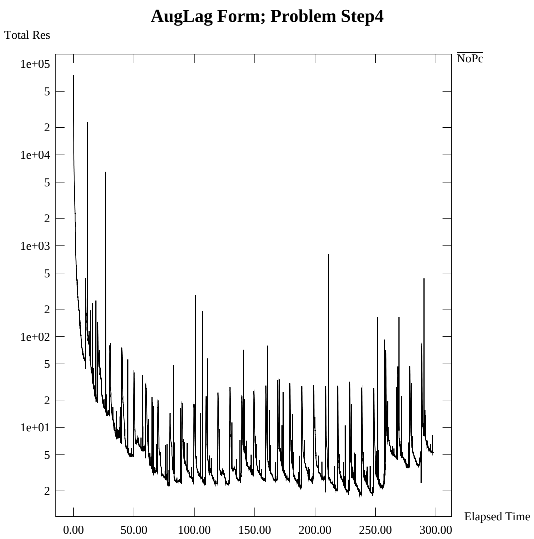

Figure 11: Step4: AugLag Form

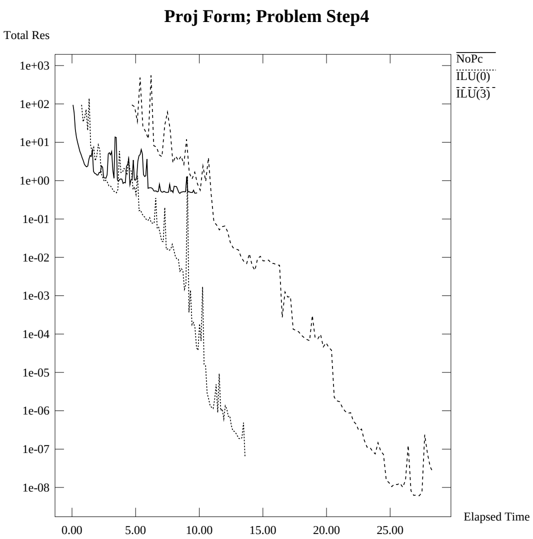

Figure 13: Step4: Proj Form

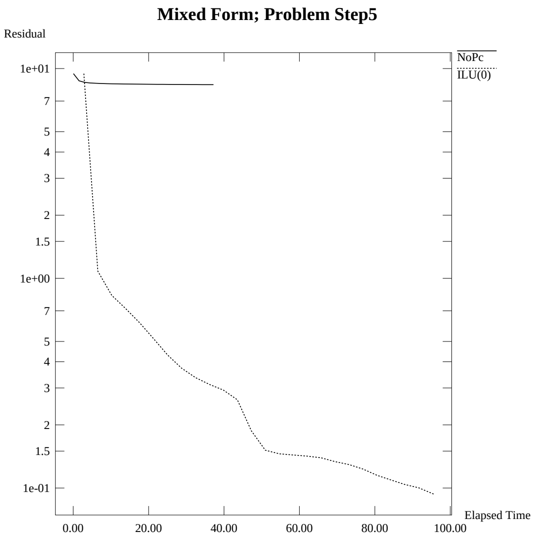

Figure 14: Step5: Mixed Form

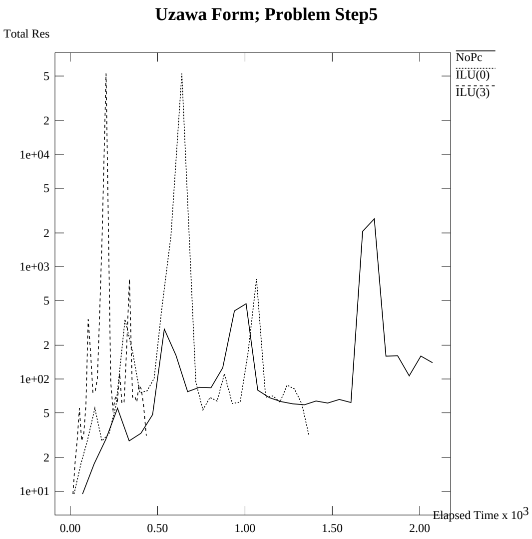

Figure 16: Step5: Uzawa Form

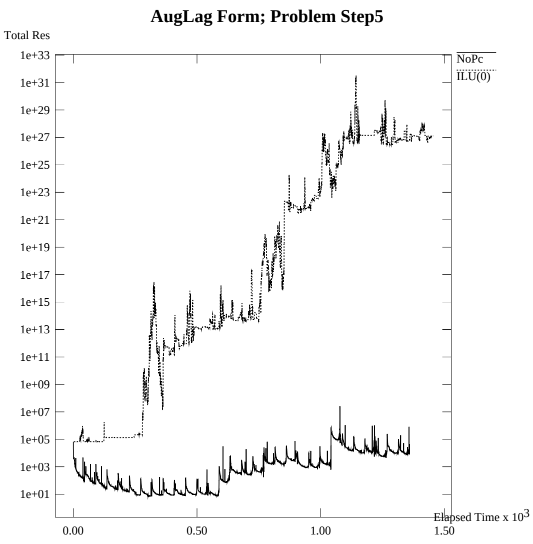

Figure 15: Step5: AugLag Form

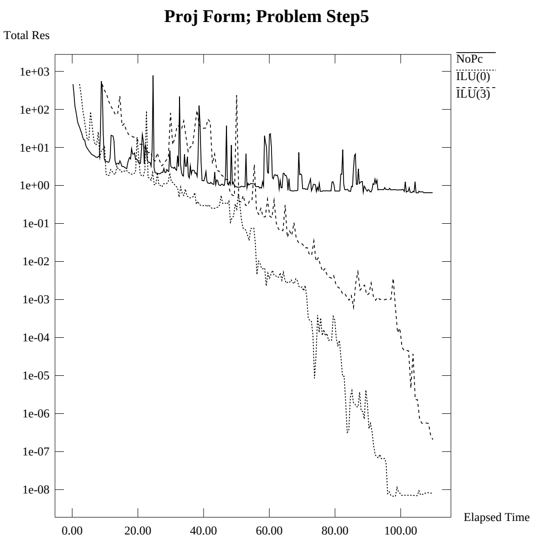

Figure 17: Step5: Proj Form

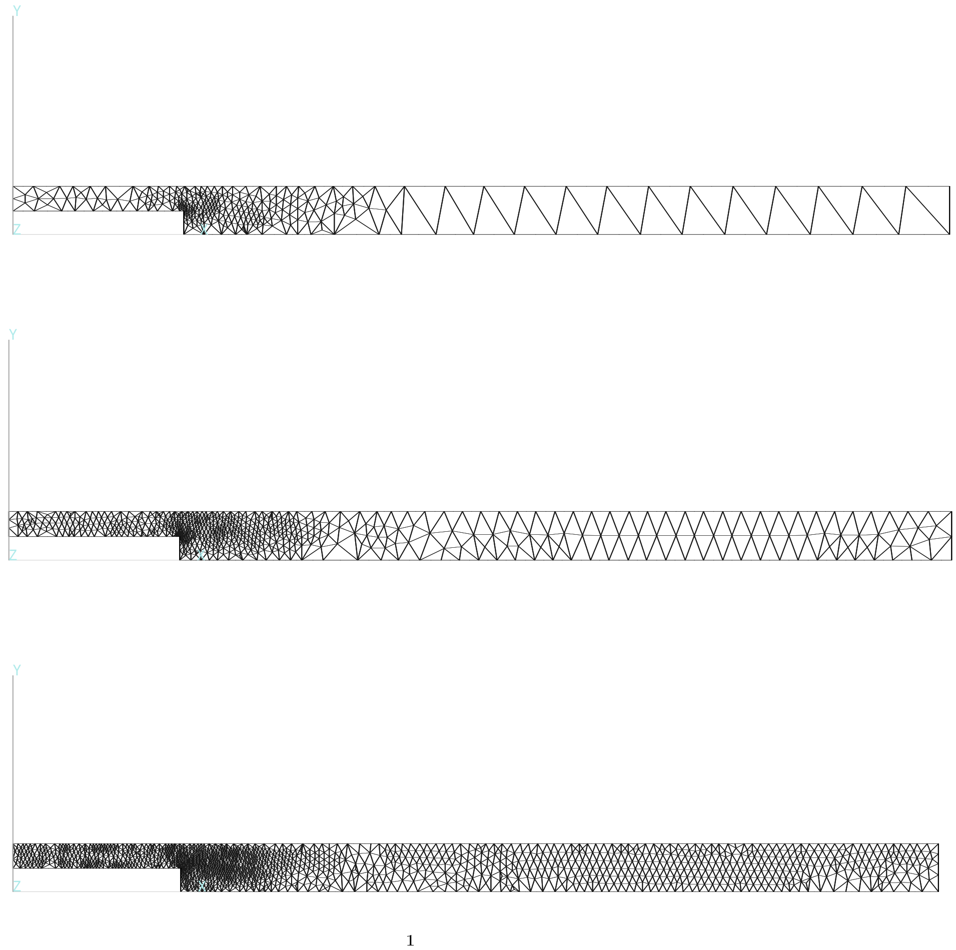

Figure 5: Meshes for Backward Facing Step Problem