Proc. of the XXVII Iberian Latin-American Congress on Computational Methods in Engineering CILAMCE 2006
Brazilian Assoc. for Comp. Mechanics & Latin American Assoc. of Comp. Methods in Engineering
Belém, Pará, Brazil, 3^{rd} – 6^{th} September 2006

Paper CIL04-557

REORDERING EFFECTS ON PRECONDITIONED KRYLOV METHODS IN AMR SOLUTIONS OF FLOW AND TRANSPORT

Lucia Catabriga
José J. Camata
Andrea M. P. Valli
luciac@inf.ufes.br
jcamata@inf.ufes.br
avalli@inf.ufes.br
Computer Science Department, Federal University of Espírito Santo
Av. Fernando Ferrari, s/n, 29060-970 ,Vitória - ES - Brazil

Alvaro L.G.A. Coutinho
alvaro@nacad.ufrj.br
Department of Civil Engineering, COPPE, Federal University of Rio de Janeiro
PO Box 68506, 21945-970, Rio de Janeiro - RJ - Brazil

Graham F. Carey
carey@cfdlab.ae.utexas.edu
Institute for Computational Engineering and Sciences, The University of Texas at Austin
201 E. 24th St., ACES 6.430, 1 University Station C0200, 78712-0227, Austin - TX - USA

Abstract. *This paper evaluates the effects of reordering the unknowns on the convergence of preconditioned Krylov subspace methods for the solution of nonsymmetric linear systems that arise from the finite element discretization of flow and transport. Of particular interest is the iterative solver behavior when adaptive mesh refinement (AMR) is utilized. Numerical studies are conducted using the object oriented AMR software system LibMesh with the PETSc Library. Using incomplete factorization preconditioners with several levels of fill-in, we investigate the effects of the Reverse Cuthill-McKee algorithm on GMRES, LCD and BICGSTAB methods. It is shown that the reordering applied in this finite element implementation with adaptive mesh refinement can reduce the number of iterations and, consequently, improve CPU time for some incomplete factorization preconditioners.*

Keywords: *Adaptive Mesh Refinement, Krylov subspaces methods, ILU preconditioners, Orderings*

1. INTRODUCTION

Incomplete factorization preconditioners are sensitive to the ordering of unknowns and equations. Reorderings have been used to reduce fill-in (as with sparse direct solvers), to introduce parallelism in the construction of an application of ILU preconditioners, and to improve the stability of the incomplete factorization. In most cases, reorderings tend to affect the rate of convergence of preconditioned Krylov subspace methods, Benzi (2002). The effects of reordering on the convergence of preconditioned Krylov subspace methods have been studied by a number of authors, mostly experimentally, and are still the subject of some debate (see, e.g., Langtangen (1989); Dimon (1989); Duff and Meurant (1989); Dutto (1993); Benzi et al. (1999); Benzi and Tuma (2000); Doi and Washio (1999); Heniche et al. (2001); Borne (2000)).

Sparse matrix reorderings have been in use for a long time. Classical ordering strategies include bandwidth- and profile-reducing ordering, such as reverse Cuthill-McKee (RCM), Sloan's ordering, Gibbs-Poole-Stockmeyer ordering, variants of minimum degree ordering, and nested dissection, Saad (1996). These orderings are based only on the structure of the matrix and not on the numerical values of the matrix entries. For direct solvers based on complete matrix factorization this is justified, particularly in the SPD (symmetric positive definitive) case. For incomplete factorizations, however, the effectiveness of reordering is strongly affected by the size of the dropped entries. Those orderings based solely on graph information may result in a poor preconditioner when applied to matrices (Benzi (2002); Duff and Meurant (1989)). For SPD finite element matrices, where a "natural" ordering of unknowns may not exist, Duff and Meurant (1989) recommend the use of RCM ordering.

The situation is somewhat different for nonsymmetric problems. In this case, matrix reorderings can significantly improve the performance of ILU-preconditioned Krylov subspace solvers. Dutto (1993) studied the effect on the convergence of GMRES with ILU(0) preconditioning in the context of solving the compressible Navier-Stokes equations on unstructured grids. RCM ordering was often better than other orderings not only in terms of performance but also in terms of robustness.

Benzi et al. (1999) showed numerical experiments about the effects of reorderings on the convergence of preconditioned Krylov subespace methods for the solution of nonsymmetric linear systems. They studied how different reorderings affect the convergence of the Krylov subspace methods when incomplete LU factorizations are used as preconditioners. Their focus was on linear systems arising from the discretization of second order partial differential equations by finite differences. They also studied a selection of nonsymmetric matrices from various sources (for example, Harwell-Boeing collection and Saad's SPARSKIT). These matrices arise from different application areas: oil reservoirs, modeling, plasma physics, neutron diffusion, and more. Some of these matrices, from finite element models, have a much more complicated structure than those arising from finite differences. For the test matrices that are nearly symmetric, the reorderings have no positive effect on the convergence of the preconditioned Krylov methods. On the other hand, for the highly nonsymmetric test matrices, i.e., when the nonsymmetric part is large, they conclude that reorderings can indeed make a big difference. It is shown that the reverse Cuthill-McKee reordering can be very beneficial in terms of the number of iterations. Benzi et al. (1999) concluded that the RCM ordering should be used as the default ordering with incomplete factorization preconditioners if some amount of fill-in is allowed.

Oliker et al. (2002) investigated the effects of various ordering and partitioning strategies on the performance of parallel conjugate gradient(CG) and ILU(0) preconditioned CG (PCG) using different programming paradigms and architectures. Their results show that for SPD matrices, ordering significantly improves overall performance on both distributed and distributed shared-memory systems, but they concluded that the quality of an ILU preconditioner in terms of the convergence rate has a nontrivial dependence on the ordering.

Hou (2005) presented a study of an incomplete ILU preconditioner for solving linear systems arising from the finite element method using reordering to improve the efficiency of serial and parallel implementations. He compared the classic Cuthill-McKee ordering and the popular Reverse Cuthill-McKee, both applied in ILU(0) and ILUT preconditioners. He concluded that nodal reordering strategies go side-by-side with preconditioners. While one strategy works very poorly for one preconditioner, it can be the best choice for another. No ordering is perfect in all situations, and therefore we must be careful when choosing one for each individual problem.

In this work we evaluate the effects of reordering the unknowns on the convergence of preconditioned Krylov subspace methods for the solution of nonsymmetric linear systems that arise from the finite element discretization of flow and transport. Of particular interest is the iterative solver behavior when adaptive mesh refinement (AMR) is utilized. Numerical studies are conducted using the object oriented AMR software system LibMesh with the PETSc Library (LibMesh (2006); PETSc (2006)). Using incomplete factorization preconditioners, we investigate the GMRES, LCD (Catabriga et al. (2004)) and BICGSTAB methods for solving the Navier-Stokes equations and compare standard nodal ordering with reverse Cuthill-McKee ordering.

2. GOVERNING EQUATIONS AND DISCRETE FORMULATION

The system of equations considered is the unsteady Navier-Stokes equations for low-speed incompressible fluid flow, in the velocity-pressure formulation,

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} - \frac{1}{Re}\nabla^2\mathbf{u} + \boldsymbol{\nabla} p = \mathbf{f} \qquad \text{in } \Omega\times(0,T) \tag{1}$$

$$\boldsymbol{\nabla}\cdot\mathbf{u} = 0 \qquad \text{in } \Omega\times(0,T) \tag{2}$$

where Ω is the flow domain, $\mathbf{u}$ is the velocity vector, p is the pressure, Re is the Reynolds number and $\mathbf{f}$ is an applied body force. In addition, we require boundary data on $\partial\Omega \times [0,T]$, $\mathbf{u} = \mathbf{u}_0$, and initial data at $t = 0$, $\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x})$ in Ω, to complete the specification of the evolution problem. To satisfy the LBB criterion, different approximation spaces are used for the velocity and pressure fields. Here we use a mixed finite element formulation as developed in Carey and Oden (1986).

Let us consider the spatial discretization of the viscous flow equation. Introducing a finite element discretization and basis for the velocity components and for the pressure on a discretization Ω_h, the semidiscrete projection of the variational formulation of the Navier-Stokes equations (1), (2) reduces to: Find the pair $(\mathbf{u}_h, p_h)$ with $\mathbf{u}_h \in V^h$ satisfying the initial condition with $\mathbf{u}_h$ = $\mathbf{u}_0$ on $\partial\Omega_h$ and $p_h \in P^h$, such that

$$\int_{\Omega_h} ((\frac{\partial \mathbf{u}_h}{\partial t} + (\mathbf{u}_h\cdot\nabla)\mathbf{u}_h)\cdot\mathbf{v}_h + \frac{1}{Re}\nabla\mathbf{u}_h{:}\nabla\mathbf{v}_h - \mathbf{p}_h\nabla\cdot\mathbf{v}_h)\, d\Omega = \int_{\Omega_h} \mathbf{f}\cdot\mathbf{v}_h\, d\Omega \tag{3}$$

$$\int_{\Omega_h} (\nabla\cdot\mathbf{u}_h)\, q_h\, d\Omega = 0 \tag{4}$$

for all admissible $\mathbf{v}_h \in V_h$, with $\mathbf{v} = \mathbf{0}$ on $\partial\Omega$, and $p_h \in P_h$. Here $\nabla \mathbf{u}_h : \nabla \mathbf{v}_h$ is the dyadic product. Introducing expansions for $\mathbf{u}_h$ and p_h and finite element test bases for $\mathbf{v}_h$ and q_h into the variational statements (3) and (4) and integrating, we obtain the following nonlinear semidiscrete system of ordinary differential equations

$$\mathcal{M}\frac{d\mathbf{U}}{dt} + \mathcal{D}(\mathbf{U}) + \frac{1}{Re}\mathcal{A}\mathbf{U} - \mathcal{B}\mathbf{P} = \mathcal{F} \quad (5)$$

$$\mathcal{B}^T\mathbf{U} = \mathbf{0} \quad (6)$$

where $\mathbf{U}^T = [\mathbf{u}_1^T \mathbf{u}_2^T]$, $\mathbf{U}_i^T = [U_i^1 \cdots U_i^N]$, $i = 1, 2$, for N nodal velocities and $\mathbf{P}^T = [p_1 \cdots p_M]$ for M nodal pressure. The matrices $\mathcal{M}$, $\mathcal{A}$ and $\mathcal{B}$ correspond to the respective mass, viscous and pressure terms on the left in (5), $\mathcal{F}$ corresponds to the source term on the right and $\mathcal{D}(\mathbf{U})$ is a nonlinear function of the nodal velocities correspond to the inertial term. The resulting semidiscrete system (5) is integrated implicitly using a standard θ-method, $0 \leq \theta \leq 1$. At each timestep, we have a nonlinear system of the form

$$\begin{aligned} \mathcal{M}\frac{\mathbf{U}^n - \mathbf{U}^{n-1}}{\Delta t} &+ \theta\,[\mathcal{D}(\mathbf{U}^n) + \frac{1}{Re}\mathcal{A}\mathbf{U}^n - \mathcal{B}\mathbf{P}^n] \\ &+ (1-\theta)\,[\mathcal{D}(\mathbf{U}^{n-1}) + \frac{1}{Re}\mathcal{A}\mathbf{U}^{n-1} - \mathcal{B}\mathbf{P}^{n-1}] = \mathcal{F}^n \end{aligned} \quad (7)$$

$$\mathcal{B}^T\mathbf{U}^n = \mathbf{0} \quad (8)$$

where n denotes the timestep index. In the numerical studies, we are using the implicit Euler method ($\theta = 1$) even though it is only first-order accurate in time. The reason for this decision is that the second-order Crank-Nicolson method is notoriously oscillatory for problems with discontinuous initial data such as the lid-driven cavity. We therefore, sacrifice accuracy in time for stability. Since however the solution reaches steady-state relatively quickly, we can afford to take small timesteps.

The nonlinear system (7), (8) is solved by Newton's method in the present study. Wrinting the nonlinear system formally as

$$\mathbf{g}(\mathbf{r}^n) = \mathbf{0}, \quad \text{with} \quad (\mathbf{r}^n)^T = [(\mathbf{U}^n)^T \;\; (\mathbf{P}^n)^T] \quad (9)$$

and given $\mathbf{r}_0^n$, we solve the linear Jacobian system

$$\mathbf{J}(\mathbf{r}_{k+1}^n - \mathbf{r}_k^n) = -\mathbf{g}(\mathbf{r}_k^n), \quad \text{where} \quad \mathbf{J} = (\mathbf{J}_{ij}) = (\frac{\partial g_i}{\partial r_j}), \;\; i, j = 1, 2, \cdots, N \quad (10)$$

for iterate $k = 0, 1, 2, \ldots$, at each timestep. The resulting linear system of equations are solved using preconditioned GMRES, LCD, and BICGSTAB methods.

The error indictor for the Adaptive Mesh Refinement (AMR) procedure used is explained in Kelley et al. (1983). The error is computed for each active element using the provided flux-jump indicator. Comparing the element error in the active elements with a global error estimate we can select regions where the mesh is refined or coarsened.

3. NUMERICAL EXPERIMENTS

The lid-driven cavity flow is a standard test case for steady Navier-Stokes computations and there are numerous published results that can be used for comparison purposes

De Sampaio et al. (1993); Spotz and Carey (1995); Ghia et al. (1982). The domain of analysis is a unit square. Both velocity components are prescribed to be zero, except at the top boundary (the lid) where the horizontal velocity component has a unit value.

For all tests the linear solver tolerance is $\epsilon = 10^{-6}$ and the non-linear tolerance is $\tau = 10^{-4}$. The AMR parameters are the initial mesh (*initial-mesh*), the number of uniform refinement steps (*uniform-refinement*), the error percentage to refine (*refine-percentage*), the error percentage to coarse (*coarsen-percentage*) and the number of maximum refinements levels (*refinement-levels*). An AMR procedure is identified by the 5-tuple: AMR(*initial-mesh*, *uniform-refinement*, *refine-percentage*, *coarsen-percentage*, *refinement-levels*).

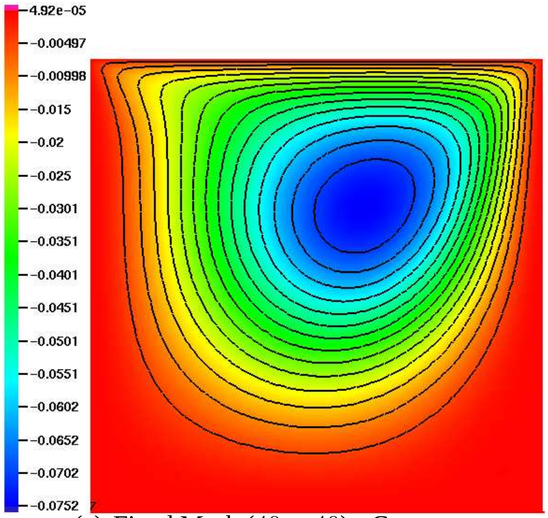
We use the Reverse Cuthill-McKee (RCM) technique (available in the PETSC Library) to reorder the matrix of the resulting system in each nonlinear step. We compare the behavior of preconditioned GMRES, LCD, and BICGSTAB methods when the RCM is considered in the AMR solution of the lid-driven cavity problem.

First, the Reynolds number $Re = 200$ is considered. To compare the linear solver behavior for different orderings, we consider a time step $\Delta t = 1$ for both procedures (AMR and fixed mesh), a quadratic triangular mesh for velocity and linear triangular mesh for pressure and a final time of 20. We consider a 40×40 fixed mesh and an adaptive mesh refinement defined by AMR(15×15,0,0.3,0.0,2). Figure 1 shows the streamlines. The AMR and fixed mesh have a similar solution. Figure 2 shows the horizontal velocity u along the vertical centerline and the vertical velocity v along the horizontal centerline. The results are comparable with Ghia et al. (1982), Spotz and Carey (1995).

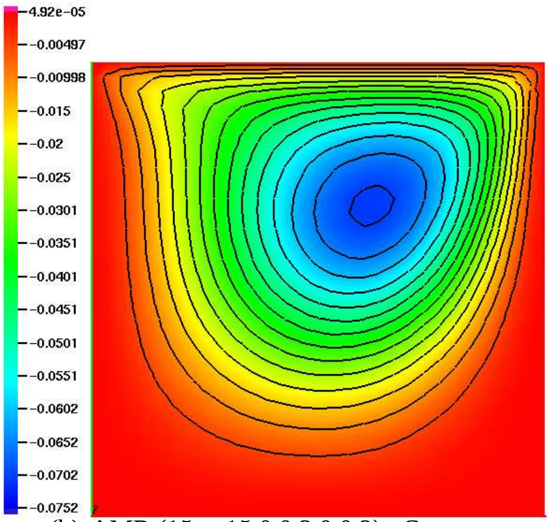
Table 1 and 2 show, respectively, for the natural and RCM orderings, the number of linear iterations (LI), the number of nonlinear iterations (NLI), the CPU time when the AMR and fixed mesh procedures are considered and the rate between the CPU time for AMR and fixed mesh procedures (CPU(AMR)/CPU(Fixed)). The results for GMRES(30), LCD(30) and BICGSTAB methods are considered for ILU(0), ILU(1) and ILU(4) preconditioners. For all linear solvers the total CPU time in the AMR procedure is smaller than the fixed mesh solution. The GMRES(30), LCD(30) and BICGSTAB methods do not converge using AMR or fixed meshes for the ILU(0) preconditioner with RCM reordering. When the natural ordering is considered, the CPU rates between AMR and fixed mesh solutions are around 22% for ILU(1) and ILU(4) preconditioners and around 34% for ILU(0) preconditioner. However, when the RCM ordering is considered, the CPU rates are 27% for ILU(1) and 38% for ILU(4).

The BICGSTAB method with AMR and ILU(0) preconditioner achieved the smaller CPU time when natural ordering is considered and the GMRES(30) method with AMR and ILU(0) the largest (Tab. 1). On the other hand, when RCM ordering is considered the solution with GMRES(30), AMR and ILU(0) preconditioner has the smallest CPU time and the BICGSTAB method with AMR and ILU(4) has the largest (Tab. 2). In general the BICGSTAB method has smaller number of linear iterations than the GMRES(30) and LCD(30) methods, but the CPU time can be larger. We can say that BICGSTAB iterations are more expensive than LCD iterations in terms of CPU time, and LCD iterations can be more expensive than GMRES iterations, Catabriga et al. (2004).

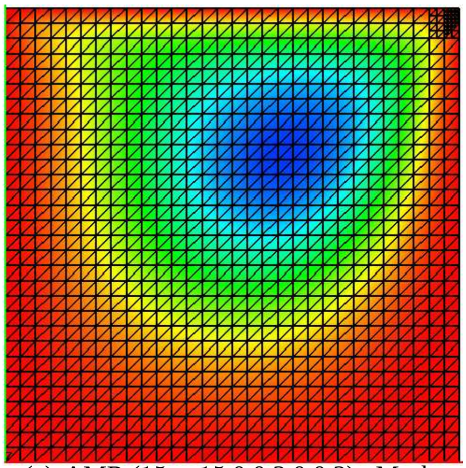
For $Re = 400$ is considered a time step $\Delta t = 1$ for both procedures (AMR and fixed mesh), a quadratic triangular mesh for velocity and linear triangular mesh for pressure,



(a) Fixed Mesh (40×40) - Contours

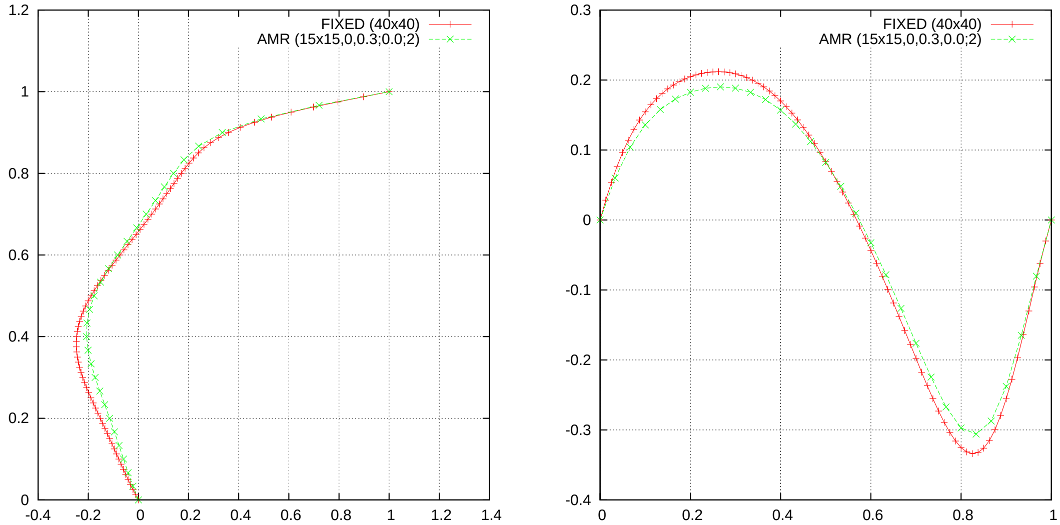


(b) AMR (15×15,0,0.3,0.0,2) - Contours



(c) AMR (15×15,0,0.3,0.0,2) - Mesh

Figure 1: Lid-driven cavity $Re = 200$ - Structured Triangular mesh - GMRES(30) Streamlines



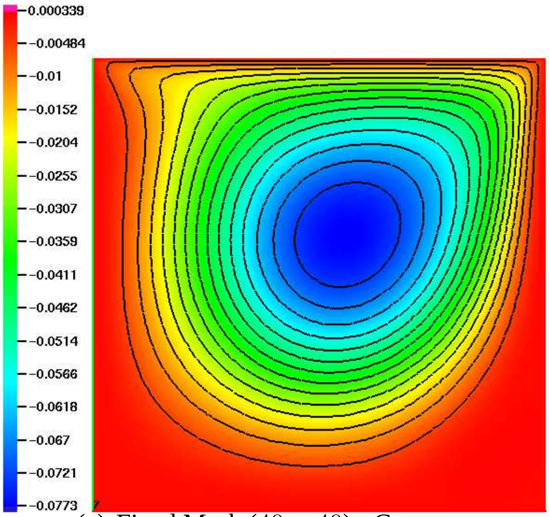
(a) Horizontal velocity component along the vertical centerline (b) Vertical velocity component along the horizontal centerline

Figure 2: Lid-driven cavity $Re = 200$ - Structured Triangular mesh - GMRES(30) velocities components along the vertical and horizontal centerlines

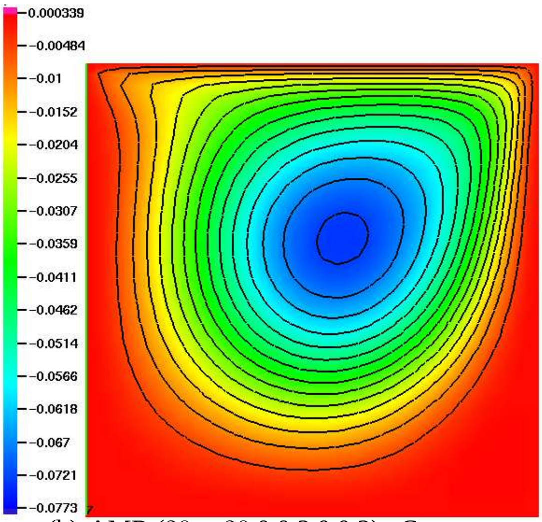
afinal time of 40, a 40×40 fixed mesh and AMR(20×20,0,0.3,0.0,2). Figure 3 and 4 show, respectively, the streamlines and the velocities along the centerlines. Again, the results are comparable with Ghia et al. (1982), Spotz and Carey (1995).

Table 3 and 4 show, respectively, the computational cost for $Re = 400$ when natural and RCM orderings are considered. For all linear solvers the total CPU time in the AMR procedure is shorter than with the fixed mesh. One exception is the GMRES solution with ILU(0) preconditioner. In this case, the AMR CPU time is larger than fixed mesh CPU time. The CPU rates for $Re = 400$ are larger than the CPU rates for $Re = 200$. One more time, the BICGSTAB method with AMR and ILU(0) preconditioner achieved the smallest CPU time when natural ordering is considered and the GMRES(30) solution with AMR and ILU(0) the largest (Tab. 3). On the other hand, when RCM ordering is considered, the GMRES(30) solution with AMR and ILU(1) preconditioner achieved the smallest CPU time and the BICGSTAB method with AMR and ILU(4) the largest (Tab. 4).

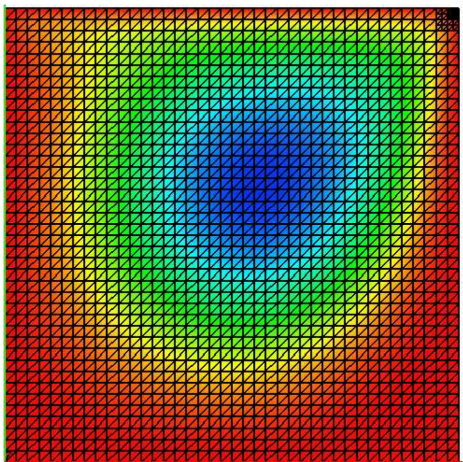
Figure 5 shows the sparsity patterns of the GMRES(30) Jacobian matrix for $Re = 400$ case, with AMR and fixed mesh. In Figs. 5(a) and (c) no bandwidth reduction was considered, while in Figs. 5(b) and (d) we can see the effects of RCM ordering in the bandwidth reduction of corresponding the original matrix. Table 5 shows the number of nonzeros of the resulting matrix ($nz(A)$) and the rate between the number of nonzeros of the preconditioner matrix ($nz(M)$) and the resulting matrix ($nz(A)$) for natural and RCM orderings, for GMRES(30) with ILU(0), ILU(1) and ILU(4) preconditioners. As the ILU(0) preconditioner does not take into account any level of fill-in, the rate $nz(M)/nz(A)$ is equal to 1.0 for AMR and fixed mesh procedures. But, as can be observed in Tab. 5, for the ILU(1) preconditioner the rate is around 30% for the natural ordering and around 75% for the RCM ordering. For the ILU(4) preconditioner, the rate



(a) Fixed Mesh (40×40) - Contours

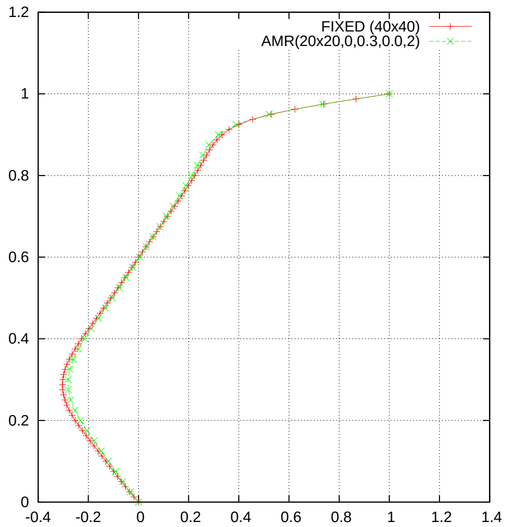


(b) AMR (20×20,0,0.3,0.0,2) - Contours

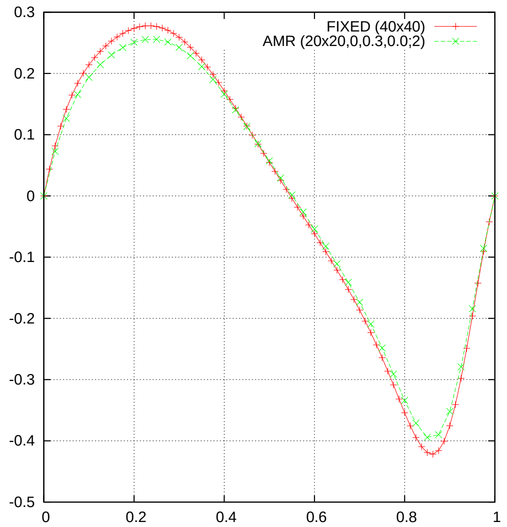


(c) AMR (20×20,0,0.3,0.0,2) - Mesh

Figure 3: Lid-driven cavity $Re = 400$ - Structured Triangular mesh - GMRES(30) Streamlines

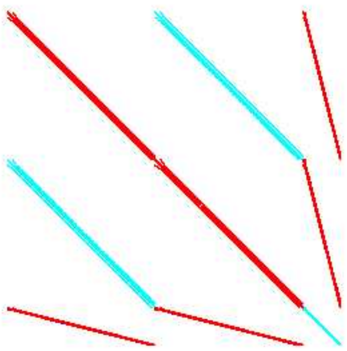


(a) Horizontal velocity component along the vertical centerline



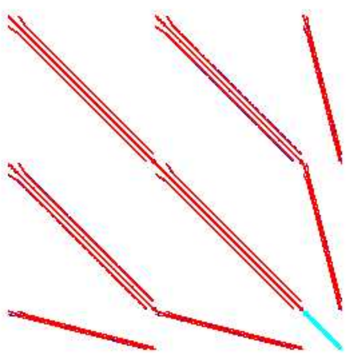
(b) Vertical velocity component along the horizontal centerline

Figure 4: Lid-driven cavity $Re = 400$ - Structured Triangular mesh - GMRES(30) velocities components along the vertical and horizontal centerlines



(a) Fixed mesh - Natural ordering

(b) Fixed mesh - RCM ordering



(c) AMR (step 40) - Natural ordering

(d) AMR (step 40) - RCM ordering

Figure 5: Lid-driven cavity $Re = 400$ - Structured Triangular mesh - GMRES(30) with ILU(1) - Pattern of the Jacobian matrix

Table 1: Lid-driven cavity $Re = 200$ - Structured Triangular mesh - Natural Ordering - Computational costs

ILU(0)							
Linear Solver	AMR(15×15,0,0.3,0.0,2)			Fixed Mesh (40×40)			$\frac{\text{CPU(AMR)}}{\text{CPU(Fixed)}}$
	LI	NLI	CPU	LI	NLI	CPU	
GMRES(30)	7764	83	39.1387	3762	60	101.1261	0.39
LCD(30)	7180	83	38.9059	4545	61	115.0701	0.34
BICGSTAB	2986	83	34.1845	2694	61	113.5000	0.30
ILU(1)							
Linear Solver	AMR(15×15,0,0.3,0.0,2)			Fixed Mesh (40×40)			$\frac{\text{CPU(AMR)}}{\text{CPU(Fixed)}}$
	LI	NLI	CPU	LI	NLI	CPU	
GMRES(30)	1565	83	36.6392	1287	61	171.9519	0.21
LCD(30)	1575	83	36.7778	1378	61	175.8095	0.21
BICGSTAB	1049	83	38.0464	872	61	185.9538	0.20
ILU(4)							
Linear Solver	AMR(20×20,0,0.3,0.0,2)			Fixed Mesh (40×40)			$\frac{\text{CPU(AMR)}}{\text{CPU(Fixed)}}$
	LI	NLI	CPU	LI	NLI	CPU	
GMRES(30)	878	83	129.9476	487	61	594.2065	0.22
LCD(30)	878	83	129.3947	495	61	597.3094	0.22
BICGSTAB	587	83	132.7534	495	61	594.9729	0.22

is around 7% for the natural ordering and around 35% for the RCM ordering.

4. CONCLUSIONS

This paper evaluated the effects of reordering the unknowns on the convergence of preconditioned Krylov subspace methods for the solution of nonsymmetric linear systems that arise from the finite element discretization of flow and transport. Of particular interest was the iterative solver behavior when AMR is utilized. Using incomplete factorization preconditioners, we investigated GMRES, LCD and BICGSTAB methods for solving the Lid-driven cavity application and then compared natural ordering with reverse Cuthill-McKee ordering. The ILU(0) preconditioner is less sensitive to RCM reordering for fixed meshes and AMR. However, for ILU(1) and ILU(4) preconditioners, RCM reordering applied with adaptive mesh refinement reduces considerably CPU time. The ILU(1) preconditioner offers the best results for the GMRES, LCD, and BICGSTAB methods in terms of CPU time.

Acknowledgements

Dr. Catabriga was supported by a CNPq grant from the Ministry of Science and Tecnology, Brazil, during her research visit at the University of Texas at Austin. This work was developed under the Joint Cooperation Agreement between the University of Texas

Table 2: Lid-driven cavity $Re = 200$ - Structured Triangular mesh - RCM Ordering - Computational costs

ILU(1)							
Linear Solver	AMR(15×15,0,0.3,0.0,2)			Fixed Mesh (40×40)			$\frac{\text{CPU(AMR)}}{\text{CPU(Fixed)}}$
	LI	NLI	CPU	LI	NLI	CPU	
GMRES(30)	1207	83	28.2950	921	60	104.2174	0.27
LCD(30)	1193	83	28.3976	952	61	106.3417	0.27
BICGSTAB	793	83	28.8487	837	62	120.8106	0.27
ILU(4)							
Linear Solver	AMR(15×15,0,0.3,0.0,2)			Fixed Mesh (40×40)			$\frac{\text{CPU(AMR)}}{\text{CPU(Fixed)}}$
	LI	NLI	CPU	LI	NLI	CPU	
GMRES(30)	721	83	31.6383	289	61	83.3840	0.38
LCD(30)	727	83	31.7965	290	61	83.6173	0.38
BICGSTAB	450	83	32.0581	172	61	86.4968	0.38

at Austin and COPPE, the Graduate School of Engineering of the Federal University of Rio de Janeiro, Brazil. This work has been supported in part by DoD High Performance Computing Modernization Program and ICES/UT, US.

REFERENCES

Benzi, M., 2002. Preconditioning techniques for large linear system: A survey. *Journal of Computational Physics*, vol. 182, pp. 418–477.

Benzi, M., Szyld, D. B., & van Duin, A., 1999. Ordering for incomplete factorization preconditioning of nonsymmetric problems. *SIAM J. Sci. Comput*, vol. 20, n. 5, pp. 1652–1670.

Benzi, M. & Tuma, M., 2000. Orderings for factorized sparse approximate inverse preconditioners. *SIAM Journal on Scientific Computing*, vol. 21, n. 5, pp. 1851–1868.

Borne, S. L., 2000. Ordering techniques for two- and three-dimensional convection-dominated elliptic boundary value problems. *Computing*, vol. 64, n. 2, pp. 123–155.

Carey, G. & Oden, J., 1986. *Finite Elements: Fluid Mechanics*, volume 6. Prentice–Hall, Inc., Englewood Cliffs, NJ.

Catabriga, L., Coutinho, A., & Franca, L., 2004. Evaluating the lcd algorithm for solving linear systems of equations arising from implicit SUPG formulation of compressible flows. *International Journal for Numerical Methods in Engineering*, vol. 60, pp. 1513–1534.

De Sampaio, P. A. B., Lyra, P. R. M., Morgan, K., & Weatherill, N. P., 1993. Petrov-Galerkin solutions of the incompressible Navier-Stokes equations in primitive variables

Table 3: Lid-driven cavity $Re = 400$ - Structured Triangular mesh - Natural Ordering - Computational costs

ILU(0)							
Linear Solver	AMR(20×20,0,0.3,0.0,3)			Fixed Mesh (40×40)			$\frac{\text{CPU(AMR)}}{\text{CPU(Fixed)}}$
	LI	NLI	CPU	LI	NLI	CPU	
GMRES(30)	31228	154	186.2470	5531	112	171.3527	1.09
LCD(30)	18313	154	145.3640	6666	113	190.1501	0.76
BICGSTAB	6935	154	121.4028	3856	110	186.0044	0.65
ILU(1)							
Linear Solver	AMR(20×20,0,0.3,0.0,3)			Fixed Mesh (40×40)			$\frac{\text{CPU(AMR)}}{\text{CPU(Fixed)}}$
	LI	NLI	CPU	LI	NLI	CPU	
GMRES(30)	4432	154	135.3617	1909	111	200.9618	0.67
LCD(30)	4127	154	131.7928	2070	111	207.6273	0.63
BICGSTAB	2489	154	133.9542	1197	111	207.0563	0.65
ILU(4)							
Linear Solver	AMR(20×20,0,0.3,0.0,3)			Fixed Mesh (40×40)			$\frac{\text{CPU(AMR)}}{\text{CPU(Fixed)}}$
	LI	NLI	CPU	LI	NLI	CPU	
GMRES(30)	2054	154	480.6099	895	110	1067.1002	0.45
LCD(30)	2055	154	473.6868	914	111	1079.4909	0.43
BICGSTAB	1375	154	482.8285	579	112	1103.7567	0.44

with adaptive remeshing. *Computer Methods in Applied Mechanics and Engineering*, vol. 106, pp. 143–178.

Dimon, D., 1989. Incomplete LU preconditioners for conjugate-gradient-type iterative methods. In *SPE Reservoir Simulation Symposium*, Dallas, Texas, Society of Petroleum Engineers.

Doi, S. & Washio, T., 1999. Ordering strategies and related techniques to overcome the trade-off between parallelism and convergence in incomplete factorizations. *Parallel Computing*, vol. 25, n. 13-14, pp. 1995–2014.

Duff, I. S. & Meurant, G., 1989. The effect of ordering on preconditioned conjugate gradient. *BIT*, vol. 29, pp. 635–657.

Dutto, L. C., 1993. The effect of reordering on preconditioned GMRES algorithm for solving the compressible navier-stokes equations. *Internat. J. Numer. Methods Engrg.*, vol. 36, pp. 457–497.

Ghia, U., Ghia, K. N., & Shin, C. T., 1982. High re solutions for incompressible flow using Navier-Stokes equations and a multi-grid method. *Journal Comput. Phys.*, vol. , pp. 387–411.

Table 4: Lid-driven cavity $Re = 400$ - Structured Triangular mesh - RCM Ordering - Computational costs

ILU(1)							
Linear Solver	AMR(20×20,0,0.3,0.0,2)			Fixed Mesh (40×40)			$\frac{\text{CPU(AMR)}}{\text{CPU(Fixed)}}$
	LI	NLI	CPU	LI	NLI	CPU	
GMRES(30)	2572	154	96.6348	1753	112	133.9053	0.72
LCD(30)	2579	154	96.9079	1797	114	137.7249	0.70
BICGSTAB	1850	154	99.5153	1521	112	149.3103	0.67
ILU(4)							
Linear Solver	AMR(20×20,0,0.3,0.0,2)			Fixed Mesh (40×40)			$\frac{\text{CPU(AMR)}}{\text{CPU(Fixed)}}$
	LI	NLI	CPU	LI	NLI	CPU	
GMRES(30)	1349	154	108.2803	463	111	149.6599	0.72
LCD(30)	1351	154	108.8921	464	111	149.7065	0.73
BICGSTAB	832	154	110.1914	284	111	153.9423	0.72

Heniche, M., Secretan, Y., & Leclerc, M., 2001. Efficient ILU preconditioning and inexact-newton-gmres to solve the 2d steady shallow water equations. *Communications in Numerical Methods in Engineering*, vol. 17, n. 2, pp. 69–75.

Hou, P. S., 2005. Nodal reordering strategies to improve preconditioning for finite element systems. Master's Thesis, Virginia Polytechnic Institute and State University.

Kelley, D. W., Gago, J. P. S. R., Zienkiewicz, O. C., & Babuska, I., 1983. A posteriori error analysis and adaptive processes in the finite element method. Part I: Error analysis. *International Journal for Numerical Methods in Engineering*, vol. 18, pp. 1593–1619.

Langtangen, H. P., 1989. Conjugate gradient methods and ILU preconditioning of non-symmetric matrix systems with arbitrary sparsity patterns. *Internat. J. Numer. Methods Fluids*, vol. 9, pp. 213–233.

LibMesh, visited 05-29-2006. libmesh home page. http://libmesh.sourceforge.net/.

Oliker, L., Li, X., Husbands, P., & Biswas, R., 2002. Effects of ordering strategies and programming paradgms on sparse matrix computations. *SIAM Review*, vol. 44, n. 3, pp. 373–393.

PETSc, visited 05-29-2006. Pestc home page. http://www-unix.mcs.anl.gov/petsc/petsc-2/.

Saad, Y., 1996. *Iterative Methods for Sparse Linear Systems*. PWS Publishing, Boston.

Spotz, W. F. & Carey, G. F., 1995. High-order compact scheme for the esteady stream-function vorticity equations. *International Journal for Numerical Methods in Engineering*, vol. 38, pp. 3497–3512.

Table 5: Lid-driven cavity $Re = 400$ - Structured Triangular mesh - GMRES(30) - Ordering Evaluation

Natural Ordering				
Preconditioner	AMR(20×20,0,0.3,0.0,2)		Fixed Mesh (40×40)	
	$nz(A)$	$nz(M)/nz(A)$	$nz(A)$	$nz(M)/nz(A)$
ILU(0)	110903	1.0	431609	1.0
ILU(1)	349193	0.32	1388419	0.31
ILU(4)	1456243	0.08	6224865	0.07
RCM Ordering				
Preconditioner	AMR(20×20,0,0.3,0.0,2)		Fixed Mesh (40×40)	
	$nz(A)$	$nz(M)/nz(A)$	$nz(A)$	$nz(M)/nz(A)$
ILU(1)	148569	0.75	583943	0.74
ILU(4)	301161	0.37	1282355	0.33