Journal of Mathematics and Statistics 7 (4): 255-261, 2011
ISSN 1549-3644

A Comparative Study of Block Preconditioners for Incompressible Flow Problems

Jia Liu
Department of Mathematics and Statistics,
University of West Florida Pensacola, Florida, 32514, USA

Abstract: Problem statement: We consider the numerical solvers for the linearized Navier-Stokes problem. Both the Stokes problem and Oseen problems are considered. **Approach:** We used the Mark and Cell (MAC) discretization method to discretize the Navier-Stokes equations. We used preconditioned Krylov subspace methods to solve the resulting linear systems. **Results:** Numerical experimental results are performed to compare the different preconditioners. **Conclusion:** The choice of the preconditioner is highly problem dependent and we give the suggestions for individual cases.

Key words: Fluid mechanics, iterative methods, navier-stokes, stokes problem, oseen problem, hermitian and skew-hermitian splitting

INTRODUCTION

We study numerical solution methods of the incompressible viscous fluid problem. For an open bounded domain $\Omega \subset R^d$ (d = 2, 3) with boundary, time interval [0,T] and data f, g and u_2 , we aim to find a velocity field u = u(x,t) and pressure field p = p(x,t) such that:

$$\frac{\partial u}{\partial u} - v\Delta u + (u.\nabla)u + \nabla p = f \text{ in } \Omega \times [0,\Gamma] \quad (1)$$

$$\nabla.u = 0 \text{ in } \Omega \times [0,\Gamma] \quad (2)$$

$$Bu = g \text{ in } \vartheta\Omega \times [0,\Gamma] \quad (3)$$

$$u(x,0) = u_0 \text{ in } \Omega \quad (4)$$

Equation 1 represents the conservation of momentum and it is called the convection form of the momentum equation. Equation 2 represents the conservation of mass, since for an incompressible and homogeneous fluid the density is constant both with respect to time and the spatial coordinates. Equations 1-4 describe the dynamic behavior of Newtonian fluids, such as water, oil and other liquids. Acheson (1990) and Batchelor (2000) for more details. Here v is the kinematic viscosity, Δ is the Lapalcian, ∇ is the gradient, ∇ and is the divergence. We can use implicity discretization and linearization (for an example, Picard's iteration) of the Navier-Stokes equations to obtain a sequence of generalized Oseen problems of the form:

$$au - v\Delta u + (v.\nabla)u + \nabla p = f \text{ in in } \Omega \times [0,\Gamma] \quad (5)$$

$$\nabla.u = 0 \text{ in } \Omega \times [0,\Gamma] \quad (6)$$

$$Bu = g \text{ in } \vartheta\Omega \times [0,\Gamma] \quad (7)$$

$$u(x,0) = u_2 \text{ in} \Omega \quad (8)$$

where, v is a know velocity field from a previous iteration. And we call v a wind function. Here $a = 0(\frac{1}{\delta t})$, where δt is the time step. If a=0, we have the steady state Oseen problem Eq. 9-12:

$$-v\Delta u + (v.\nabla)u + \nabla p = f \text{ in } \Omega \times [0,\Gamma] \quad (9)$$

$$\nabla.u = 0 \text{ in } \Omega \times [0,\Gamma] \quad (10)$$

$$Bu = g \text{ in } \vartheta\Omega \times [0,\Gamma] \quad (11)$$

$$u(x,0) = u_0 \text{ in } \Omega \quad (12)$$

When the wind function is zero, we obtain the generalized Stokes problem Eq. 13-16:

$$au - v\Delta + \nabla p = f \text{ in } \Omega \times [0,\Gamma] \quad (13)$$

$$\nabla.u = 0 \text{ in } \Omega \times [0,\Gamma] \quad (14)$$

$$Bu = g \text{ in } \vartheta\Omega \times [0,\Gamma] \quad (15)$$

$$u(x,0) = u_0 \text{ in } \Omega \quad (16)$$

Again, if a=0, we will obtain the steady-state Stokes problem.

Discretization of Eq. 5-8 using a div-stable strategy leads to a linear system of the form Eq. 17:

$$A = \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \quad (17)$$

where, A is a discrete convection-diffusion operator, i.e., A=αI-vH+N. Here H is a discrete diffusion operator and N is a discrete convection operator. B and B^T are discrete divergence and gradient operators, respectively. In this study, we will use the Marker and Cell (MAC) discretization which is one of the div-stable discretization methods, (Elman *et al*., 2006).

Numerical methods for solving the saddle point linear system (17) are developed actively. However, all existing methods are not robust with respect to all problem parameters such as the time step and the viscosity. Once common approach for solving the Navier-Stokes equation is the preconditioned Krylov subspace method. However, the rate sof convergence of the Krylove submspace methods are very slow in general. We need to speed up the rate of convergence. This goal can be achieved by preconditioning. Preconditioning is a key ingredient for the success of Krylov subspace methods. Generally speaking, preconditioning is a transformation of the original system into another system such that the new system has more favorable properties for iterative solution. A *preconditioner* P is a matrix that effects such transformation. After we apply the preconditioner matrix P to the original matrix A, the preconditioned system $P^{-1}A$ is supposed to have a better spectral properties. If the matrix is symmetric, the rate of convergence of the Conjugate Gradient (CG) method or Minimum Residual Method (MINRES) depend on the distribution of the eigenvalues of the matrix A. If the preconditioned matrix $P^{-1}A$ has a smaller spectral condition number or the eigenvalues are clustered around 1, then we can expect a fast rate of convergence. For nonsymmetric (nonnormal) problems the situation is more complicated and the eigenvalues may not describe the convergence of nonsymmetric matrix iterations like General Minimum Residual Method (GMRES); see the discussion Elman *et al*. (2005). Nevertheless, a clustered spectrum (away from 0) often results in rapid convergence, especially if the departure from normality of the preconditioned matrix is not too high. We can find detailed discussions Elman *et al*. (2005).

Our aim of this study is to study the behavior of the difference preconditioners for the Navier-Stokes problems. We find out that even though there is no "ideal" preconditioners exist for all the cases, we could choose the one with the best performance under difference cases. The remainder of the study is organized as follows. Section 2 introduces the different preconditioners. Section 3 will provides the numerical experimental results for the preconditioners we have introduced. Based on the results of section 3, an analysis of the preconditioners will be given for the Navier-Stokes problems in section 4 and we will make acknowledgement in section 5.

MATERIALS AND METHODS

In this section, we will derive a series of block preconditioners based on the block factorization:

$$\begin{pmatrix} A & B \\ C & 0 \end{pmatrix} = \begin{pmatrix} I & 0 \\ CA^{-1} & I \end{pmatrix} \begin{pmatrix} A & B \\ 0 & -S \end{pmatrix}$$

where, S is the Schur complement. Therefore:

$$\begin{pmatrix} A & B \\ C & 0 \end{pmatrix} \begin{pmatrix} A & B \\ 0 & -S \end{pmatrix}^{-1} = \begin{pmatrix} I & 0 \\ CA^{-1} & I \end{pmatrix}$$

Based on the block factorization above, it is possible to use the matrix as a right-oriented preconditioner. Therefore it is very nature to choose the preconditioner of the following form:

$$P = \begin{pmatrix} A & B \\ 0 & -S \end{pmatrix}$$

This preconditioned system $P^{-1}A = \begin{pmatrix} I & 0 \\ CA^{-1} & I \end{pmatrix}$, which contains eigenvalues with the same value 1. It can be shown the preconditioned GMRES iteration would be finished at most two iterations. /cite[]. The bottleneck for the preconditioner we have proposed above is to calculate $P^{-1}x_k$ =w_k can be very expensive. The Schur complement is $S = CA^{-1}B$ where A^{-1} is a dense matrix. Inverting the matrix S requires solving a very expensive system. Therefore we have to replace the Schur complement by a relative easy matrix. Based on this idea, we have the following preconditioners to consider in general.

Block diagonal preconditioner: The basic block diagonal preconditioner is given by:

$$P_d = \begin{pmatrix} A & 0 \\ 0 & I \end{pmatrix}$$

here we choose the diagonal matrices and replace the Schur complement by the identity matrix. Thus the total cost of applying such a preconditioner only involves solving linear systems with the matrix A. If A is positive symmetric definite, then we can use many efficient methods like CG, Multigrid methods to solve the matrix A. The cost of this type of the preconditioner is low.

Block triangular preconditioner: The block triangular preconditioned is obtained by:

$$P_t = \begin{pmatrix} A & B \\ 0 & I \end{pmatrix}$$

We replace the Schur complement by the identity matrix. The total cost of applying such a preconditioner only involves solving linear systems with the matrix A and matrix vector products. The cost of the block triangular preconditioner is also very low and this preconditioned contains more information than the block diagonal preconditioned.

Uzawa preconditioner:

$$P_u = \begin{pmatrix} A & 0 \\ -\omega B & I \end{pmatrix}$$

where, $\omega$is a parameter. The Uzawa's preconditioner is also regarded as a lower block triangular preconditioner. This class of the preconditioners includes some of the most effective solvers for saddle points problems. Again, we replace the Schur complement by the identity matrix.

SIMPLE preconditioner: The SIMPLE scheme (Semi-Implicit Method for Pressure Linked Equations) is very popular in computational fluid dynamics. Consider the block preconditioner as follows:

$$P_s = \begin{pmatrix} A & 0 \\ -B & BD^{-1}B^T \end{pmatrix} \begin{pmatrix} I & -D^{-1}B^T \\ 0 & I \end{pmatrix}^{-1}$$

where, D is the main diagonal of A. This scheme was originally developed by Patankar and Spalding in (Paige and Saunders, 1975; Patankar, 1980) and there are many variants of this approach since then. The cost of SIMPLE preconditioner is one solve for A and one solve for the approximate Schur complement. The Schur complement system is a discrete elliptic scalar PDE which can be solved by Multigrid. Therefore SIMPLE is a relatively cheap preconditioner.

The pressure convection-diffusion preconditioner: The pressure convection-diffusion preconditioner (Elman *et al*., 2002) is defined as the following block triangular preconditioner:

$$P_{ap} = \begin{pmatrix} A & B^T \\ 0 & \hat{S} \end{pmatrix}$$

where, $\hat{S} = BB^T A_p^{-1}$, where A_p is the discrete (reaction) convection--diffusion operator on the pressure space. To implement such a preconditioner, solving the Schur complement part requires the action of a Poisson solve and a matrix-vector product with a specially constructed matrix A_p. In addition we need to perform the solves for A which are the same as the previous preconditioners.

The least-squares commutator preconditioner: Another approach for the approximation of the Schur complement gives us a new block triangular preconditioned:

$$P_{lsc} = \begin{pmatrix} A & B^T \\ 0 & bS \end{pmatrix}$$

where, $bS^{-1} = (B\ B^T)^{-1}\ BAB^T\ (BB^T)^{-1}$. This preconditioner was proposed by Elman *et al*. (2006) and this approach for approximating the Schur complement operator is only applicable when the discretization is uniformly stable (which is the case with MAC), (Elman *et al*., 2006) and (Elman *et al*., 2005). We refer to this type of preconditioner as the least-squares commutator preconditioner. In contrast to what is done for the pressure convection-diffusion preconditioner, this methodology does not require the explicit construction of the matrix A_p. Implementing this preconditioner, we need one solve for A, two solves for two discrete Poisson-type matrices B B^T and matrix-vector products with the matrices B, B^T and A. The main advantage of the least-square approach is that it is fully automated, that is, it is defined in terms of matrices that are available in the statement of the problem and it does not require the construction of the auxiliary operators A_p that are needed for the pressure convection-diffusion preconditioner. However, the cost of the least-squares commutator preconditioner is a little higher. It needs one more solve for the discrete Poisson matrix.

Hermitian and Skew-Hermitian (HSS) preconditioner: The Hermitian/Skew-Hermitian splitting (HSS) preconditioner is based on Herimitian and skew-Hermitian splitting of the coefficient matrix. Letting $H = \frac{1}{2}(A + A^{\tau}), K = \frac{1}{2}(A - A^{\tau})$ we have the following splitting of A into its symmetric and skew-symmetric parts:

$$A = \begin{pmatrix} A & B^T \\ -B & C \end{pmatrix} = \begin{pmatrix} H & 0 \\ 0 & C \end{pmatrix} + \begin{pmatrix} K & B^T \\ -B & 0 \end{pmatrix} = H + K$$

Note that H, the symmetric part of A, is symmetric positive semidefinite since H and C are. K is a skew symmetric matrix. Let ρ>0 be a parameter, the HSS preconditioner is defined as follows:

$$P_{hss} \frac{1}{2\rho}(H + \rho I_{m+N})(K + \rho I_{m+n})$$

where, I_{m+n} is the identity matrix of size m+n. To Solve this preconditioner, it requires solving a shifted Hermitian system and a shifted Skew Hermitian system. This preconditioner was first proposed by Benzi and Golub (2004). Then it is used as a preconditioner for the Oseen problem in rotation form by Benzi and Liu (2007). This preconditioner also has a good performance for the Stokes problem and the Oseen problem in convection form (which is the case we discuss).

RESULTS AND DISCUSSION

In this section, we will show the numerical experimental results for the Navier-Stokes problems with different preconditioned GMRES methods. All results were computed in MATLAB 7.1.0 on one processor of an AMD Opteron with 32 GB of memory. Numerical experiments are presented for the famous lid driven cavity problems in two dimensions. We have tested for both constant wind function and variable constant function. The linear iteration was stopped when the residual of the linear system satisfied.

Again in all experiments, symmetric diagonal scalings was applied before forming the preconditioners. We found that this scaling is not only beneficial to convergence, but also it makes finding (nearly) optimal values of the shift ρ easier. Of course, the right-hand side and the solution vector were scaled accordingly. We used right preconditioning in all cases.

The stokes type flow: Here we consider the generalized Stokes problem on the unit square. The right-hand side is given by f(x,y)= (sin(πx)sin(πy),0). The computational domain is the unit square for two dimensional problems. The equations were discretized with the MAC scheme with a uniform mesh size h. The outer iteration (full GMRES) was stopped when $\frac{\|r_k\|_2}{\|r_0\|_2} < 10^{-6}$, where r_k denotes the residual vector at step k. For the results presented in this section, the symmetric positive definite systems were solved ‘exactly’ by means of the sparse Cholesky factorization available in MATLAB, in combination with an approximate minimum degree ordering to reduce fill-in. For the sparse, nonsymmetric Schur complement system we used the sparse LU solver available in MATLAB with the original (lexicographic) ordering. We found this to be faster than a minimum degree ordering, probably because the need for pivoting makes the fill-reducing ordering ineffective or even harmful.

Figure 1 and 2 are obtained using the Incompressible Flow Iterative Solution Software (IFISS) by Elman and co-workers. These figures show the computed solutions of the Navier--Stokes problem. Figure 1 shows the solution of the Stokes problem for a leaky driven cavity problem; while Fig. 2 is the solution of a (regularized) driven cavity problem on a square domain, which is a fast-flowing analogue of the Stokes flow in a cavity. The solution shown in Fig. 2 corresponds to a viscosity of 0.001. Note the recirculation at the bottom corners.

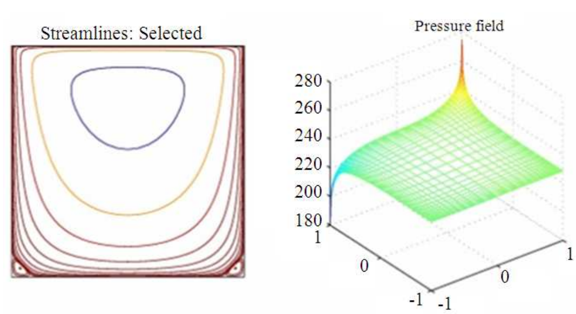

Fig. 1: The solution of the Stokes problem for a leaky driven cavity problem

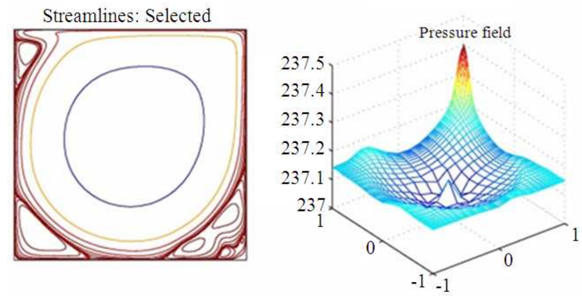

Fig. 2: The solution of a (regularized) driven cavity problem on a square domain

Table 1: Iteration number of the Stokes problem

Grid size	Diagonal	Triangular	Uzawa	HSS	Simple
8 by 8	15	10	12	22	18
16 by 16	17	11	13	29	28
32 by 32	17	12	14	38	45
64 by 64	19	13	15	51	71
128 by 128	21	13	15	76	109

Table 2: Iteration number of the unsteady Stokes problem with time step 1/20

Grid size	Diagonal	Triangular	Uzawa	HSS	Simple
8 by 8	25	13	13	9	16
16 by 16	29	15	15	10	27
32 by 32	29	15	15	11	42
64 by 64	31	16	16	13	53
128 by 128	33	17	17	17	67

Table 3: Iteration number of the unsteady Stokes problem with time step 1/100

Grid size	Diagonal	Triangular	Uzawa	HSS	Simple
8 by 8	31	16	16	12	13
16 by 16	35	18	18	13	22
32 by 32	38	19	19	13	38
64 by 64	39	20	20	13	39
128 by 128	39	20	20	15	41

It is well know that for the Stokes problem (both steady and unsteady), there exist many optimal solvers. As is clearly shown in Table 1, the block diagonal preconditioner, the block triangular preconditioner and the Uzawa preconditioner are all ideal preconditioners for the steady-state Stokes problem when we approximate the Schur complement by I. The reason is that, as already mentioned, for the Stokes problem, I turns out to be a good approximation of the Schur complement S. Especially for the block triangular preconditioner and the Uzawa preconditioner, the iteration counts are independent of the mesh size. For the block diagonal preconditioner, there is a very tiny increase for grid sizes 64 and 128. Notice that the optimal parameter ω in the Uzawa preconditioner is always around 0.9-1.0. Therefore, Uzawa preconditioner is quite close to the block triangular preconditioner, which is the reason why the behaviors of the block triangular preconditioner and Uzawa preconditioner are similar. Unfortunately, the behavior of SIMPLE is not competitive since the iteration count is strongly mesh size dependent. We can see that from the table that as the mesh size doubles, the iteration counts of the SIMPLE preconditioner increase by 50%.

For the unsteady Stokes problem, these preconditioners have similar behaviors. However, since in this case A is a discrete shifted Poisson operator, the Schur complement $S=BA^{-1}B^T$ is not close to the identity unless a is small. Experimental results show that for the unsteady problem, the block triangular preconditioner, Uzawa preconditioner and HSS preconditioners are the best. Iteration counts of GMRES with the block triangular preconditioner or the Uzawa preconditioner, (in this case, best ω=0.9or 1) are independent of mesh size. Iteration counts of GMRES also independent of the mesh size and time steps with the HSS preconditioner. The results in Table 2 and 3 show the behaviors of the different block preconditioning. Table 2 is the iteration counts for the unsteady Stokes problem with time step a=20 and Table 3 is the iteration counts for the unsteady Stokes problem with the time step a = 100. Both tables show that, for the HSS and block triangular preconditioners, iteration counts are bounded as the mesh size grows or time steps changes. For the block diagonal preconditioner, even though the iteration counts are bounded with respect to the time step and mesh size, the iteration number is larger than the block triangular or HSS preconditioners. We also can see that the SIMPLE preconditioner is not recommended. We can see that iteration count depends on the mesh size. Although it is robust with respect to the time step. As time steps become smaller, iteration counts decrease; however, iteration counts still increase as mesh size goes to zero ven for the smallest time step parameter a=0.

The Oseen flow: Here we consider linear systems arising from the discretization of the Oseen problems. Again the computational domain is the unit square for two dimensional problems. We used Dririchelet boundary conditions. For the wind function, we choose the constant wind and variable wind. Since the performances of those two situation are quite similar, we only introduce the results for constant wind functions here.

Table 4: Iteration number of the steady-state Oseen problem with viscosity v = 0.1

Grid size	Diagonal	Triangular	Uzawa	Simple	HSS	Convection -diffusion	Lest square
8 by 8	67	34	21	22	19	9	27
16 by 16	77	39	23	31	25	9	39
32 by 32	85	43	25	48	34	13	59
64 by 64	91	46	27	75	51	17	85
128 by 128	95	48	29	116	72	18	91

Table 5: Iteration number of the steady-state Oseen problem with viscosity v = 0.01

Grid size	Diagonal	Triangular	Uzawa	Simple	HSS	Convection -diffusion	Lest square
8 by 8	127	71	64	59	15	27	22
16 by 16	459	270	178	78	19	24	31
32 by 32	459	334	187	54	25	25	43
64 by 64	685	343	189	66	36	29	55
128 by 128	>1000	356	190	107	57	30	63

Table 6: Iteration number of the steady-state Oseen problem with viscosity v = 0.001

Grid size	Diagonal	Triangular	Uzawa	Simple	HSS	Convection -diffusion	Lest square
8 by 8	127	127	70	63	14	54	44
16 by 16	479	507	260	120	14	93	24
32 by 32	479	>1000	822	208	17	74	34
64 by 64	>1000	>1000	>1000	172	23	72	51
128 by 128	>1000	>1000	>1000	130	32	72	51

Table 7: Iteration counts of simple preconditioned GMRES for the unsteady-state Oseen problem with viscosity v = 0.1

Grid size	α = 1	α = 10	α = 20	α = 50	α = 100
8 by 8	22	17	14	11	8
16 by 16	30	26	23	17	13
32 by 32	47	42	38	29	22
64 by 64	74	69	63	51	38
128 by 128	115	109	103	94	72

Table 8: Iteration counts of SIMPLE preconditioned GMRES for the unsteady-state Oseen problem with viscosity v = 0.001

Grid size	α = 1	α = 10	α = 20	α = 50	α = 100
8 by 8	57	27	17	10	7
16 by 16	93	39	24	13	9
32 by 32	52	40	32	20	12
64 by 64	65	55	46	31	20
128 by 128	106	92	80	56	38

Table 9: Iteration counts of HSS preconditioned GMRES for the unsteady-state Oseen problem with viscosity v = 0.1

Grid size	α = 1	α = 10	α = 20	α = 50	α = 100
8 by 8	18	15	13	13	14
16 by 16	25	22	19	16	15
32 by 32	34	31	29	23	19
64 by 64	60	44	41	37	49
128 by 128	72	53	52	44	51

Table 9: Iteration counts of HSS preconditioned GMRES for the unsteady-state Oseen problem with viscosity v = 0.001

Grid size	α = 1	α = 10	α = 20	α = 50	α = 100
8 by 8	10	10	11	13	15
16 by 16	12	10	12	14	15
32 by 32	16	11	11	14	16
64 by 64	22	16	13	14	16
128 by 128	27	17	13	16	16

Table 4-6 show the results of the iteration counts for convergence of the preconditioned GMRES solver on the steady-state Oseen problem with viscosity v=0.1 to v=0.001. The results show that the block diagonal, block triangular, Uzawa and SIMPLE preconditioners should not be considered as the preconditioners for the Oseen problem. The number of the iterations strongly depends on the mesh size. For the relative large viscosity cases, the pressure convection diffusion preconditioners wins and for the smaller viscosity cases, the HSS preconditioner works better. The least square preconditoner is also a good choice. Even though the number of the iterations is higher than the other two preconditioners, the cost of this preconditioner is the lowest.

For the unsteady oseen problem, numerical experiments show the block diagonal, block triangular, or Uzawa preconditioners are the ones we should avoid. For the SIMPLE preconditioner, it works well for the smaller time steps. Those results can be explained by the construction of the preconditioners. For the unsteady Oseen or steady state Oseen problem, the Schur complement is given by $S=BA^{-1}B^T$. Here A is not an identity matrix. Therefore, it is not a good approximation of the Schur complement if we replace S by I in the (2,1) block of the preconditioner. However, for the unsteady Oseen problem with the SIMPLE preconditioner, diag(A) can be a good approximate of the matrix A once the step time is small (which means a is large). For the pressure convection diffusion preconditioner or the least square communicator preconditioner, the y only works well for the large viscosity. Among all the preconditioners we have introduced, HSS preconditioner will be the "best" choice for the unsteady Oseen problem.

Table 7 and Table 8 show iteration counts for the SIMPLE preconditioned GMRES methods of the unsteady Oseen problem with viscosity 0.1 or 0.001. From the both tables, we observe that the SIMPLE preconditioner has a better performance when the time steps get smaller.

Table 8 and Table 9 are the experimental results for the iteration counts of HSS preconditioned GMRES for the unsteady Oseen problem with viscosity 0.1 and 0.01. We can see that the HSS preconditioner works even better for the smaller viscosity. The number of the iterations is bounded by 20 for most of the cases and it is independent of the mesh size, viscosity and time step.

CONCLUSION

The purpose of this study was to explore the properties of the preconditioned Krylov subspace methods for the Navier-Stokes equation. We introduce seven most popular preconditioners for the linearized Naveri-Stokes equation. We analysis the construction, computation cost, performance of each preconditioner. Numerical experimental results are given for both the Stokes problem and the Oseen problem. We find out there is no univercial “best” preconditioner for all the problems. The choice of the preconditioner is strongly case dependent.

For the steady-state Stokes problem, the best preconditioner is the block triangular preconditioner. The convergence rates are independent of discretization mesh size. The computing cost only involve solving a Possion type equation. For the unsteady-State Stokes probmes, both the block triangular preconditioner and HSS preconditioner works well. Especially for the HSS preconditioner, the convergence rate are independent of the discretization mesh size, viscosity and the time step.

For the Oseen problem, we conclude that the HSS preconditioner, pressure convection-diffusion preconditioner and the communicator preconditioner are the better choices. The block triangular, block diagonal, Uzawa and SIMPLE preconditioners are not suitable for the Oseen problem anymore. If we consider the steady state Oseen problem, we will recommend to choose the pressure convection-diffusion preconditioner or the communicator preconditioner. In this case, the h-independent convergence rates are observed for both preconditioners. For the unsteady Oseen problem, then HSS preconditioner will be a good choice. The rate of convergence is independent of the mesh size, viscosity and the time step. We also notice that the HSS preconditioner works better for the smaller viscosity. However, most existing methods get worse as the viscosity goes smaller.

Finally, the numerical experiments have been limited to the GMRES for the Krylov subspace method. However we expect other solvers to perform similar. The results are also similar if we use other discretization methods such as finite element method.

ACKNOWLEDGMENT

The researchers would like to thank Michele Benzi for helpful instructions and suggestions.

REFERENCES

Acheson, D.J., 1990. Elementary Fluid Dynamics. 1st Edn., Oxford University Press, Oxford, ISBN: 0198596790, pp: 397.

Batchelor, G.K., 2000. An Introduction to Fluid Dynamics. 2nd Edn., Cambridge University Press, Cambridge, ISBN: 0521663962, pp: 615.

Benzi, M. and G.H. Golub, 2004. A preconditioner for generalized saddle point problems. SIAM J. Matrix Anal. Appl., 26: 20-41. DOI: 10.1137/S0895479802417106

Benzi, M. and J. Liu, 2007. An Efficient solver for the incompressible navier-stokes equations in rotation form. SIAM J. Sci. Comput., 29: 1959-1981. DOI: 10.1137/060658825

Elman, H.C., D.J. Silvester and A.J. Wathen, 2002. Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations. Numer. Math., 90: 665-688. DOI: 10.1007/s002110100300

Elman, H.C., D.J. Silvester and A.J. Wathen, 2005. Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics. 1st Edn., Oxford University Press, Oxford, ISBN: 019852868X, pp: 400.

Elman, H., V.E. Howle, J. Shadid, R. Shuttleworth and R. Tuminaro, 2006. Block preconditioners based on approximate commutators. SIAM J. Sci. Comput., 27: 359-373. DOI: 10.1137/040608817

Paige, C.C. and M.A. Saunders, 1975. Solution of sparse indefinite systems of linear equations, SIAM J. Numer. Anal., 12: 617-629. DOI: 10.1137/0712047

Patankar, S.V., 1980. Numerical Heat Transfer and Fluid Flow. 1st Edn., McGraw-Hill, New York, ISBN: 0070487405, pp: 197.