Prepared for
Computational Mechanics

Murat Manguoglu · Ahmed H. Sameh · Tayfun E. Tezduyar · Sunil Sathe

A nested iterative scheme for computation of incompressible flows in long domains

Abstract We present an effective preconditioning technique for solving the nonsymmetric linear systems encountered in computation of incompressible flows in long domains. The application category we focus on is arterial fluid mechanics. These linear systems are solved using a nested iterative scheme with an outer Richardson scheme and an inner iteration that is handled via a Krylov subspace method. Test computations that demonstrate the robustness of our nested scheme are presented.

Keywords Incompressible flows, arterial fluid mechanics, long domains, nested iterative schemes, preconditioners

1 Introduction

Computer modeling in arterial fluid mechanics, especially the fluid–structure interaction (FSI) aspect of this class of applications, has been receiving much attention in recent years (see, for example, [1; 2; 3; 4; 5; 6; 7; 8; 9; 10]). This calls for development of effective iterative solution techniques for the nonsymmetric linear systems encountered in computation of incompressible flows in long domains. The stabilized space–time FSI (SSTFSI) technique introduced recently in [11] is one of the FSI modeling technique applied to arterial fluid mechanics, with a number of test computations presented in [7; 9]. The SSTFSI technique is based on the new versions [11] of the Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) formulation [12; 13; 14; 15].

We present here a nested iterative scheme for the nonsymmetric linear systems generated in arterial fluid mechanics computations, and test this scheme in conjunction with the fluid mechanics part of the SSTFSI-TIP1 technique (see Remarks 5 and 10 in [11] for this specific version of the SSTFSI technique). As test problem, we consider the carotid artery bifurcation computed in [7]. The model is shown in Figure 1. The linear systems that arise in our test problem

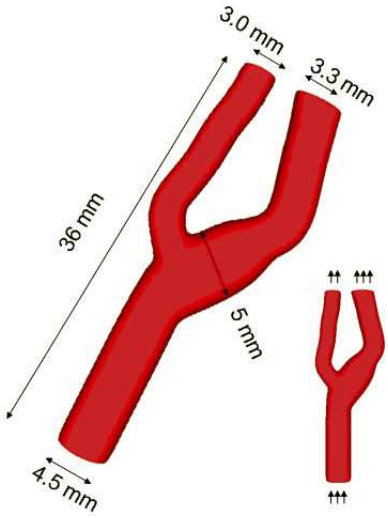

Fig. 1 Carotid artery bifurcation. The model and mesh are from [7].

have the following general 2×2 block form:

$$\begin{bmatrix} A & B \\ C^T & D \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \tag{1}$$

where $A \in \mathbb{R}^{n \times n}$ is nonsymmetric, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{n \times m}$, and $D \in \mathbb{R}^{m \times m}$, with $m < n$. Here, D is positive semidefinite, $n = 110,052$ and $m = 23,392$. The linear systems we consider here were extracted from the first and the last nonlinear Newton–Rhapson iterations at a given time step.

Our method is based on first reordering the $(1,1)$ and $(2,2)$ blocks so as to minimize the bandwidths of A and D. This is followed by extracting a narrow banded preconditioner. We study the performance of the diagonal, global

Murat Manguoglu and Ahmed H. Sameh
Department of Computer Science
Purdue University
305 N. University Street
West Lafayette, IN 47906, USA

Tayfun E. Tezduyar and Sunil Sathe
Mechanical Engineering, Rice University – MS 321
6100 Main Street, Houston, TX 77005, USA

$ILU(0)$, and block-diagonal preconditioned BiCGStab schemes as well as our nested iterative scheme.

2 Algorithms

2.1 Reordering

Before reordering, the coefficient matrix has the sparsity structure given in Figure 2. Using the Reverse Cuthill-

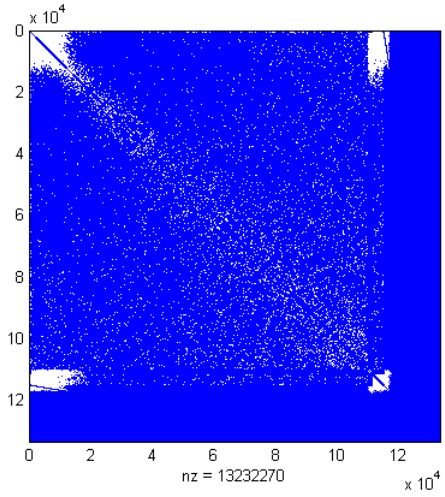

Fig. 2 Sparsity structure of the coefficient matrix before reordering.

McKee [16] scheme, we obtain the symmetric permuatations P_A and P_D of A and D, respectively. We apply the permutations to the whole system as follows:

$$\begin{bmatrix} P_A & 0 \\ 0 & P_D \end{bmatrix} \begin{bmatrix} A & B \\ C^T & D \end{bmatrix} \begin{bmatrix} P_A^T & 0 \\ 0 & P_D^T \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{p} \end{bmatrix} = \begin{bmatrix} P_A & 0 \\ 0 & P_D \end{bmatrix} \begin{bmatrix} f \\ g \end{bmatrix}, \quad (2)$$

which yields the reordered system

$$\begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C}^T & \hat{D} \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{p} \end{bmatrix} = \begin{bmatrix} \hat{f} \\ \hat{g} \end{bmatrix}. \quad (3)$$

This reordering of the original system needs be done once since the sparsity structure of the coefficient matrix does not change during the nonlinear iterations. To simplify the notation, we assume that the system

$$\begin{bmatrix} A & B \\ C^T & D \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \quad (4)$$

is the one resulting after the reordering. The sparsity structure of the reordered coefficient matrix is shown in Figure 3. We note that the banded blocks are much narrower.

In this paper, we consider only the sequential versions of our preconditioned iterative schemes. If one wishes to consider their implementations on parallel architectures, the

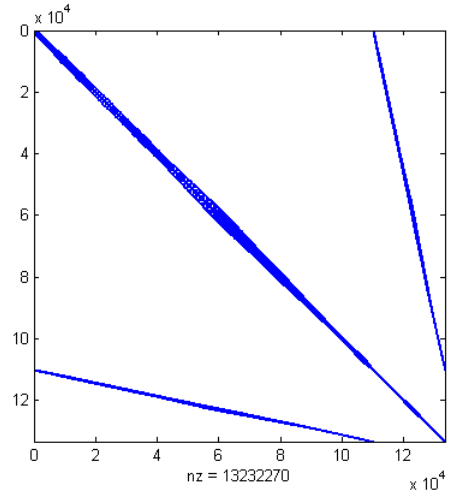

Fig. 3 Sparsity structure of the coefficient matrix after reordering.

compute $\hat{D} = D + \alpha I$;
compute ILU(0) factorization of $A \approx \tilde{A} = \tilde{L}_A \tilde{U}_A$;
compute LU factorization of $\hat{D} = L_{\hat{D}} U_{\hat{D}}$;
solve $\begin{bmatrix} A & B \\ C^T & D \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$ via BiCGStab with $rel.res. \leq \varepsilon_{out}$
  with preconditioner $M = \begin{bmatrix} \tilde{A} & 0 \\ 0 & \hat{D} \end{bmatrix}$;
  solve $M \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$
    solve $\tilde{A}v = a$ via triangular solves ;
    solve $\hat{D}w = b$ via MKL-PARDISO ;
  end
end

Fig. 4 Block-diagonal preconditioner.

SPIKE scheme [17; 18; 19; 20] can be a viable parallel system solver in the preconditioning step, taking the maximum advantage of the banded nature of the diagonal blocks.

2.2 Block-diagonal preconditioner

The block-diagonal preconditioned BiCGStab [21] scheme is illustrated in Figure 4. The first diagonal block $\tilde{A}$ in the preconditioner

$$M = \begin{bmatrix} \tilde{A} & 0 \\ 0 & \hat{D} \end{bmatrix} \quad (5)$$

contains the incomplete LU factors of A with zero fill-in (e.g. see [22]). The matrix $\hat{D}$ is a regularized version of the positive semidefinite matrix D, given by $D + \alpha I$, in which α is a small, positive parameter. Here, we factorize $\hat{D}$ directly via MKL-PARDISO [23].

compute $\hat{D} = D + \alpha I$;
compute ILU(0) factorization of $A \approx \tilde{A} = \tilde{L}_A \tilde{U}_A$;
compute LU factorization of $\hat{D} = L_{\hat{D}} U_{\hat{D}}$;
solve $\begin{bmatrix} A & B \\ C^T & D \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$ via Richardson iterations with $rel.res. \leq \varepsilon_{out}$
  $\begin{bmatrix} u_{k+1} \\ p_{k+1} \end{bmatrix} = \begin{bmatrix} u_k \\ p_k \end{bmatrix} + M^{-1}(\begin{bmatrix} f \\ g \end{bmatrix} - \begin{bmatrix} A & B \\ C^T & D \end{bmatrix} \begin{bmatrix} u_k \\ p_k \end{bmatrix})$ where $M = \begin{bmatrix} \tilde{A} & B \\ C^T & \hat{D} \end{bmatrix}$;
  solve $M \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$
    solve $\tilde{A}v = a$ via triangular solves ;
    solve$Sw = C^T v - b$ via BiCGStab ($rel.res. \leq \varepsilon_{in}$) with preconditioner $\hat{M} = \hat{D}$;
    (where $S = \hat{D} - C^T \tilde{A}^{-1} B$);
    solve $\tilde{A}v = a - Bw$ via triangular solves;
  end
end

Fig. 5 Nested preconditioner.

2.3 Nested iterative scheme

Inner-outer iterative schemes have been studied in [24; 25; 26]. The nested scheme we propose is illustrated in Figure 5. The outer iterative scheme consists of the Richardson iteration:

$$\begin{bmatrix} u_{k+1} \\ p_{k+1} \end{bmatrix} = \begin{bmatrix} u_k \\ p_k \end{bmatrix} + M^{-1}(\begin{bmatrix} f \\ g \end{bmatrix} - \begin{bmatrix} A & B \\ C^T & D \end{bmatrix} \begin{bmatrix} u_k \\ p_k \end{bmatrix}), \quad (6)$$

where

$$M = \begin{bmatrix} \tilde{A} & B \\ C^T & \hat{D} \end{bmatrix}. \quad (7)$$

Here $\tilde{A}$ and $\hat{D}$ are the same as those used in the block-diagonal preconditioned BiCGStab algorithm. Systems involving the preconditioner

$$M \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} \quad (8)$$

are solved by forming the block LDU factorization of M as follows:

$$M = \begin{bmatrix} \tilde{A} & 0 \\ C^T & I \end{bmatrix} \begin{bmatrix} \tilde{A}^{-1} & 0 \\ 0 & -S \end{bmatrix} \begin{bmatrix} \tilde{A} & B \\ 0 & I \end{bmatrix}, \quad (9)$$

where $S = \hat{D} - C^T \tilde{A}^{-1} B$. Systems involving $\tilde{A}$ are solved via forward and back substitutions, while those involving the Schur complement S are solved via BiCGStab with the preconditioner $\hat{D}$, and the systems involving $\hat{D}$ are solved via MKL-PARDISO.

3 Numerical experiments

All the numerical experiments were performed on a single processor of an Intel quad-core machine (Clovertown).

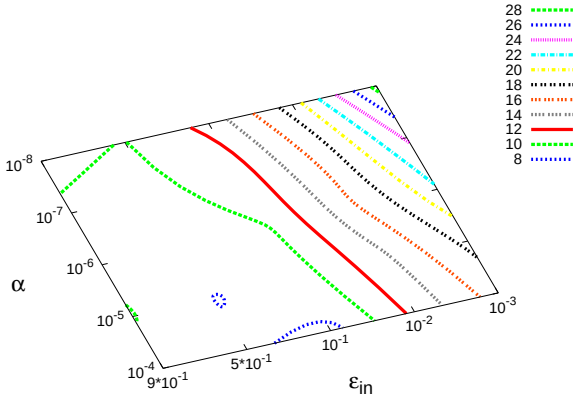

Fig. 6 Total time for the first nonlinear iteration, $\varepsilon_{out} = 10^{-2}$, various $(\alpha, \varepsilon_{in})$.

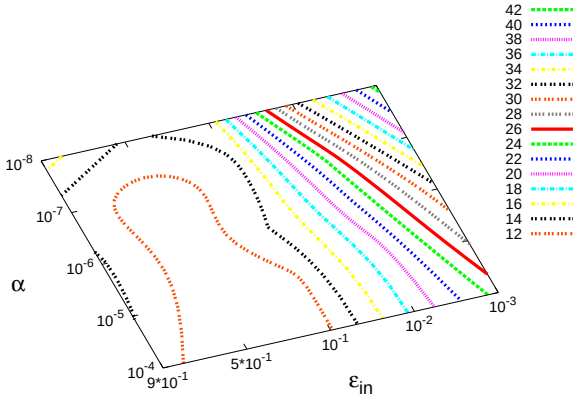

Fig. 7 Total time for the first nonlinear iteration, $\varepsilon_{out} = 10^{-3}$, various $(\alpha, \varepsilon_{in})$.

For the first nonlinear iteration, the initial residual norm is $||r_0||_\infty = 1.6 \times 10^{-3}$, while for the fifth (last) nonlinear iteration $||r_0||_\infty = 5.7 \times 10^{-5}$. The stopping criterion used is $||r_k||_\infty / ||r_0||_\infty \leq \varepsilon_{out}$, where we tested several values of ε_{out}: 10^{-2}, 10^{-3} and 10^{-5}.

For the nested scheme, we have two parameters, namely α and ε_{in}. The purpose of the first set of experiments is to seek an optimal pair $(\alpha^*, \varepsilon_{in}^*)$ that yields the best solution time. Here we only consider $\varepsilon_{out} = 10^{-2}$ and 10^{-3}. Figures 6, 7, 8 and 9 depict the total solution time (including the reordering and factorizations). The total number of BiCGStab iterations for the nested scheme are given in Figures 10, 11, 12 and 13.

For the block-diagonal preconditioned BiCGStab scheme, we have only one parameter, α. In Figures 14 and 15, the total solution time for various pairs of $(\alpha, \varepsilon_{out})$ is shown for the first and the fifth nonlinear iterations, respectively. The corresponding number of iterations for the block-diagonal preconditioned BiCGStab scheme is given in Figures 16 and

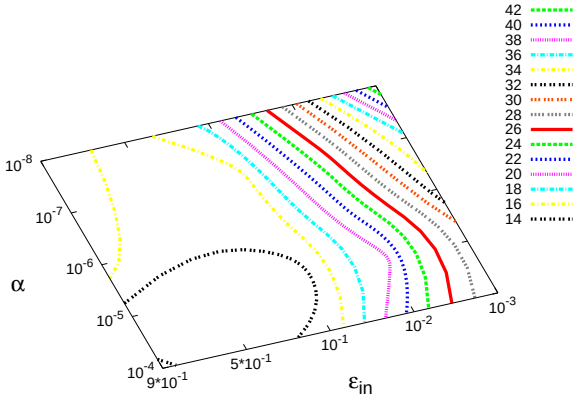

Fig. 8 Total time for the fifth nonlinear iteration, $\varepsilon_{out} = 10^{-2}$, various ($\alpha$, ε_{in}).

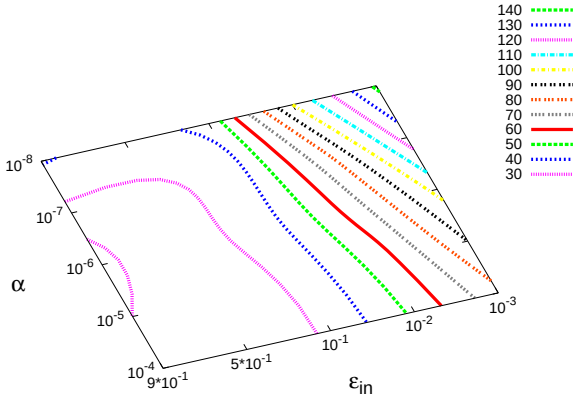

Fig. 11 Total number of inner BiCGStab iterations for the first nonlinear iteration, $\varepsilon_{out} = 10^{-3}$, various ($\alpha$, ε_{in}).

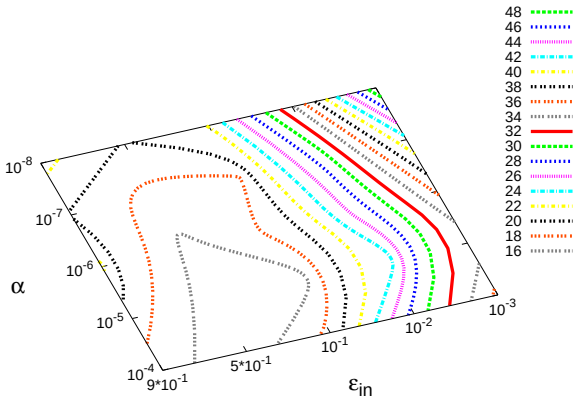

Fig. 9 Total time for the fifth nonlinear iteration, $\varepsilon_{out} = 10^{-3}$, various ($\alpha$, ε_{in}).

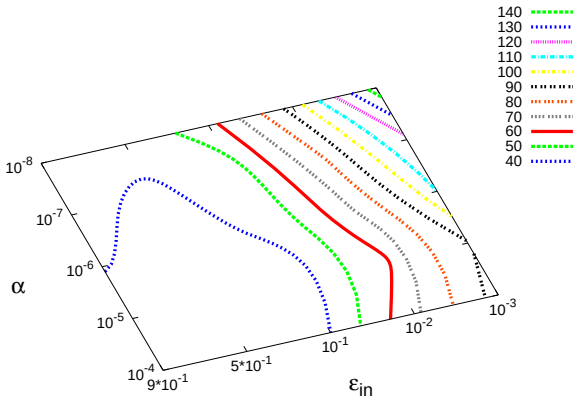

Fig. 12 Total number of inner BiCGStab iterations for the fifth nonlinear iteration, $\varepsilon_{out} = 10^{-2}$, various ($\alpha$, ε_{in}).

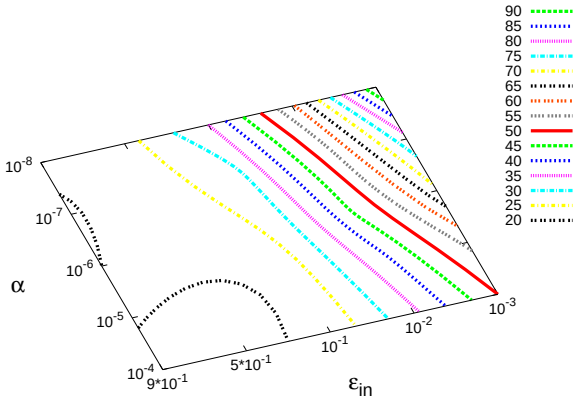

Fig. 10 Total number of inner BiCGStab iterations for the first nonlinear iteration, $\varepsilon_{out} = 10^{-2}$, various ($\alpha$, ε_{in}).

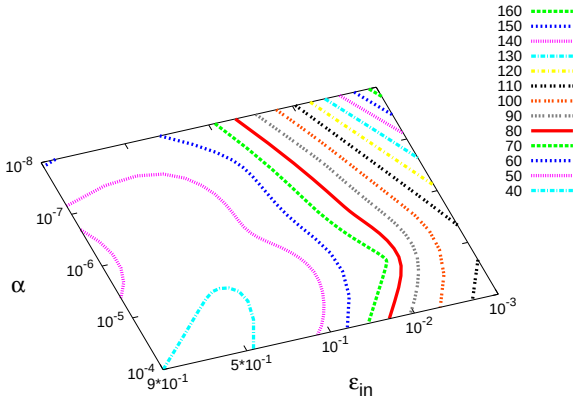

Fig. 13 Total number of inner BiCGStab iterations for the fifth nonlinear iteration, $\varepsilon_{out} = 10^{-3}$, various ($\alpha$, ε_{in}).

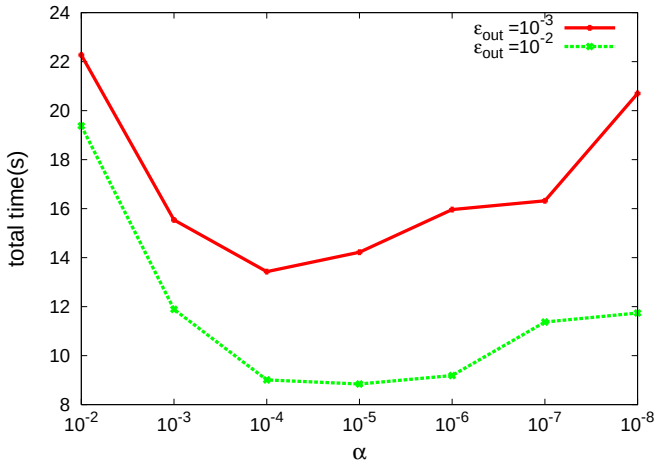

Fig. 14 Total time for the first nonlinear iteration, block-diagonal preconditioner, various (α).

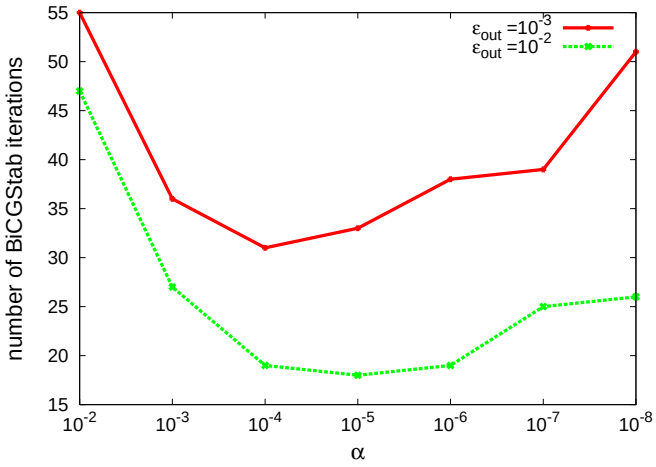

Fig. 16 Number of outer BiCGStab iterations for the first nonlinear iteration, block-diagonal preconditioner, various (α).

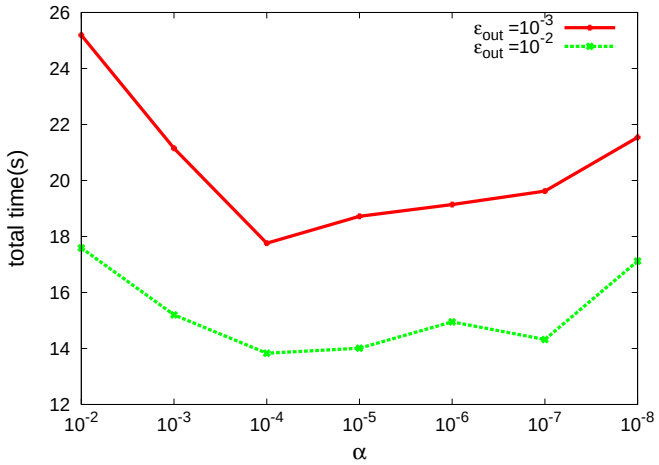

Fig. 15 Total time for the fifth nonlinear iteration, block-diagonal preconditioner, various (α).

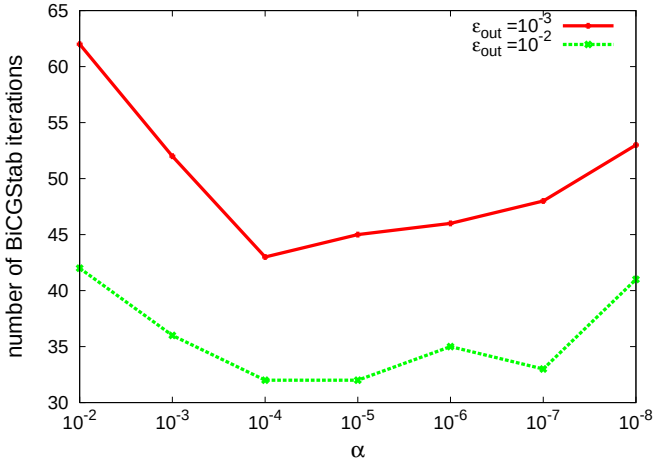

Fig. 17 Number of outer BiCGStab iterations for the fifth nonlinear iteration, block-diagonal preconditioner, various (α).

Table 1 First iteration, $\varepsilon_{out} = 10^{-2}$ and $\alpha = 1.0 \times 10^{-4}$.

Preconditioner	ε_{in}	Iterations Inner(Avg.)	Outer	Time(s)
ILU(0)	-	-	57	26.6
Diagonal	-	-	389	62.2
Block-Diagonal	-	-	19	9.0
Nested	5.0×10^{-1}	3.8	5	8.4

Table 2 First iteration, $\varepsilon_{out} = 10^{-3}$ and $\alpha = 1.0 \times 10^{-4}$.

Preconditioner	ε_{in}	Iterations Inner(Avg.)	Outer	Time(s)
ILU(0)	-	-	75	34.5
Diagonal	-	-	> 500	> 81.1
Block-Diagonal	-	-	31	13.4
Nested	5.0×10^{-1}	3.9	7	10.8

Table 3 First iteration, $\varepsilon_{out} = 10^{-5}$ and $\alpha = 1.0 \times 10^{-4}$.

Preconditioner	ε_{in}	Iterations Inner(Avg.)	Outer	Time(s)
ILU(0)	-	-	87	39.7
Diagonal	-	-	> 500	> 81.1
Block-Diagonal	-	-	74	28.6
Nested	5.0×10^{-1}	3.8	12	16.7

17. From Figures 6, 7, 8 and 9, we observe that the choice ($\alpha^* = 10^{-4}, \varepsilon_{in}^* = 5 \times 10^{-1}$) yields the best solution time for various outer stopping criteria. From Figures 14 and 15, we see that the choice ($\alpha^* = 10^{-4}$) yields the best solution time. In Tables 1, 2 and 3, we present the number of iterations and total solution time for the first nonlinear iteration using a global ILU(0), diagonal and block-diagonal precondtioned BiCGStab, as well as our nested scheme for $\varepsilon_{out} = 10^{-2}$, 10^{-3} and 10^{-5}, respectively. For $\varepsilon_{out} = 10^{-2}$, the best time is 8.4 seconds using the nested scheme, with the block-diagonal preconditioner being a close second, consuming

Table 4 Fifth iteration, $\varepsilon_{out} = 10^{-2}$ and $\alpha = 1.0 \times 10^{-4}$.

Preconditioner	ε_{in}	Iterations Inner(Avg.)	Outer	Time(s)
ILU(0)	-	-	74	33.5
Diagonal	-	-	325	52.0
Block-Diagonal	-	-	32	14.0
Nested	5.0×10^{-1}	4.0	8	12.9

Table 5 Fifth iteration, $\varepsilon_{out} = 10^{-3}$ and $\alpha = 1.0 \times 10^{-4}$.

Preconditioner	ε_{in}	Iterations Inner(Avg.)	Outer	Time(s)
ILU(0)	-	-	79	35.9
Diagonal	-	-	467	74.4
Block-Diagonal	-	-	43	17.8
Nested	5.0×10^{-1}	4.3	9	14.6

Table 6 Fifth iteration, $\varepsilon_{out} = 10^{-5}$ and $\alpha = 1.0 \times 10^{-4}$.

Preconditioner	ε_{in}	Iterations Inner(Avg.)	Outer	Time(s)
ILU(0)	-	-	92	42.0
Diagonal	-	-	> 500	> 81.1
Block-Diagonal	-	-	67	26.1
Nested	5.0×10^{-1}	4.2	14	20.4

9.0 seconds. For $\varepsilon_{out} = 10^{-3}$, the corresponding times for the two above schemes are 10.8 and 13.4 seconds, respectively. For $\varepsilon_{out} = 10^{-5}$, the best time is 16.7 seconds using the nested scheme, while the block-diagonal preconditioner consumes 28.6 seconds, almost twice that of the nested scheme.

Similarly, for the fifth nonlinear iteration the summary of the results are given in Tables 4, 5 and 6, where the nested scheme performs the best for all three stopping criteria.

In all cases, the diagonal and the global ILU(0) preconditioners consume much larger time. We note that one needs to find only an optimal α^* for the block-diagonal preconditioner, while for the nested scheme an optimal pair $(\alpha^*, \varepsilon_{in}^*)$ needs to be chosen. This can make the block-diagonal preconditioner more attractive even though it consumes larger time. The corresponding residual plots for the first and the fifth nonlinear iterations in combination with $\varepsilon_{out} = 10^{-2}$ and $\varepsilon_{out} = 10^{-5}$ are given in Figures 18, 19, 20 and 21. The diagonal preconditioner requires more than 300 iterations for both stopping criteria, and therefore is not included in the residual plots. Figures 18, 19, 20 and 21 illustrate clearly the rapid reduction of the residual for our nested scheme compared to the block-diagonal preconditioned BiCGStab algorithm. This observation is typical of what we see when we compare a nested scheme with outer Richardson iterations to a Krylov subspace method.

4 Conclusions

Even though the nested scheme requires the choice of two optimal parameters, it is robust and very effective. Moreover, its performance is not that sensitive to the choice of

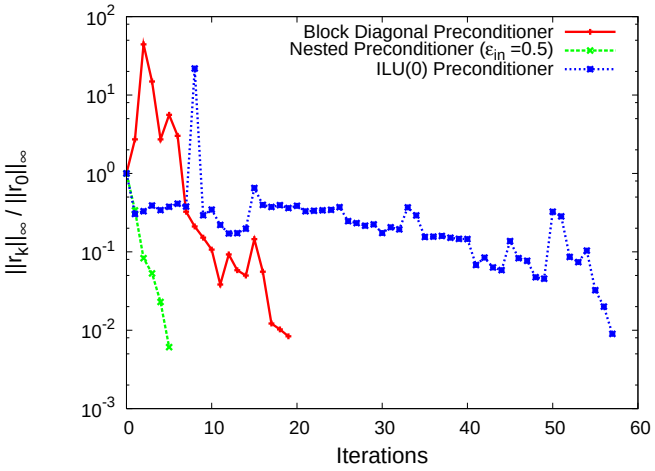

Fig. 18 Residual plots for the first nonlinear iteration, $\varepsilon_{out} = 10^{-2}$.

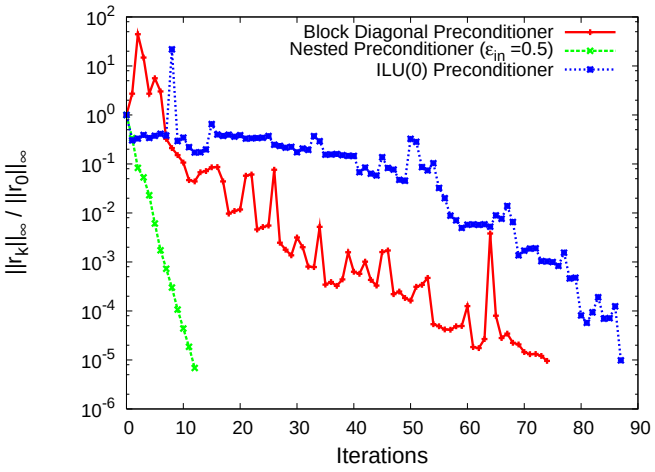

Fig. 19 Residual plots for the first nonlinear iteration, $\varepsilon_{out} = 10^{-5}$.

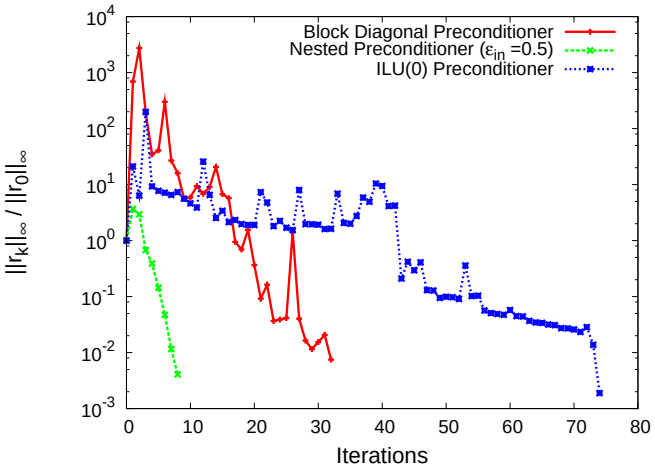

Fig. 20 Residual plots for the fifth nonlinear iteration, $\varepsilon_{out} = 10^{-2}$.

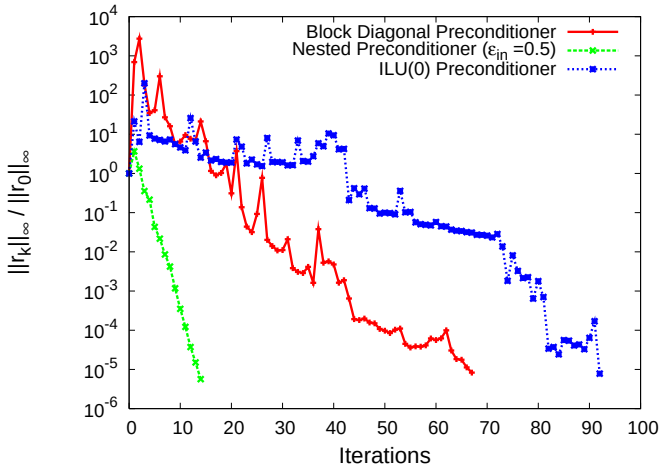

Fig. 21 Residual plots for the fifth nonlinear iteration, $\varepsilon_{out} = 10^{-5}$.

the optimal pair $(\alpha^*, \varepsilon_{in}^*)$. Furthermore, if a tighter outer stopping criterion is needed, the advantage of our nested scheme becomes more pronounced. Another advantage of the nested scheme is that it requires only inner products of much smaller vectors compared to the block-diagonal preconditioned BiCGStab method. This could be an advantage in solving very large problems on massively parallel computing platforms.

Acknowledgements This work has been partially supported by grants from NSF (NSF-CCF-0635169), DARPA/AFRL (FA8750-06-1-0233), and a gift from Intel. The efforts of the last two authors were supported in part by a Seed Grant from the Gulf Coast Center for Computational Cancer Research funded by John & Ann Doerr Fund for Computational Biomedicine and also in part by the Rice Computational Research Cluster funded by NSF under Grant CNS-0421109 and a partnership between Rice University, AMD and Cray. We would like to thank Eric Polizzi for allowing us to use the Intel Clovertown Quad-Core computing platform.

References

1. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Influence of wall elasticity on image-based blood flow simulation", *Japan Society of Mechanical Engineers Journal Series A*, **70** (2004) 1224–1231, in Japanese.
2. J.-F. Gerbeau, M. Vidrascu, and P. Frey, "Fluid–structure interaction in blood flow on geometries based on medical images", *Computers and Structures*, **83** (2005) 155–165.
3. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Computer modeling of cardiovascular fluid–structure interactions with the Deforming-Spatial-Domain/Stabilized Space–Time formulation", *Computer Methods in Applied Mechanics and Engineering*, **195** (2006) 1885–1895.
4. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Fluid–structure interaction modeling of aneurysmal conditions with high and normal blood pressures", *Computational Mechanics*, **38** (2006) 482–490.
5. Y. Bazilevs, V.M. Calo, Y. Zhang, and T.J.R. Hughes, "Isogeometric fluid–structure interaction analysis with applications to arterial blood flow", *Computational Mechanics*, **38** (2006) 310–322.
6. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Influence of wall elasticity in patient-specific hemodynamic simulations", *Computers & Fluids*, **36** (2007) 160–168.
7. T.E. Tezduyar, S. Sathe, T. Cragin, B. Nanna, B.S. Conklin, J. Pausewang, and M. Schwaab, "Modeling of fluid–structure interactions with the space–time finite elements: Arterial fluid mechanics", *International Journal for Numerical Methods in Fluids*, **54** (2007) 901–922.
8. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Numerical investigation of the effect of hypertensive blood pressure on cerebral aneurysm — Dependence of the effect on the aneurysm shape", *International Journal for Numerical Methods in Fluids*, **54** (2007) 995–1009.
9. T.E. Tezduyar, S. Sathe, M. Schwaab, and B.S. Conklin, "Arterial fluid mechanics modeling with the stabilized space–time fluid–structure interaction technique", *International Journal for Numerical Methods in Fluids*, published online, DOI: 10.1002/fld.1633, October 2007.
10. Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and Y. Zhang, "A fully-integrated approach to fluid–structure interaction", in preparation for *Computational Mechanics*, 2008.
11. T.E. Tezduyar and S. Sathe, "Modeling of fluid–structure interactions with the space–time finite elements: Solution techniques", *International Journal for Numerical Methods in Fluids*, **54** (2007) 855–900.
12. T.E. Tezduyar, "Stabilized finite element formulations for incompressible flow computations", *Advances in Applied Mechanics*, **28** (1992) 1–44.
13. T.E. Tezduyar, M. Behr, and J. Liou, "A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure: I. The concept and the preliminary numerical tests", *Computer Methods in Applied Mechanics and Engineering*, **94** (1992) 339–351.
14. T.E. Tezduyar, M. Behr, S. Mittal, and J. Liou, "A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders", *Computer Methods in Applied Mechanics and Engineering*, **94** (1992) 353–371.
15. T.E. Tezduyar, "Computation of moving boundaries and interfaces and stabilization parameters", *International Journal for Numerical Methods in Fluids*, **43** (2003) 555–575.
16. E. Cuthill and J. McKee, "Reducing the bandwidth of sparse symmetric matrices", in *Proceedings of the 1969 24th national conference*, ACM Press, New York, NY, USA, (1969) 157–172.
17. E. Polizzi and A.H. Sameh, "A parallel hybrid banded system solver: the SPIKE algorithm", *Parallel Comput.*, **32** (2006) 177–194.
18. J.J. Dongarra and A.H. Sameh, "On some parallel banded system solvers", *Parallel Computing*, **1** (1984) 223–235.
19. M.W. Berry and A. Sameh, "Multiprocessor schemes for solving block tridiagonal linear systems", *The International Journal of Supercomputer Applications*, **1** (1988) 37–57.
20. D.J. Kuck S.C. Chen and A.H. Sameh, "Practical parallel band triangular system solvers", *ACM Transactions on Mathematical Software*, **4** (1978) 270–277.
21. H.A. van der Vorst, "BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems", *SIAM J. Sci. Stat. Comput.*, **13** (1992) 631–644.
22. Y. Saad, "SPARSKIT: A basic tool kit for sparse matrix computations", Technical Report 90-20, NASA Ames Research Center, Moffett Field, CA, 1990.
23. O. Schenk, K. Gärtner, W. Fichtner, and A. Stricker, "PARDISO: a high-performance serial and parallel sparse linear solver in semiconductor device simulation", *Future Generation Computer Systems*, **18** (2001) 69–78.
24. A. Baggag and A. Sameh, "A nested iterative scheme for indefinite linear systems in particulate flows", *Comp. Meth. Appl. Mech. Engng*, **193** (2004) 1923–1957.

25. H.A. van der Vorst and C. Vuik, “GMRESR: A family of nested GMRES methods”, Technical Report DUT-TWI-91-80, Delft, The Netherlands, 1991.
26. V. Simoncini and D.B. Szyld, “Flexible inner-outer Krylov subspace methods”, *SIAM Journal on Numerical Analysis*, **40** (2002) 2219–2239.