Electronic Transactions on Numerical Analysis.
Volume 22, pp. 146-162, 2006.
Copyright © 2006, Kent State University.
ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

PARALLEL FULLY COUPLED SCHWARZ PRECONDITIONERS FOR SADDLE POINT PROBLEMS*

FENG-NAN HWANG† AND XIAO-CHUAN CAI‡

Abstract. We study some parallel overlapping Schwarz preconditioners for solving Stokes-like problems arising from finite element discretization of incompressible flow problems. Most of the existing methods are based on the splitting of the velocity and pressure variables. With the splitting, fast solution methods are often constructed using various fast Poisson solvers for one of the variables. More recently, several papers have investigated the so-called fully coupled approaches in which the variables are not separated. The fully coupled approach has some advantages over the variable splitting method when solving Stokes-like equations with many variables, where a good splitting may be hard to obtain. In this paper we systematically study the parallel scalability of several versions of the fully coupled Schwarz method for both symmetric and nonsymmetric Stokes-like problems. We show numerically that the performance of a two-level method with a multiplicative iterative coarse solver is superior to the other variants of Schwarz preconditioners.

Key words. saddle point problem, two-level Schwarz preconditioning, fully coupled methods, finite element, parallel processing

AMS subject classifications. 65F10, 65N30, 65N55

1. Introduction.

We study some parallel fully coupled Schwarz preconditioned iterative methods [36, 42, 43] for solving saddle point problems, which appear in many areas of computational science and engineering [3, 4, 21, 32, 35, 40, 44]. Due to the indefiniteness of saddle point problems the convergence of most iterative methods is often not guaranteed, and when they converge, the convergence rates of the methods are sometimes too slow to be considered practical, especially for large-scale applications. To resolve the situation, finding a good preconditioner is critical. In this paper, we focus on two types of saddle point problems, namely the symmetric Stokes problem discretized with a stabilized finite element method and the more general, nonsymmetric Jacobian system arising in the Newton-Krylov-Schwarz (NKS) algorithm for solving nonlinear incompressible Navier-Stokes equations. In NKS, the Jacobian system has to be solved as a part of the nonlinear iteration, and is often the most expensive part [17, 18, 23, 24, 25, 30]. The performance of Schwarz methods depends on several important parameters, including the overlapping size, the quality of the coarse and subdomain solutions, the coarse mesh size, as well as some physical parameters of the continuous equations. We perform a large number of numerical experiments to understand how these parameters affect the overall parallel performance of Schwarz methods for both symmetric and nonsymmetric saddle point problems. Our investigation is purely numerical, and to the best of our knowledge, the corresponding optimal convergence theory for Stokes-like problems is yet to be established.

Many iterative methods for solving saddle point problems are available, such as Uzawa's algorithms and their variants [7, 15, 33, 45], multigrid methods [5, 16, 23, 24, 25], Krylov subspace methods with block-type preconditioners [15, 17, 26, 27, 29, 41], and domain decomposition methods [6, 28, 29, 30, 34, 45]. Most of the existing work explicitly uses the

*Received December 15, 2004. Accepted for publication February 13, 2006. Recommended by E. de Sturler. The research was supported in part by the Department of Energy, DE-FC02-01ER25479, and in part by the National Science Foundation, CCR-0219190, ACI-0072089 and ACI-0305666.

†Department of Mathematics, National Central University, Jhongli City, Taoyuan County 32001, TAIWAN (hwangf@math.ncu.edu.tw).

‡Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309, USA (cai@cs.colorado.edu).

ETNA
Kent State University
etna@mcs.kent.edu

block structure of the discrete problem,

$$\begin{pmatrix} C & B^T \\ B & \alpha M_p \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{1.1}$$

where C is a symmetric positive definite matrix corresponding to a diffusion operator for the Stokes problem, B is a matrix corresponding to a discrete divergence operator, and M_p is a mass pressure matrix with a stabilization parameter α. The block structure is obtained by using a variable splitting-based ordering of the unknowns. First, the velocity variables for all mesh points are ordered, followed by the pressure variables for all mesh points. Even though the decoupled block structure is non-physical, i.e. it splits the variables, velocity and pressure, which are supposedly to be physically coupled at each mesh point, such algebraic structure is very useful in the design and analysis of algorithms. The popular pressure projection methods, such as the SIMPLE and SIMPLER algorithms [33], also known as Uzawa's algorithms, are typical examples of the *fully decoupled* solution strategy. These methods decouple the discrete system (1.1) into two subsystems, the Schur complement system for the pressure and the Laplacian system for the velocity; then Uzawa's algorithm can be constructed by solving these two subsystems iteratively. Also based on the block structure of (1.1), the block diagonal preconditioner is constructed by dropping both off-diagonal blocks [16, 18, 27, 29] and the block-triangular preconditioner is obtained by replacing the lower off-diagonal block with a zero block [17, 26, 29, 41].

In contrast to the fully decoupled approaches, the Schwarz preconditioned iterative methods are considered to be the *fully coupled* approaches in which the velocity and pressure variables are always coupled together throughout the computation. Without being restricted to the block structure, the fully coupled methods can be applied easily to other multi-component indefinite problems, such as the flow control problem [4, 21, 35], which is one of the target applications of our algorithms and software. Furthermore, it is also possible to generalize these preconditioners for the purpose of nonlinear preconditioning, similar to the additive Schwarz preconditioned inexact Newton algorithms [8, 9, 22]. The idea of the fully coupled methods for saddle point problems is not new, and the methods have been studied in the context of multigrid methods [5]. According to the multigrid results [23, 24, 25], the fully coupled methods seem to be a better choice for saddle point problems over the decoupled methods.

In this paper, we consider three types of Schwarz preconditioners. We begin by studying the one-level additive Schwarz preconditioner [28], which is defined by summing up the solutions of the Stokes problem defined on overlapping subdomains with certain artificial boundary conditions. The main advantage of the one-level Schwarz preconditioner is that all subproblems are independent of each other and can be solved in parallel. However, as expected, its convergence rate, in terms of the number of iterations, degenerates when the number of processors is large due to the lack of communication between subdomains. For a fully scalable method, a coarse mesh problem is required. Based on sequential numerical results [28], Klawonn and Pavarino showed that the number of GMRES [39] iterations for the two-level additive Schwarz methods for symmetric indefinite problems (Stokes and linear elasticity) with minimal overlap using exact subdomain and coarse solvers is independent of the mesh size and the number of subdomains, provided that the coarse mesh is sufficiently fine. However, the parallel performance of Schwarz methods for saddle point problems was not studied until the work in [44]. For the Stokes problem on unstructured meshes, Tuminaro *et al.* compared one-level methods with two-level methods using an inexact LU as the subdomain solver and SuperLU [13], a parallel direct LU solver, as the coarse solver. Their numerical experiments showed that although the two-level method seems to be scalable in

ETNA
Kent State University
etna@mcs.kent.edu

terms of the number of GMRES iterations, the computing time increases more than 200% from 64 to 256 processors. A similar problem also arises in the case of nonsymmetric indefinite problems (a thermal convection problem) solved by a GMRES-multigrid method using two meshes, in which Gauss-Seidel is used as the smoother and SuperLU is used as the coarse solver. One solution to the problem of high computing time is to replace SuperLU with a less expensive inexact solver, such as the one-step Schwarz-Richardson method as in [44]. Indeed, total computing time can be saved to a certain extent, but both the number of GMRES iterations and the computing time grow as the number of processors is increased. Therefore, it seems to us that these two coarse solvers are either too expensive or too inexact to be effective. In this paper, we propose and test a parallel preconditioned iterative coarse solver so that the cost of the coarse solve can be controlled easily by adjusting the coarse stopping condition. The preconditioner for the coarse mesh problem is the same as the one-level additive Schwarz preconditioner, except that it is constructed on a partitioned coarse mesh. For the two-level methods, the coarse mesh part of the preconditioner can be incorporated into its local subdomain part either additively or multiplicatively. All components of the two-level method are parallel, including the coarse solver, and the restriction and interpolation operators. We show, through parallel numerical experiments, that the performance of the two-level method with a multiplicative iterative coarse solver is superior to the other two variants of Schwarz preconditioners.

The paper is organized as follows. In Section 2, we describe the Stokes and Jacobian problems arising from finite element discretization of incompressible flow problems. We introduce three types of Schwarz preconditioners in Section 3. Then, in Section 4, we present some numerical results obtained on a parallel computer. Finally, conclusions and some remarks are given in Section 5.

2. Model problems and their stabilized finite element formulations. We first consider the two-dimensional steady-state Stokes problem [20, 37], which can be described as

$$\left\{\begin{array}{rll} -\triangle \boldsymbol{u} + \nabla p = \boldsymbol{f} & \text{in} & \Omega, \\ \nabla \cdot \boldsymbol{u} = 0 & \text{in} & \Omega, \\ \boldsymbol{u} = \boldsymbol{g} & \text{on} & \Gamma, \end{array}\right. \tag{2.1}$$

where $\boldsymbol{u} = (u_1, u_2)^T$ is the velocity, p is the pressure, and $\boldsymbol{f} = (f_1, f_2)^T$ is the body force. Here we assume that Ω is a bounded domain in R^2 with a polygonal boundary Γ. Since only Dirichlet boundary condition is considered, the pressure p is determined up to a constant. To make p unique, we impose the condition $\int_\Omega p\, dx = 0$.

To discretize (2.1), we use a stabilized finite element method on a given conforming quadrilateral mesh $\mathcal{T}^h = \{K\}$ [20]. For each element K, we denote by h_K as its diameter. Let V^h and P^h be a pair of finite element spaces for the velocity and pressure given by

$$V^h = \{\boldsymbol{v} \in (C^0(\Omega) \cap H^1(\Omega))^2 : \ \boldsymbol{v}|_K \in Q_1(K)^2, \ K \in \mathcal{T}^h \}$$

and

$$P^h = \{p \in C^0(\Omega) \cap L^2(\Omega) : \ p|_K \in Q_1(K), \ K \in \mathcal{T}^h\}.$$

Here, $C^0(\Omega)$, $L^2(\Omega)$, and $H^1(\Omega)$ are the standard notations with usual meanings in the finite element literature [20]. For simplicity, our implementation uses a $Q_1 - Q_1$ finite element (continuous bilinear velocity and pressure). The weighting and trial velocity function spaces V_0^h and V_g^h are defined as

$$V_0^h = \{\boldsymbol{v} \in V^h : \boldsymbol{v} = 0 \text{ on } \Gamma\} \text{ and } V_g^h = \{\boldsymbol{v} \in V^h : \ \boldsymbol{v} = \boldsymbol{g} \text{ on } \Gamma\}.$$

Similarly, let the finite element space P_0^h be both the weighting and trial pressure function spaces:

$$P_0^h = \left\{ p \in P^h : \int_\Omega p\, dx = 0 \right\}.$$

As suggested by [14], the stabilized finite element method for the steady-state Stokes problem reads: Find $\boldsymbol{u}^h \in V_g^h$ and $p^h \in P_0^h$, such that

$$B(\boldsymbol{u}^h, p^h; \boldsymbol{v}, q) = F(\boldsymbol{v}, q) \qquad \forall (\boldsymbol{v}, q) \in V_0^h \times P_0^h \tag{2.2}$$

with

$$\begin{aligned} B(\boldsymbol{u}, p; \boldsymbol{v}, q) = \ & (\nabla \boldsymbol{u}, \nabla \boldsymbol{v}) - (\nabla \cdot \boldsymbol{v}, p) - (\nabla \cdot \boldsymbol{u}, q) - \\ & \alpha \sum_{K \in \mathcal{T}^h} h_K^2 (-\triangle \boldsymbol{u} + \nabla p, \triangle \boldsymbol{u} + \nabla q)_K \end{aligned}$$

and

$$F(\boldsymbol{v}, q) = (\boldsymbol{f}, \boldsymbol{v}) - \alpha \sum_{K \in \mathcal{T}^h} h_K^2 (\boldsymbol{f}, \triangle \boldsymbol{v} + \nabla q)_K.$$

In [14], Douglas and Wang showed that this method is stable and has the optimal convergence for any choices of positive stabilization parameter α. We use a constant of $\alpha = 1$ throughout this paper. The equivalent matrix form of (2.2) can be written as

$$Ax = b, \tag{2.3}$$

where A is a large, sparse, symmetric indefinite matrix, and x is a vector corresponding to the nodal values of $\boldsymbol{u}^h = (u_1^h, u_2^h)$ and p^h in (2.2). In our implementation, we number the unknown nodal values in the order of u_1^h, u_2^h, and p^h at each mesh point. The mesh points are grouped subdomain by subdomain for the purpose of parallel processing. The subdomain partitioning will be discussed further in the next section. Note that the linear system (2.3) can be written explicitly in the same block-structure form as in (1.1) by rearrangement of the unknown variables in the order of u_1^h, u_2^h, and p^h. We solve the linear systems (2.3) by a right preconditioned Krylov subspace method, i.e.

$$AM^{-1}y = b, \text{ with } x = M^{-1}y, \tag{2.4}$$

where M^{-1} is called a right preconditioner to be defined in the next section.

In addition to the Stokes problem, we also consider some nonsymmetric indefinite Jacobian systems that we have to solve in order to obtain search directions when using NKS for solving incompressible Navier-Stokes equations. Briefly speaking, NKS is a general framework for solving a large, sparse, nonlinear system of equations, $F(x^*) = 0$, arising from a discretization of nonlinear partial differential equations. As its name suggests, NKS consists of three main components: a Newton-type method as the nonlinear solver, a Krylov subspace method as the linear solver, and a Schwarz-type method as the preconditioner. To be more specific, NKS can be described as follows. Let $x^{(0)}$ be a given initial guess. Assume $x^{(k)}$ is the current approximation of x^*. Then a new approximation $x^{(k+1)}$ can be computed by first finding a Newton direction $s^{(k)}$ by solving the following preconditioned Jacobian system approximately,

$$J(x^{(k)}) M_k^{-1} y = -F(x^{(k)}), \text{ with } s^{(k)} = M_k^{-1} y, \tag{2.5}$$

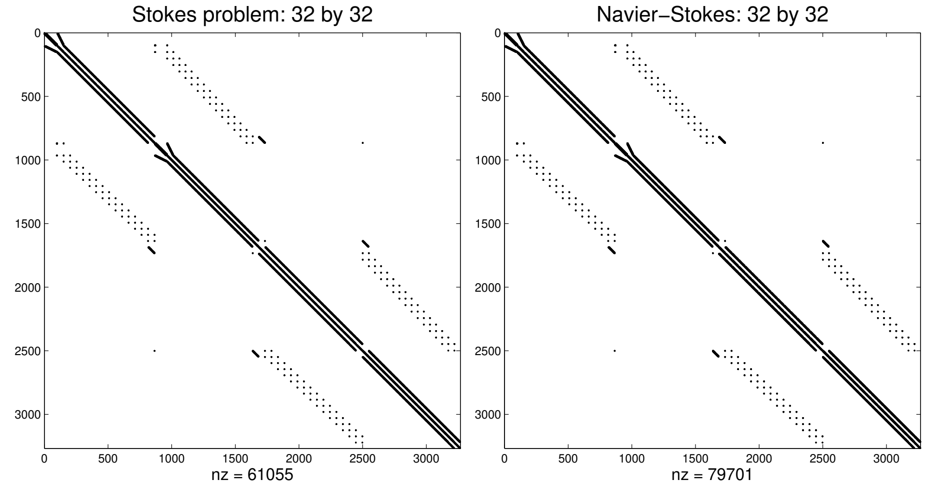

FIG. 2.1. *Nonzero pattern of the discrete Stokes matrix (left) and the Jacobian matrix (right) for Re* $= 1,000$ *on a 32* $\times$ *32 mesh. nz is the number of nonzero elements.*

then obtaining the new approximation via $x^{(k+1)} = x^{(k)} + \lambda^{(k)} s^{(k)}$, where $J(x)$ is the Jacobian of F evaluated at $x^{(k)}$ and $\lambda^{(k)} \in (0, 1]$ is a damping parameter. In our case, the nonlinear system, $F(x) = 0$, is obtained by applying $Q_1 - Q_1$ Galerkin least square (GLS) finite element discretization [37] on the mesh $\mathcal{T}^h$ and the corresponding Jacobian J is constructed by using a multi-colored forward finite difference method [12]. Figure 2.1 is an example showing that the discrete Stokes problem in (2.3) and the Jacobian matrix in (2.5) have similar nonzero structures. The matrix for the discrete Stokes problem has about 3/4 the number of nonzero elements as in the Jacobian matrix. The additional nonzero elements in the Jacobian matrix come from mainly the derivatives of the convection and stabilization terms in GLS. For incompressible Navier-Stokes equations, the Reynolds number (Re) plays an important role in the behavior of the numerical solution and the convergence of iterative methods. Several numerical difficulties arise due to high Re. It is well known that the convergence radius for a Newton-type method becomes smaller as Re increases. Newton-type methods often fail to converge when a good initial guess is unavailable. In this paper we will not address the issues related to the robustness of Newton-type methods; interested readers may consult [8, 9, 22, 40]. Instead, we restrict our study to the parallel performance of Schwarz preconditioned Krylov subspace methods for solving the Jacobian systems. Note that the Jacobian systems become more nonsymmetric and ill-conditioned as Re increases.

3. Fully coupled Schwarz methods.

In this section, we introduce several Schwarz preconditioners for the Stokes and Jacobian problems.

3.1. One-level additive Schwarz preconditioner.

To define parallel Schwarz type preconditioners we need to partition the finite element mesh $\mathcal{T}^h$ introduced in the previous section. Let $\{\Omega_i^h, i = 1, ..., N\}$ be a non-overlapping subdomain partition whose union covers the entire domain Ω and its mesh $\mathcal{T}^h$. We use $\mathcal{T}_i^h$ to denote the collection of mesh points in Ω_i^h. To obtain overlapping subdomains, we expand each subdomain Ω_i^h to a larger subdomain $\Omega_i^{h,\delta}$ with the boundary $\partial\Omega_i^{h,\delta}$. Here δ is an integer indicating the level of overlap. We assume that neither Ω_i^h nor $\partial\Omega_i^{h,\delta}$ cut any elements of $\mathcal{T}^h$. Similarly, we use $\mathcal{T}_i^{h,\delta}$ to denote the collection of mesh points in $\Omega_i^{h,\delta}$.

ETNA
Kent State University
etna@mcs.kent.edu

Now, we define the subdomain velocity space as

$$V_i^h = \{\boldsymbol{v}^h \in V^h \cap \left(H^1(\Omega_i^{h,\delta})\right)^2 : \boldsymbol{v}^h = 0 \text{ on } \partial\Omega_i^{h,\delta}\}$$

and the subdomain pressure space as

$$P_i^h = \{p^h \in P^h \cap L^2(\Omega_i^{h,\delta}) : p^h = 0 \text{ on } \partial\Omega_i^{h,\delta}\backslash\Gamma\}.$$

Both are subspaces of V^h and P^h, respectively, if all subdomain functions are extended to the whole domain by zero. Note that for Q_1-Q_1 elements, each interior node has three degrees of freedom, two for the velocity and one for the pressure. On the physical boundaries, we impose Dirichlet conditions according to the original equations (2.1). On the artificial boundaries, we assume both $\boldsymbol{u} = 0$ and $p = 0$. Similar boundary conditions are used in [29]. Although it is still not certain that these boundary conditions are mathematically correct, they work well in practice. The subdomain problem reads as follows: Find $\boldsymbol{u}_i^h \in V_i^h$ and $p_i^h \in P_i^h$, such that

$$B(\boldsymbol{u}_i^h, p_i^h; \boldsymbol{v}, q) = F(\boldsymbol{v}, q) \qquad \forall(\boldsymbol{v}, q) \in V_i^h \times P_i^h, \tag{3.1}$$

which takes the matrix form

$$A_i x_i = b_i.$$

Note that the right-hand side of (3.1) is not important, since we only use the left-hand side matrix A_i to define the subdomain preconditioner. The right-hand side is not used at all in the computation.

Let $R_i : V^h \times P^h \to V_i^h \times P_i^h$ be a restriction operator, which returns all degrees of freedom (both velocity and pressure) associated with the subspace $V_i^h \times P_i^h$. R_i is an $(3n_i - 2d_i)\times(3n - 2d)$ matrix with values of either 0 or 1, where n and n_i are the total number of mesh points in $\mathcal{T}^h$ and $\mathcal{T}_i^{h,\delta}$, respectively. Similarly, r and r_i are total number of mesh points imposed the Dirichlet boundary conditions for velocity in $\mathcal{T}^h$ and $\mathcal{T}_i^{h,\delta}$, respectively. Since our subdomain partition is element-based, $\sum_{i=1}^N(3n_i - 2r_i) \geq 3n - 2r$. Then, the interpolation operator R_i^T can be defined as the transpose of R_i. The multiplication of R_i (and R_i^T) with a vector does not involve any arithmetic operations, but does involve communication in a distributed memory implementation. Using the restriction operator, we write the one-level additive Schwarz preconditioner (**ASM1**) in the matrix form as

$$P_{ASM1}^{-1} = \sum_{i=1}^{N} R_i^T A_i^{-1} R_i,$$

where A_i^{-1} is the subspace inverse of A_i. In practice, it is more convenient to obtain A_i from $A_i = R_i A R_i^T$. We remark that the global-to-local restriction operator R_i collects the data from neighboring subdomains, while the local-to-global prolongation operator R_i^T sends partial solutions to neighboring subdomains.

3.2. Two-level methods with a parallel coarse preconditioner. The direct coarse preconditioner proposed in [28] provides good mathematical properties, but is not easy to parallelize. Here we propose a parallel coarse solver, which is the same as the one-level additive Schwarz preconditioned Krylov subspace method discussed in the previous subsection, except that it is constructed on a partitioned coarse mesh. More precisely, we assume there exists a finite element mesh $\mathcal{T}^H$ covering the domain Ω. The two meshes $\mathcal{T}^H$ and $\mathcal{T}^h$ do not have to be nested. The coarse mesh problem will help speed up the convergence of the iterative method, but has nothing to do with the accuracy of the discretization, which is determined by the fine mesh $\mathcal{T}^h$ only.

For the purpose of parallel computing, the coarse mesh is partitioned to non-overlapping subdomains $\{\Omega_i^H\}$ and overlapping subdomains $\{\Omega_i^{H,\delta}\}$. The corresponding sets of mesh points are denoted by $\{\mathcal{T}_i^H\}$, and $\{\mathcal{T}_i^{H,\delta}\}$. For the simplicity of our software implementation, we assume that we have a *nested* non-overlapping partition. In other words, we have

$$\Omega_i^h = \Omega_i^H$$

for $i = 1, \ldots, N$, even though the corresponding sets of mesh points do not have to be nested, i.e.

$$\mathcal{T}_i^H \not\subseteq \mathcal{T}_i^h.$$

This also means that the same number of processors is used for both the fine and coarse mesh problems. If the overlap is taken into account, in general,

$$\Omega_i^{h,\delta} \neq \Omega_i^{H,\delta} \quad \text{and} \quad \mathcal{T}_i^{H,\delta} \not\subseteq \mathcal{T}_i^{h,\delta}.$$

As in the fine mesh case, we can also define the restriction and extension operators R_i^c and $(R_i^c)^T$ for each coarse subdomain. On the coarse mesh $\mathcal{T}^H$, we can define finite element subspaces similar to the ones defined on the fine mesh, and discretize the original Stokes problem to obtain a linear system of equations

$$A^c x^c = b^c.$$

Note that the vectors b^c and x^c are not used in the computation; only the matrix A^c is used to define our coarse preconditioner. Following a similar argument, on the coarse subdomains, we obtain the coarse submatrices

$$A_i^c, i = 1, \ldots, N.$$

As opposed to strongly elliptic problems, our experiments show that the coarse mesh size for indefinite problems needs to be sufficiently fine to retain the scalability of the algorithm. (see also [11]). Hence, the parallelization of a coarse solver is necessary, and, in general, there are three strategies: (1) A direct exact approach, which performs an LU decomposition of A^c in parallel and then does the forward/backward substitutions using some parallel sparse direct solvers, such as SuperLU_DIST [13]. (2) A direct inexact approach, which replaces $(A^c)^{-1}$ by $\sum_{i=1}^N (R_i^c)^T (B_i^c)^{-1} R_i^c$, where B_i^c could be an ILU(k) decomposition of A_i^c. (3) An iterative approach, which solves the coarse mesh problem, $A^c z^c = w^c$, by some parallel iterative methods, such as parallel GMRES, with a coarse additive Schwarz preconditioner.

Aitbayev *et al.* [1] and Tuminaro *et al.* [44] have studied the performance of some direct exact coarse solvers based on the first approach and inexact domain decomposition coarse solvers based on the second approach. However, they are either too expensive or too inexact to be effective. Instead, in this paper we adopt the third approach, which is something in between the first and the second approaches, and the solution accuracy of the coarse mesh problem can be controlled easily. Now we discuss the parallel iterative coarse solver and the operators that transfer the data between the fine and coarse meshes. We define the coarse preconditioner $(M^c)^{-1}$ in terms of a matrix-vector multiply: For any given coarse mesh vector w^c,

$$z^c = (M^c)^{-1} w^c$$

ETNA
Kent State University
etna@mcs.kent.edu

is understood as an approximate solution of the following preconditioned linear system of equations

$$A^c \left[\sum_{i=1}^{N} (R_i^c)^T (A_i^c)^{-1} R_i^c \right] y^c = w^c, \text{ with } z^c = \left[\sum_{i=1}^{N} (R_i^c)^T (A_i^c)^{-1} R_i^c \right] y^c, \tag{3.2}$$

which satisfies the condition

$$\|w^c - A^c z^c\|_2 \le \epsilon_C \|w^c\|_2 .$$

Note that for any given w^c, the computation of z^c can be carried out in parallel using all N processors. M^c is not exactly a matrix, but a preconditioned iterative solver.

We define the coarse-to-fine and fine-to-coarse mesh transfer operators as follows. Let $\{\phi_j^H(x), j = 1, \ldots, m\}$ be the finite element basis functions on the coarse mesh, where m is the total number of coarse mesh points in $\mathcal{T}^H$. Let s be the total number of coarse mesh points at which the Dirichlet boundary condition for velocity in $\mathcal{T}^H$ is imposed. We define an $(3n - 2r)\times(3m - 2s)$ matrix I_H^h, the coarse-to-fine extension operator, as

$$I_H^h = [E_1 E_2 \cdots E_n]^T ,$$

where the block matrix E_i of size $3 \times 3m$ is given by

$$E_i = \begin{bmatrix} (e_H^h)_i & 0 & 0 \\ 0 & (e_H^h)_i & 0 \\ 0 & 0 & (e_H^h)_i \end{bmatrix}$$

and the row vector $(e_H^h)_i$ of length m is given by

$$(e_H^h)_i = \left[\phi_1^H(x_i), \phi_2^H(x_i), \ldots \phi_m^H(x_i) \right] , \; x_i \in \mathcal{T}^h$$

for $i = 1, \ldots, n$. A global fine-to-coarse restriction operator I_h^H can be defined as the transpose of I_H^h.

Using the coarse preconditioner defined above, we define a two-level additive Schwarz preconditioner (**ASM2**)

$$P_{ASM2}^{-1} = I_H^h (M^c)^{-1} I_h^H + \sum_{i=1}^{N} R_i^T A_i^{-1} R_i .$$

Because the partitioned coarse mesh problem and the partitioned fine mesh problem are allocated to the same processor, the coarse and fine subproblems cannot be solved at the same time. Therefore it is reasonable to consider a multiplicative approach for the coarse mesh problem in order to obtain a faster convergence. Following [31], we define the following hybrid Schwarz method (**HSM**),

$$P_{HSM}^{-1} = I_H^h (M^c)^{-1} I_h^H + \left(I - I_H^h (M^c)^{-1} I_h^H A \right) \left(\sum_{i=1}^{N} R_i^T A_i^{-1} R_i \right) ,$$

which is additive among all fine mesh subdomains, and multiplicative between the fine and coarse preconditioners.

We remark that once the overlapping subdomains and the mesh transfer operators are defined on both coarse and fine meshes, the three parallel overlapping Schwarz preconditioners just described are easily adapted for solving nonsymmetric Jacobian systems. The only issue now is, how to define, for each Newton iteration, the subdomain problems on both fine and coarse meshes and the coarse mesh problem. We define the matrices associated with the fine mesh subdomain problems and coarse mesh problem in a simple way. For the subdomain problems, we define J_i by $J_i = R_i J(x^{(k)}) R_i^T$. For the coarse problem, on the coarse mesh $\mathcal{T}^H$, we first discretize the original Navier-Stokes equations to obtain a nonlinear system of equations, $F^c(x^c) = 0$. Then the corresponding coarse Jacobian $J^c(x)$ is computed using multi-colored finite differences. Hence at the i-th Newton iteration, the coarse mesh matrix takes the form $J^c(\widehat{I_H^h} x^{(k)})$, where $\widehat{I_H^h}$ is the standard injection operator, which restricts the solution from the fine mesh to the coarse mesh. Similar to the fine mesh, on the coarse subdomains, we define the coarse Jacobian submatrices as $J_i^c = (R_i^c) J^c(\widehat{I_H^h} x^{(k)}) (R_i^c)^T$.

4. Numerical experiments. In this section, we consider a two-dimensional lid-driven cavity flow problem as a benchmark for evaluating the parallel performance of the three preconditioners introduced in the previous section. A detailed description of the lid-driven cavity problem can be found in [19]. Several parameters in the Schwarz methods need to be specified for achieving optimal performance, in this paper, we focus on the impact of the following parameters on the convergence rate and the overall execution time: (1) the subdomain overlapping size; (2) the subdomain and coarse mesh solution quality; and (3) the coarse mesh size. We also investigate the parallel scalability, which shows how the algorithm behaves as the size of fine mesh and the number of processors grow. The Portable, Extensible Toolkit for Scientific computation (PETSc) [2] is used for the parallel implementation and all numerical results are obtained on a cluster of PCs running Linux. All timings are reported in seconds, and the execution time excludes the time spent on preprocessing steps including the setup of the problem matrix, the construction of the extension matrix, and the decomposition of the meshes. The matrix decomposition is included in the timing. The parameters and other details of the numerical experiments for the Stokes problem are summarized below:

- For the one-level Schwarz method, restarted GMRES(100) is employed for solving the preconditioned linear system (2.4). For the two-level Schwarz methods, including **ASM2** and **HSM**, Flexible GMRES (FGMRES) [38] is applied for solving the preconditioned system while restarted GMRES(100) is used for the coarse mesh problem (3.2). It should be noted that according to our numerical experiences, when standard GMRES was used instead of FGMRES for the two-level Schwarz methods, there was no convergence problem. However, we did observe some loss of solution accuracy if the coarse mesh problem is not solved accurately enough under the same outer stopping criterion. So due to the changes of two-level Schwarz preconditioners with the inexact coarse solver at each of the outer iterations, FGMRES is more suitable than the standard GMRES. We use a zero initial guess for both problems. The stopping criterion for (2.4) is that the condition
$$\left\| b - (AM_k^{-1})(M_k x^{(k)}) \right\|_2 \le \epsilon_F \, \|b\|_2$$
is satisfied and we set $\epsilon_F = 10^{-5}$ for all test cases. Similarly, the stopping criterion for the coarse mesh problem (3.2) is
$$\left\| w^c - A^c z^{(k)} \right\|_2 \le \epsilon_C \, \|w^c\|_2 \, .$$
Several values of ϵ_C are tested, ranging from 10^{-1} to 10^{-10}.
- Three uniform checkerboard subdomain partitions are used for our experiments: 2×2, 4×4, and 8×8. The number of subdomains is always the same as the number of processors, n_p.

ETNA
Kent State University
etna@mcs.kent.edu

- Three fine meshes are considered: 128×128, 256×256, and 512×512 and the number of unknowns ranges from 50,000 to 780,000. The coarse mesh is varied from 16×16 to 80×80. Since a non-nested coarse mesh is used, the number of processors and the coarse mesh size are not related.
- The overlapping size for the fine mesh is defined as

$$ovlp = \max\left\{\frac{L'_x - L_x}{2h_K}, \frac{L'_y - L_y}{2h_K}\right\}$$

  for both interior subdomains and those touching the boundary. Since square elements are used for the test problem, the elemental diameter h^Ks are the same and equal to the fine mesh size. L'_x and L'_y are defined here as the side lengths of the overlapping subdomain $\Omega_i^{h,\delta}$ in the x-direction and the y-direction, respectively. Similarly, L_x and L_y are defined as the side lengths of the non-overlapping subdomain Ω_i^h in the x-direction and the y-direction, respectively. We defined the overlapping size $ovlp^H$ for the coarse mesh problem in the same fashion as above, and used the value $ovlp^H = 1$ for all test cases.
- Either a direct sparse solver or an inexact LU(k) solver, with level of fill-in $k = 0, 1, 2, \ldots, 5$, is employed to solve the subdomain problems.

4.1. The effect of the coarse mesh size and inexact coarse solvers. In Table 4.1, we examine the effect of the coarse mesh size on the convergence and execution time of the two-level Schwarz methods. In this set of numerical experiments, we keep the subdomain size fixed and scale up $\sqrt{n_p}$ and the fine mesh size h by a factor of 64. All subdomain problems are solved by an exact sparse LU decomposition and $ovlp = 1$. We vary the coarse mesh size from 16×16 to 80×80. The tolerance for the coarse iterative solver is $\epsilon_C = 10^{-10}$. The empty entry $(-)$ in the table indicates that a uniform partitioning for such coarse mesh size is not available. In each row of Table 4.1, for both two-level Schwarz methods, the number of FGMRES iterations is reduced monotonically when the coarse mesh size is increased. However, the smallest number of iterations does not imply the fastest convergence. The two-level methods with a 80×80 coarse problem are always slower than with a 16×16 coarse problem, although they only take fewer iterations. This is because solving the larger coarse mesh problem takes a more significant portion of the total time. The optimal coarse mesh size for both methods, based on the optimal execution time, depends on the fine mesh size. Roughly $H \sim 10h$ is needed to achieve the fastest convergence.

Next, we relax the requirement of solution accuracy for the coarse problem, and vary ϵ_C from 10^{-1} to 10^{-10}. The ratio of the coarse to fine mesh sizes, H/h, is set to be 8. As shown in Table 4.2, the accuracy of the coarse mesh solution does not need to be very high in order to retain the optimal convergence rate of the two-level Schwarz preconditioners. $\epsilon_C = 10^{-2}$ is sufficient in both cases. Surprisingly, previous work has shown that this is not true for other types of problems such as indefinite elliptic problems [11]. In [11], a theory for **ASM2** requires an exact solver on the coarse mesh. Furthermore, the inexact coarse solver saves a significant amount of time in both two-level Schwarz methods, especially when the number of processors is large. In the 64-processor case, the inexact coarse solver with $\epsilon_C = 10^{-2}$ takes only about half of the time needed for the almost exact solver with $\epsilon_C = 10^{-10}$.

4.2. The effect of inexact subdomain solvers. We investigate the effect of using ILU(k) as inexact subdomain solvers. In Table 4.3 we report computational results with varying levels of fill-in in ILU(k) for a fixed 64×64 subdomain problem and increasing the number of processors and the fine mesh size. We use a fixed coarse to fine mesh size ratio of $H/h = 8$ for the two-level Schwarz preconditioners. The tolerance for the iterative coarse

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 4.1

Varying the coarse mesh size. Fixed subdomain mesh size 64×64*. All subdomain problems are solved by LU,* $ovlp = 1$*. The tolerance* ϵ_C *for the coarse mesh problem is set to be* 10^{-10}.

Fine mesh (n_p)	Coarse mesh	16×16	20×20	32×32	40×40	64×64	80×80
				ASM2			
128×128 (4)	FGMRES	25	23	19	17	15	13
	Time (sec)	***2.3***	2.4	2.8	3.5	6.4	8.9
256×256 (16)	FGMRES	33	30	25	22	19	18
	Time (sec)	3.0	***2.9***	2.9	3.1	4.7	6.9
512×512 (64)	FGMRES	59	–	40	35	27	25
	Time (sec)	6.5	–	5.4	***5.2***	6.0	7.8
				HSM			
128×128 (4)	FGMRES	18	17	14	12	10	8
	Time (sec)	***1.9***	2.1	2.5	3.0	5.0	6.3
256×256 (16)	FGMRES	24	21	17	15	11	10
	Time (sec)	2.5	2.5	***2.4***	2.5	3.3	4.6
512×512 (64)	FGMRES	39	–	27	24	18	16
	Time (sec)	4.7	–	***4.0***	***4.0***	4.4	5.4

TABLE 4.2

Varying the tolerance ϵ_C *for the coarse mesh problem. Fixed subdomain problem size:* 64×64*. The ratio of coarse to fine mesh sizes:* $H/h = 8$*. All subdomain problems are solved by exact LU,* $ovlp = 1$.

Fine mesh (n_p)	ϵ_C	10^{-1}	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}
				ASM2			
128×128 (4)	FGMRES	19	19	25	25	25	25
	Time (sec)	***1.7***	***1.7***	2.1	2.1	2.2	2.3
256×256 (16)	FGMRES	25	25	25	25	25	25
	Time (sec)	***2.1***	2.2	2.6	2.7	2.7	2.9
512×512 (64)	FGMRES	28	27	27	27	27	27
	Time (sec)	***2.6***	2.8	4.0	5.0	5.5	6.0
				HSM			
128×128 (4)	FGMRES	14	14	18	18	18	18
	Time (sec)	***1.5***	1.6	1.8	1.9	1.9	2.0
256×256 (16)	FGMRES	19	17	17	17	17	17
	Time (sec)	1.9	***1.8***	2.1	2.2	2.3	2.4
512×512 (64)	FGMRES	29	19	18	18	18	18
	Time (sec)	2.6	***2.2***	2.9	3.5	4.0	4.4

solver is set to be $\epsilon_C = 10^{-2}$ and a fixed $ovlp = 1$ is used. We vary the level of fill-ins, k, from 0 to 5. As shown in Table 4.3, we observe that the number of iterations for **ASM1** is more sensitive to the change of k in ILU(k) and the number of processors (or the fine mesh size) than **ASM2** and **HSM**: the convergence rate of **ASM1** can be improved dramatically by increasing k, especially when the number of processors is large. On the other hand, for **ASM2** and **HSM** with ILU(k), k from 0 to 3, the numbers of iterations are nearly independent of the numbers of processors. About the timing, we observe that **ASM1** with ILU(k) is faster than **ASM1** with LU only in the 4-processor case. **ASM1** with ILU(k) becomes less efficient as the number of processors increases due to the exceedingly large number of iterations required for convergence. For **ASM2** and **HSM**, the results with optimal execution time are obtained by using $k = 0$ and $k = 1$, respectively, and the time savings are remarkable for both cases compared to the cases using LU as a subdomain solver: ILU(k) is more than 55% faster than LU.

4.3. The effect of the overlapping size. Table 4.4 shows the effect of the overlapping size for the two-level Schwarz methods with 64 processors. The overlapping size varies from 1 to 16. A fixed 512×512 fine mesh and a fixed 64×64 coarse mesh are used. In

TABLE 4.3
Varying the level of ILU(k) fill-in. Fixed subdomain mesh size: 64×64. The ratio of coarse to fine mesh sizes is fixed at $H/h = 8$. The tolerance ϵ_C for the iterative coarse solver is set to be 10^{-2}, $ovlp = 1$.

Fine mesh (n_p)	ILU(k)	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	LU
				ASM1				
128×128 (4)	GMRES	190	98	67	51	44	40	22
	Time (sec)	8.4	3.4	2.2	1.8	***1.6***	1.7	1.7
256×256 (16)	GMRES	743	328	163	124	103	90	54
	Time (sec)	37.9	16.5	7.9	5.5	5.0	4.8	***3.3***
512×512 (64)	GMRES	4151	1980	962	628	490	410	168
	Time (sec)	137.8	69.8	35.5	24.6	20.3	***18.3***	9.3
				ASM2				
128×128 (4)	FGMRES	27	23	23	22	23	30	19
	Time (sec)	***0.7***	0.8	0.8	0.9	1.1	1.5	1.7
256×256 (16)	FGMRES	27	23	23	23	22	22	25
	Time (sec)	***0.9***	***0.9***	1.2	1.1	1.5	1.7	2.2
512×512 (64)	FGMRES	26	23	23	23	23	24	27
	Time (sec)	***1.2***	1.3	1.5	1.6	1.6	2.1	2.8
				HSM				
128×128 (4)	FGMRES	19	14	14	15	21	25	14
	Time (sec)	0.7	***0.6***	0.7	0.8	1.1	1.4	1.6
256×256 (16)	FGMRES	19	14	14	15	16	20	17
	Time (sec)	0.7	***0.7***	***0.7***	0.9	1.2	1.3	1.8
512×512 (64)	FGMRES	19	14	14	15	16	20	25
	Time (sec)	1.0	***0.9***	1.1	1.1	1.3	1.9	2.2

TABLE 4.4
Varying the overlapping size ovlp. Fine mesh size: 512×512. Coarse mesh size: 64×64. $n_p = 64$.

	$ovlp$	1	2	4	8	16
LU subdomain solver, almost exact coarse solver, $\epsilon_C = 10^{-10}$						
ASM2	FGMRES	27	25	21	17	17
	Time (sec)	6.3	6.0	5.4	***5.2***	6.9
HSM	FGMRES	18	16	12	13	13
	Time (sec)	4.4	3.9	***3.7***	4.5	6.1
ILU(1) subdomain solver, inexact coarse solver, $\epsilon_C = 10^{-2}$						
ASM2	FGMRES	23	22	23	26	31
	Time (sec)	***1.3***	***1.3***	1.4	1.8	2.7
HSM	FGMRES	14	14	15	19	22
	Time (sec)	***0.9***	1.0	1.1	1.4	2.1

the table, we observe that the behavior of these two-level Schwarz methods depends on the subdomain and coarse mesh solution quality. Similar to elliptic-type PDEs, when using exact LU subdomain solve and almost exact coarse solve ($\epsilon_C = 10^{-10}$), the number of iterations is reduced monotonically as we increase $ovlp$, except in the case of **HSM** with $ovlp = 8$ and 16. However, the saving in computing time is limited. With the optimal $ovlp$, the total time taken to solve the problem is about 17% less than the time with the minimal $ovlp$. On the other hand, if we relax the quality of solution for both subdomain and coarse problems, the convergence rates of **ASM2** and **HSM** degenerate with $ovlp$ increasing, and the optimal execution times are obtained by the use of the small $ovlp$. We do not fully understand why the number of iterations increases when $ovlp$ becomes larger, although we have observed that this also happens for some indefinite problems, e.g. the Helmholtz problem [10], and we believe the indefiniteness of the problem plays a role here.

4.4. Parallel performance study. Scalability is an important issue in parallel computing, and the issue is more significant when solving large-scale problems with many pro-

ETNA
Kent State University
etna@mcs.kent.edu

cessors. To evaluate the parallel scalability of the three Schwarz methods, we adopt the fixed-subdomain-mesh-per-processor scalability as a measurement. An algorithm is considered to be scalable if the computing time remains constant provided that the fine mesh and the number of processors both increase at the same rate, while the subdomain size is fixed. The scalability study of **ASM1**, **ASM2**, and **HSM** is summarized in Table 4.5. Note that the data in the table, including the iteration numbers and execution time for **ASM1** except for the single-processor case, is excerpted from the last column of Table 4.3 and for **ASM2** and **HSM** with **ASM1** as a coarse solver from the first column and second column of Table 4.3, respectively. For the purpose of comparison, we also include the numerical results for two-level Schwarz methods with two different coarse solvers, namely a redundant LU approach (RLU), which redundantly solves the same coarse mesh problem on all processors and a one-step Schwarz-Richardson method (SR), which is equivalent to the second approach mentioned in Section 3.2. For the case of single processor, the numerical result is obtained by using ILU(1) in conjunction with GMRES. The scaled efficiency shown in Table 4.5 is defined by $\eta = T_1/T_{n_p}$,where T_1 and T_{n_p} are the computing time obtained with 1 and n_p processors. In the ideal case, $\eta \sim 1$.

From Table 4.5, we observe that **ASM1** is not scalable; the number of GMRES iterations grows, roughly, as $\sqrt{n_p}$ and only 6% scaled efficiency is achieved. Comparing the two-level Schwarz methods with different coarse solvers, we find that neither RLU nor SR is as efficient as **ASM1**. Although the cost per FGMRES for SR is quite low and scalable (see the last three rows of Table 4.6), the solution of SR is not accurate enough to retain the scalability of the method. Also, as shown in Table 4.6, for the small coarse mesh problem, RLU is preferable to **ASM1**, but as the coarse problem size increases, **ASM1** outperforms RLU. **ASM1** takes only 10% of the computing time needed for RLU in the case of 64 processors. Next, we compare **ASM2** and **HSM** using **ASM1** as the coarse solver. Both **ASM2** and **HSM** are scalable in terms of FGMRES iterations and **HSM** requires only about half the FGMRES iterations needed for **ASM2**. Furthermore, **HSM** is always $20 - 25\%$ faster than ASM and the scaled efficiency of **HSM** maintains at at least 67% for all cases. While the number of FGMRES iterations is unchanged, the degradation of the scaled efficiency is mainly due to the non-scalable cost per FGMRES iteration. From the first three rows of Table 4.6, we see that the computing time increases as the number of processors grows.

4.5. Schwarz preconditioners for the nonsymmetric saddle point problem. In this section, we study the performance of **HSM** for the nonsymmetric indefinite Jacobian systems in the NKS algorithm for incompressible Navier-Stokes equations. Table 4.7 shows the average number of FGMRES iterations and the execution time for **HSM** during a Newton iteration for the two-dimensional lid-driven cavity problem on 64 processors. The stopping criterion for (2.5) is:

$$\|F(x^{(k)}) + (J_k(x^{(k)})M_k^{-1})(M_k s^{(k)})\|_2 \leq 10^{-4}\|F(x^{(k)})\|_2.$$

The numbers in parentheses are Newton iteration counts required to reduce the initial nonlinear residual by a factor of 10^{-6}, that is, $\|F(x^{(k)})\|_2 \leq 10^{-6}\|F(x^{(0)})\|_2$. The Reynolds number Re is varied from 1 to 1,000. A fixed fine mesh 512×512 and a fixed coarse mesh 64×64 are employed. Some key observations from Table 4.7 are made as follows:

(1) The number of FGMRES iterations becomes exceedingly large when ILU(k) is used as the subdomain solve. The only exception is the case of ILU(0). This phenomenon is different from the case of symmetric Stokes problems. We suspect that ILU(k) is not stable for the nonsymmetric indefinite saddle point problem, and therefore do not recommend it.

(2) **HSM** with ILU(0) is competitive with **HSM** with LU only for small Re. As Re increases, **HSM** with LU is found to be $20 - 40\%$ faster than **HSM** with ILU(0).

TABLE 4.5
Fixed-subdomain-size-per-processor scalability.

Fine mesh (n_p)	FGMRES iterations	Time (sec)	Scaled efficiency η
(1) ASM1, LU as subdomain solver			
64×64 (1)	36	0.6	100%
128×128 (4)	22	1.7	35%
256×256 (16)	54	3.3	18%
512×512 (64)	168	9.3	6%
(2) ASM2, ILU(0) as subdomain solver			
(a) Coarse solver: **ASM1**			
64×64 (1)	36	0.6	100%
128×128 (4)	27	0.8	75%
256×256 (16)	27	0.9	67%
512×512 (64)	26	1.2	50%
(b) Coarse solver: Redundant LU (RLU)			
64×64 (1)	36	0.6	100%
128×128 (4)	27	0.7	86%
256×256 (16)	27	1.1	55%
512×512 (64)	26	4.1	15%
(c) Coarse solver: One-Step Schwarz-Richardson (SR)			
64×64 (1)	36	0.6	100%
128×128 (4)	35	1.1	55%
256×256 (16)	44	1.3	46%
512×512 (64)	86	3.2	18%
(3) HSM, ILU(1) as subdomain solve			
(a) Coarse solver: **ASM1**			
64×64 (1)	36	0.6	100%
128×128 (4)	14	0.6	100%
256×256 (16)	14	0.7	86%
512×512 (64)	14	0.9	67%
(b) Coarse solver: Redundant LU (RLU)			
64×64 (1)	36	0.6	100%
128×128 (4)	14	0.6	100%
256×256 (16)	14	0.9	66%
512×512 (64)	14	3.5	17%
(c) Coarse solver: One-Step Schwarz-Richardson (SR)			
64×64 (1)	36	0.6	100%
128×128 (4)	22	0.8	75%
256×256 (16)	40	1.5	40%
512×512 (64)	82	3.7	16%

TABLE 4.6
Computing time per FGMRES iteration for the iterative and direct coarse solvers.

Coarse Mesh Size	n_p	Ave. Inner iterations	Time
ASM1			
16×16	4	5	4×10^{-3}
32×32	16	7	6×10^{-3}
64×64	64	12	2×10^{-2}
Redundant LU			
16×16	4	1	2×10^{-3}
32×32	16	1	2×10^{-2}
64×64	64	1	2×10^{-1}
One-step Schwarz-Richardson			
16×16	4	1	6×10^{-4}
32×32	16	1	7×10^{-4}
64×64	64	1	7×10^{-3}

TABLE 4.7

Average number of FGMRES iterations and computing time. HSM for the Jacobian system in the NKS algorithm for Navier-Stokes equations. Reynolds number Re is varied from 1 to 1,000. Fine mesh: 512×512 and coarse mesh: 64×64. The tolerance ϵ_C for the coarse mesh problem is set to be 10^{-2}, $ovlp = 1$, and $n_p = 64$.

Re	ILU(k)	$k = 0$	$k = 1$	$k = 2$	$k = 3$	LU
1.0	FGMRES	30.5 (2)	126.0 (2)	701.9 (2)	163.0 (2)	20.0 (2)
	Time (sec)	***2.4***	11.1	76.9	14.2	3.1
100.0	FGMRES	36.5 (4)	143.5 (4)	827.0 (4)	174.5 (4)	24.8 (4)
	Time (sec)	***3.6***	15.2	82.4	16.4	3.6
500.0	FGMRES	48.0 (7)	199.4 (7)	558.9 (7)	289.4 (7)	32.1 (7)
	Time (sec)	6.1	60.3	64.9	34.4	***5.6***
1000.0	FGMRES	66.2 (11)	279.2 (11)	354.5 (11)	319.5 (11)	38.9 (11)
	Time (sec)	9.8	34.4	42.0	38.9	***7.0***

(3) The number of FGMRES iterations for **HSM** with LU depends slightly on Re. The higher the Reynolds number is, the more linear and nonlinear iterations it takes to meet the stopping requirement.

5. Concluding remarks. We presented a fully coupled parallel solver with three types of overlapping Schwarz preconditioners for saddle point problems arising from CFD. As shown in the numerical results, additive Schwarz-type preconditioners for the Stokes problem and the elliptic-type problems have some similar convergence properties: the number of GMRES iterations decreases monotonically as the overlapping size increases when LU is used as the subdomain solve; the number of GMRES iterations for **ASM1** grows as $\sqrt{n_p}$, and **HSM** requires roughly half as many FGMRES iterations as **ASM2**. Our numerical results also showed that the ILU(k) based inexact subdomain solver is useful only for the symmetric Stokes problem but not for the nonsymmetric indefinite Jacobian systems in NKS due to the possible instability of the incomplete decomposition. The iterative inexact coarse solver is sufficient to retain the optimal convergence rate of the two-level Schwarz preconditioners. In comparison with **ASM1** and **ASM2**, **HSM** with a iterative coarse solver showed superior parallel performance.

REFERENCES

[1] R. AITBAYEV, X.-C. CAI, AND M. PARASCHIVOIU, *Parallel two-level methods for three-dimensional transonic compressible flow simulations on unstructured meshes*, in Parallel Computational Fluid Dynamics: Towards Teraflops, Optimization and Novel Formulations, D. E. Keyes et al., eds., Elsevier, Amsterdam, 2000, pp. 89–96.

[2] S. BALAY, K. BUSCHELMAN, W. D. GROPP, D. KAUSHIK, M. KNEPLEY, L. C. MCINNES, B. F. SMITH, AND H. ZHANG, *PETSc Users Manual, ANL-95/11 - Revision 2.1.5*, Argonne National Laboratory, 2004.

[3] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1–137.

[4] G. BIROS AND O. GHATTAS, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization, Part I: the Krylov-Schur solver*, SIAM J. Sci. Comput., 27 (2005), pp. 687-713.

[5] A. BRANDT, *Multigrid techniques: 1984 guide with applications to fluid dynamics*, GMD-studien Nr., 85 Bonn, 1984.

[6] J. H. BRAMBLE AND J. E. PASCIAK, *A domain decomposition technique for Stokes problems*, Appl. Numer. Math., 6 (1990), pp. 251–261.

[7] J. H. BRAMBLE, J. E. PASCIAK, AND A. T. VASSILEV, *Analysis of the inexact Uzawa algorithm for saddle point problems*, SIAM J. Numer. Anal., 34 (1997), pp. 1072–1092.

[8] X.-C. CAI AND D. E. KEYES, *Nonlinearly preconditioned inexact Newton algorithms*, SIAM J. Sci. Comput., 24 (2002), pp. 183–200.

[9] X.-C. CAI, D. E. KEYES, AND L. MARCINKOWSKI, *Nonlinear additive Schwarz preconditioners and applications in computational fluid dynamics*, Int. J. Numer. Meth. Fluids, 40 (2002), pp. 1463–1470.

[10] X.-C. CAI AND M. SARKIS, *A restricted additive Schwarz preconditioner for general sparse linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 792-797.

[11] X.-C. CAI AND O. WIDLUND, *Domain decomposition algorithms for indefinite elliptic problems*, SIAM J. Sci. Stat. Comp., 13 (1992), pp. 243–258.

[12] T. F. COLEMAN AND J. J. MORÉ, *Estimation of sparse Jacobian matrices and graph coloring problem*, SIAM J. Numer. Anal., 20 (1983), pp. 243–209.

[13] J. W. DEMMEL, J. R. GILBERT, AND X. S. LI, *SuperLU Users' Guide*, Lawrence Berkeley National Laboratory, 2003.

[14] J. DOUGLAS AND J. WANG, *An absolutely stabilized finite element for the Stokes problem*, Math. Comp., 52 (1989), pp. 495–508.

[15] H. C. ELMAN AND G. H. GOLUB, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal., 31 (1994), pp. 1645–1661.

[16] H. C. ELMAN, *Multigrid and Krylov subspace methods for the discrete Stokes equations*, Int. J. Numer. Meth. Fluids, 22 (1996), pp. 755–770.

[17] H. C. ELMAN, V. E. HOWLE, J. N. SHADID, AND R. S. TUMINARO, *A parallel block multi-level preconditioner for the 3D incompressible Navier-Stokes equations*, J. Comput. Phys., 187 (2003), pp. 504–523.

[18] H. C. ELMAN AND D. SILVESTER, *Fast nonsymmetric iterations and preconditioning for Navier-Stokes equations*, SIAM J. Sci. Comp., 17 (1996), pp. 33–46.

[19] U. GHIA, K. N. GHIA, AND C. T. SHIN, *High-Re solution for incompressible flow using the Navier- Stokes equations and the multigrid method*, J. Comput. Phys., 48 (1982), pp. 387–411.

[20] M. D. GUNZBURGER, *Finite Element Methods for Viscous Incompressible Flows*, Academics Press, New York, 1989.

[21] M. D. GUNZBURGER, *Perspectives in Flow Control and Optimization*, SIAM, Philadelphia, PA, 2003.

[22] F.-N. HWANG AND X.-C. CAI, *A parallel nonlinear additive Schwarz preconditioned inexact Newton algorithm for incompressible Navier-Stokes equations*, J. Comput. Phys., (2005), pp. 666–691.

[23] V. JOHN, *A comparison of parallel solvers for the incompressible Navier-Stokes equations*, Comput. Visual. Sci., 1 (1999), pp. 193–200.

[24] V. JOHN, G. MATTHIES, T. I. MITKOVA, L. TOBISKA, AND P. S. VASSILEVSKI, *A comparison of three solvers for the incompressible Navier-Stokes equations*, in Large-Scale Scientific Computations of Engineering and Environmental Problems II, M. Griebel et al., eds., Vieweg-Verlag, NY, 2000, pp. 215–222.

[25] V. JOHN AND L. TOBISKA, *Numerical performance of smoothers in coupled multigrid methods for the parallel solution of the incompressible Navier-Stokes equations*, Int. J. Numer. Meth. Fluids, 33 (2000), pp. 453–473.

[26] A. KLAWONN, *Block-triangular preconditioners for saddle point problems with a penalty term*, SIAM J. Sci. Comp., 19 (1998), pp. 172–184.

[27] A. KLAWONN, *An optimal preconditioner for a class of saddle point problems with a penalty term*, SIAM J. Sci. Comput., 19 (1998), pp. 540–552.

[28] A. KLAWONN AND L. F. PAVARINO, *Overlapping Schwarz methods for mixed linear elasticity and Stokes problems*, Comput. Methods Appl. Mech. Engrg., 165 (1998), pp. 233–245.

[29] A. KLAWONN AND L. F. PAVARINO, *A comparison of overlapping Schwarz methods and block preconditioners for saddle point problems*, Numer. Linear Algebra Appl., 7 (2000), pp. 1–25.

[30] J. LI, *Dual-Primal FETI Methods for Stationary Stokes and Navier-Stokes Equations*, PhD thesis, Courant Institute of Mathematical Sciences, TR-830, New York University, 2002.

[31] J. MANDEL, *Hybrid domain decomposition with unstructured subdomains*, Contemp. Math., 157 (1994), pp. 103–112.

[32] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer-Verlag, New York, NY, 1999.

[33] S. V. PATANKER, *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, 1980.

[34] L. F. PAVARINO, *Indefinite overlapping Schwarz methods for time-dependent Stokes problems*, Comput. Methods Appl. Mech. Engrg., 187 (2000), pp. 35–51.

[35] E. PRUDENCIO, R. BYRD, AND X.-C. CAI, *Parallel full space SQP Lagrange-Newton-Krylov-Schwarz algorithms for PDE-constrained optimization problems*, SIAM J. Sci. Comput., 27 (2006), pp. 1305-1328.

[36] A. QUARTERONI AND A. VALLI, *Domain Decomposition Methods for Partial Differential Equations*, Oxford University Press, Oxford, 1999.

[37] J. N. REDDY AND D. K. GARTLING, *The Finite Element Method in Heat Transfer and Fluid Dynamics*, CRC Press, Florida, 2000.

[38] Y. SAAD, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comp., 14 (1993), pp. 461–469.

[39] Y. SAAD AND M. H. SCHULTZ, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 7 (1986), pp. 856–869.

[40] J. N. SHADID, R. S. TUMINARO, AND H. F. WALKER, *An inexact Newton method for fully coupled solution of the Navier-Stokes equations with heat and mass transport*, J. Comput. Phys., 137 (1997), pp. 155–185.

ETNA
Kent State University
etna@mcs.kent.edu

[41] D. SILVESTER, H. ELMAN, D. KAY, AND A. WATHEN, *Efficient preconditioning of the linearized Navier-Stokes equations*, J. Comput. Appl. Math., 128 (2001), pp. 261–279.
[42] B. SMITH, P. BJØRSTAD, AND W. GROPP, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, Cambridge, 1996.
[43] A. TOSELLI AND O. WIDLUND, *Domain Decomposition Methods - Algorithms and Theory*, Springer, Berlin, 2005.
[44] R. S. TUMINARO, C. H. TONG, J. N. SHADID, K. D. DEVINE, AND D. M. DAY, *On a multilevel preconditioning module for unstructured mesh Krylov solvers: Two-level Schwarz*, Comm. Numer. Methods Engrg., 18 (2002), pp. 383–389.
[45] A. ZSAKI, D. RIXEN, AND M. PARASCHIVOIU, *A substructure-based iterative inner solver coupled with Uzawa's algorithm for the Stokes problem*, Int. J. Numer. Meth. Fluids, 43 (2003), pp. 215–230.