A substructure-based iterative inner solver coupled with Uzawa's algorithm for the Stokes problem [*]

Attila Zsaki[†]& Daniel Rixen[‡]& Marius Paraschivoiu [¶]

Abstract

A domain decomposition method with Lagrange multipliers for the Stokes problem is developed and analyzed. A common approach to solve the Stokes problem, termed the Uzawa algorithm, is to decouple the velocity and pressure. This approach yields the Schur complement system for the pressure Lagrange multiplier which is solved with an iterative solver. Each outer iteration of the Uzawa procedure involves the inversion of a Laplacian in each spatial direction. The objective of this paper is to effectively solve this inner system (the vector Laplacian system) by applying the finite element tearing and interconnecting (FETI) method. Previously calculated search directions for the FETI solver are reused in subsequent outer Uzawa iterations. The advantage of the approach proposed in this paper is that pressure is continuous across the entire computational domain. Numerical tests are performed by solving the driven cavity problem. An analysis of the number of outer Uzawa iterations and inner FETI iterations is reported. Results show that the total number of inner iterations is almost numerically scalable since it grows asymptotically with the mesh size and the number of subdomains.

1 Introduction

Linear systems arising from spatial discretizations of fluid mechanics problems grow rapidly with the size of the problem. For large problems iterative solvers and parallel computing are essential. To accelerate convergence of the iterative process, preconditioning of the entire system is required. Classical preconditioners approximate the inverse of the global operator, however, this task is difficult if the same problem is to be solved on multi-processors. One approach is to decompose a domain into subdomains, so that the global system is decomposed into local systems. Local systems can either contribute to the global problem at each iteration in a global iterative approach or can be solved independently with boundary conditions imposed on the interface of the subdomains. For parallel computations, the objective is to construct a preconditioner with a local decomposition

[*] submitted to International Journal for Numerical Methods in Fluids

[†] Department of Civil Engineering, University of Toronto E-mail: atila.zaki@utoronto.ca

[‡] TU-Delft, Engineering Mechanics, Mekelweg 2, Delft, 2628 CD, The Netherlands E-mail: d.j.rixen@wbmt.tudelft.nl

[§] Department of Mechanical and Industrial Engineering, Concordia University, 1455 de Maisonneuve Blvd. W. Montreal, Quebec. E-mail: paraschi@me.concordia.ca

[¶] corresponding author

and/or a coarse problem that still provides parallel and numerical scalability, namely such that the speed-up of the computing time is nearly proportional to the number of processors and that large-scale problems can be solved in a similar number of iterations as small-scale problems.

The overlapping Schwarz methods proposed by Dryja and Widlund [2] have been successful for this task (see also e.g. [11, 19]). In [9], an overlapping Schwarz procedure was also used to solve the incompressible Navier-Stokes equations. Nevertheless, these methods have some drawbacks because the overlap increases the interprocessor communication which decreases the parallel efficiency. In addition, these preconditioners are more difficult to implement for complex three-dimensional problems. An alternative is to construct non-overlapping methods.

In this work, we consider incompressible fluid flows with very low Reynolds number leading to the so-called Stokes problem. The Stokes problem are also encountered in incompressible structures and is therefore of fundamental interest in engineering.

Non-overlapping domain decomposition solvers for the Stokes problem have been proposed by several authors [1, 12, 22, 16, 10]. The methods published therein typically assume that the pressure field across the interface is discontinuous, either because the mixed finite element discretization uses piece-wise constant pressure fields, or because the pressure continuity across the interface is relaxed.

Dual domain decomposition approaches where the velocity field on the interface is enforced by Lagrange multipliers were investigated in [1, 10]. These methods correspond to standard Finite Element Tearing and Interconnecting (FETI) approaches [8], but it appears that using the related standard Dirichlet or lumped preconditioners is not straightforward and might lead to suboptimal convergence. In [12], an additional coarse grid problem related to constant pressure fields per subdomains was included and a Dual-Primal FETI approach [6] was applied.

In [22, 16], primal domain decomposition methods closely related to the Balancing Neumann-Neumann [14] method were investigated for the Stokes problem. The method in [22] is related to hierarchical finite elements and was extended to Navier-Stokes problems in [15]. In [16] a Balancing method with an additional coarse grid of the subdomain constant pressures was proposed (primal counterpart of the work in [12]).

In [1], the authors also present some results for a FETI-like method when the interface pressure continuity is enforced, but the convergence is very poor due to the fact that standard FETI preconditioners are no longer applicable in that case.

In the present work, we introduce and study an iterative solver that combines Uzawa iterations for the pressure field while the velocity problem is solved by a standard FETI approach. In this way, we do not have to relax the pressure continuity on the interface and therefore we can use Taylor-Hood elements [23] to discretize the Stokes problem with quadratic velocity fields and linear pressure fields continuous across elements.

Applying Uzawa's algorithm to the Stokes problem is common. It consists of first decoupling the Stokes saddle point problem into the Schur complement system for pressure and second solving each component of velocity by inverting a Laplace operator. This approach suffers from the fact that one solution of a symmetric positive-definite system of linear equations for each velocity component must be calculated at each iteration. To address this problems, Elman and Golub have proposed an approximate iterative solution [4]. This preconditioned inexact Uzawa algorithm seems to be competitive with multigrid and Krylov subspace methods [3]. The approach followed in this paper is however different from [4]. Indeed, here the Poisson problem is solved at every Uzawa iteration by FETI for which inherent parallel and efficient preconditioners can be used. In addition, this method is enhanced by reusing the previously calculated search directions allowing additional reduction in computational cost.

The remainder of this paper is structured as follows. In Section 2, the Stokes problem, the variational form, the finite element discretization and the domain decomposition is described. Uzawa's algorithm is reviewed in Section 3. In Section 4, the FETI method is summarized with and without reconjugaison. Results and comparison of the different methods are reported in Section 5.

2 The steady Stokes problem

2.1 Governing equations

The steady creeping flow of an incompressible ($\rho =$ constant) Newtonian fluid with constant dynamic viscosity in a driven cavity is considered.

To describe this flow the Laplacian form of the incompressible Stokes equations is used. In vector notation, the velocity vector $\mathbf{u}$ and pressure p satisfy

$$\begin{aligned} -\mu\Delta\mathbf{u} + \nabla p &= \mathbf{f} \text{ in } \Omega \qquad (1) \\ -\nabla\cdot\mathbf{u} &= 0 \text{ in } \Omega \end{aligned}$$

with imposed velocity $\mathbf{u_D} = \mathbf{e_1}$ on the top boundary and no-slip Dirichlet $\mathbf{u} = 0$ elsewhere. The vector $\mathbf{e_1}$ is the unit vector in the x direction. Let $\Omega \in R^2$ be a square domain with sides $\Gamma_j,\ j = \{1, ..., 4\}$.

2.2 Variational formulation

Introducing, $\mathcal{L}^2(\Omega)$ the space of functions which are square integrable over Ω and $\mathcal{H}^1(\Omega)$ the space of functions v such that $v \in \mathcal{L}^2(\Omega)$ and $\nabla v \in \mathcal{L}^2(\Omega)$ then

$$\mathcal{L}^2_0(\Omega) = \{q \in \mathcal{L}^2(\Omega) | \int_\Omega q \mathrm{d}A = 0\} \qquad (2)$$

$$\mathcal{H}^1_0(\Omega) = \{v \in \mathcal{H}^1(\Omega) |\ v|_{\Gamma_i} = 0, i \in \{1, 2, 3, 4\}\}. \qquad (3)$$

The variational form is: Given $\mathbf{f} \in \mathcal{H}^{-1}(\Omega)^2$ find $\mathbf{u} \in (\mathcal{H}^1_0(\Omega))^2$ and $p \in \mathcal{L}^2_0(\Omega)$ such that

$$a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = L(\mathbf{v}) \ \forall \mathbf{v} \in (\mathcal{H}^1_0(\Omega))^2 \qquad (4)$$

$$b(\mathbf{u}, q) = -b(\mathbf{u}_D, q) \ \forall q \in \mathcal{L}^2_0(\Omega) \qquad (5)$$

where $\mathbf{u} + \mathbf{u}_D$ is the velocity field and p is the pressure field.

In this context the bilinear forms a and b and the linear form L are defined by

$$a(\mathbf{u}, \mathbf{v}) = \int_\Omega \mu\nabla\mathbf{u}\cdot\nabla\mathbf{v}\, \mathrm{d}A \ \forall \mathbf{u}, \mathbf{v} \in (\mathcal{H}^1_0(\Omega))^2 \qquad (6)$$

$$b(\mathbf{v}, q) = \int q \ \operatorname{div}(\mathbf{v})\, \mathrm{d}A \ \forall \mathbf{v} \in \mathcal{H}^1_0(\Omega)^2, \ \forall q \in \mathcal{L}^2_0(\Omega) \qquad (7)$$

$$L(\mathbf{v}) = \int_\Omega \mathbf{f}\cdot\mathbf{v}\, \mathrm{d}A - a(\mathbf{u}_D, \mathbf{v}). \qquad (8)$$

2.3 Finite element discretization

The continuous form of the Stokes problem is discretized in $I\!R^2$ using a finite element approximation $(\mathbf{u}_h, p_h)$ to $(\mathbf{u}, p)$. The computational domain Ω is decomposed into K conforming triangular elements,

$$\Omega = \bigcup_K \overline{T}_h^k \tag{9}$$

The discrete problem is then: Find $\mathbf{u}_h \in V_h$ and $p_h \in W_{h0}$ such that

$$\begin{aligned} a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) &= L(\mathbf{v}_h) \qquad \forall \mathbf{v}_h \in V_h \\ b(\mathbf{u}_h, q_h) &= -b(\mathbf{u}_{\mathbf{D}h}, q_h) \;\; \forall q_h \in W_{h0} \end{aligned} \tag{10}$$

where V_h and W_{ho} are the Taylor-Hood approximation spaces [23] defined as

$$V_h = \{\mathbf{v}_h|_{T_h} \in (P_2(T_h))^2, \; \forall T_h \in \mathcal{T}_h\} \cap (\mathcal{H}_0^1(\Omega))^2 \tag{11}$$

$$W_{h0} = \{q_h|_{T_h} \in P_1(T_h), \; \forall T_h \in \mathcal{T}_h\} \cap \mathcal{L}_0^2(\Omega). \tag{12}$$

Note that both velocity and pressure fields are continuous over Ω. Let's denote by A the discrete Laplacian and by D_i the derivative matrix in each direction and f_i the augmented inhomogeneity for each direction which incorporates the boundary terms. The matrix form of the Stokes problem (10) is then

$$\begin{bmatrix} A & 0 & -D_1^T \\ 0 & A & -D_2^T \\ -D_1 & -D_2 & 0 \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ p \end{pmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ 0 \end{bmatrix} \tag{13}$$

2.4 Domain decomposition

A non-overlapping decomposition of Ω into N subdomains is constructed such that

$$\Omega = \bigcup_s \overline{\Omega}_s \tag{14}$$

For simplicity each subdomain is a square of size $H \times H$ as shown in Figure 1. A set of discrete equations is assembled by integrating the expression (6),(7) and (8) over individual subdomains. On each subdomain, the discrete Laplacian is denoted by $A^{(s)}$, the derivative operator by $D_i^{(s)}$ and the component of the nodal forces and boundary terms by $f_i^{(s)}$. Nodal values of velocity are stored in $u_i^{(s)}$ and nodal pressure values are stored in p. The restriction of p on each subdomain is obtained by multiplication with an operator $Q^{(s)}$. The inter-subdomain continuity condition for each velocity component is imposed through the construction of a matrix $\mathtt{B} = [B^{(1)} \dots B^{(N_{sub})}]$, such that $\sum_{N_{sub}} B^{(s)} u_1^{(s)} = 0$ and $\sum_{N_{sub}} B^{(s)} u_2^{(s)} = 0$.

The global system can be written in discrete form as

$$\begin{aligned} A^{(s)} u_1^{(s)} &= f_1^{(s)} - {B^{(s)}}^T \lambda_1 + {D_1^{(s)}}^T Q^{(s)} p, \;\; s = 1, ..., N_{sub} \\ A^{(s)} u_2^{(s)} &= f_2^{(s)} - {B^{(s)}}^T \lambda_2 + {D_2^{(s)}}^T Q^{(s)} p, \;\; s = 1, ..., N_{sub} \\ \sum_{N_{sub}} B^{(s)} u_1^{(s)} &= 0 \end{aligned} \tag{15}$$

$$\begin{aligned} \sum_{N_{sub}} B^{(s)} u_2^{(s)} &= 0 \\ \sum_{N_{sub}} \left(D_1^{(s)} u_1^{(s)} + D_2^{(s)} u_2^{(s)} \right) &= 0 \end{aligned}$$

Introducing

$$\mathtt{A} = \begin{bmatrix} A^{(1)} & 0 & \dots & 0 \\ 0 & A^{(2)} & & \\ \vdots & & \ddots & \\ 0 & \dots & 0 & A^{(N_{sub})} \end{bmatrix} \tag{16}$$

$$\mathtt{B} = \left[B^{(1)}\ B^{(2)}\ \dots B^{(N_{sub})} \right] \tag{17}$$

$$\mathtt{D}_1 = \left[D_1^{(1)}\ D_1^{(2)}\ \dots D_1^{(N_{sub})} \right] \tag{18}$$

$$\mathtt{D}_2 = \left[D_2^{(1)}\ D_2^{(2)}\ \dots D_2^{(N_{sub})} \right] \tag{19}$$

$$\mathtt{u}_1 = \begin{bmatrix} \mathtt{u}_1^{(1)} \\ \mathtt{u}_1^{(2)} \\ \vdots \\ \mathtt{u}_1^{(N_{sub})} \end{bmatrix} \tag{20}$$

$$\mathtt{u}_2 = \begin{bmatrix} \mathtt{u}_2^{(1)} \\ \mathtt{u}_2^{(2)} \\ \vdots \\ \mathtt{u}_2^{(N_{sub})} \end{bmatrix} \tag{21}$$

$$\mathtt{f}_1 = \begin{bmatrix} \mathbf{f}_1^{(1)} \\ \mathtt{f}_1^{(2)} \\ \vdots \\ \mathtt{f}_1^{(N_{sub})} \end{bmatrix} \tag{22}$$

$$\mathtt{f}_2 = \begin{bmatrix} \mathbf{f}_2^{(1)} \\ \mathtt{f}_2^{(2)} \\ \vdots \\ \mathtt{f}_2^{(N_{sub})} \end{bmatrix} \tag{23}$$

The system becomes

$$\begin{bmatrix} \mathtt{A} & \mathtt{B}^T & 0 & 0 & -\mathtt{D}_1^T \\ \mathtt{B} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathtt{A} & \mathtt{B}^T & -\mathtt{D}_2^T \\ 0 & 0 & \mathtt{B} & 0 & 0 \\ -\mathtt{D}_1 & 0 & -\mathtt{D}_2 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \mathrm{u}_1 \\ \lambda_1 \end{bmatrix} \\ \begin{bmatrix} \mathrm{u}_2 \\ \lambda_2 \end{bmatrix} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ 0 \\ \mathbf{f}_2 \\ 0 \\ 0 \end{bmatrix} \tag{24}$$

Let

$$U = \begin{bmatrix} \mathtt{u}_1 \\ \lambda_1 \\ \mathtt{u}_2 \\ \lambda_2 \end{bmatrix}, \mathcal{A} = \begin{bmatrix} \mathtt{A} & \mathtt{B}^T & 0 & 0 \\ \mathtt{B} & 0 & 0 & 0 \\ 0 & 0 & \mathtt{A} & \mathtt{B}^T \\ 0 & 0 & \mathtt{B} & 0 \end{bmatrix}, \mathcal{D} = \begin{bmatrix} \mathtt{D}_1 \\ 0 \\ \mathtt{D}_2 \\ 0 \end{bmatrix}, \mathcal{F} = \begin{bmatrix} \mathbf{f}_1 \\ 0 \\ \mathbf{f}_2 \\ 0 \end{bmatrix}$$

then

$$\begin{bmatrix} \mathcal{A} & -\mathcal{D}^T \\ -\mathcal{D} & 0 \end{bmatrix} \begin{bmatrix} U \\ p \end{bmatrix} = \begin{bmatrix} \mathcal{F} \\ 0 \end{bmatrix} \tag{25}$$

The solution of this positive semi-definite system with zeros on the diagonal at degrees of freedom associated with pressure and continuity enforced across inter-subdomains is difficult. This work presents a domain decomposition iterative solver to efficiently solve the above system (25).

3 Uzawa's method

In the context of a non-decoupled domain, a popular iterative method to solve the Stokes problem is the Uzawa saddle-decoupling algorithm. This algorithm is based on constructing a positive semi-definite problem for the pressure, i.e.,

$$(\mathtt{D}_1\mathtt{A}^{-1}\mathtt{D}_1^T + \mathtt{D}_2\mathtt{A}^{-1}\mathtt{D}_2^T)p = -\mathtt{D}_1\mathtt{A}^{-1}\mathbf{f}_1 - \mathtt{D}_2\mathtt{A}^{-1}\mathbf{f}_2 \tag{26}$$

by substituting the velocity vector in the velocity divergence equation. Once p is determined the velocity is obtained by solving the viscous system

$$\mathtt{A}\mathtt{u}_1 = \mathtt{D}_1^T p + \mathbf{f}_1 \tag{27}$$

$$\mathtt{A}\mathtt{u}_2 = \mathtt{D}_2^T p + \mathbf{f}_2 \tag{28}$$

Because the matrix $S = (\mathtt{D}_1\mathtt{A}^{-1}\mathtt{D}_1^T + \mathtt{D}_2\mathtt{A}^{-1}\mathtt{D}_2^T)$ is symmetric and positive-definite, the pressure field can be solved using a preconditioned $(\tilde{M})$ conjugate gradient (CG) algorithm.

The Uzawa algorithm is summarized as:

1. Initialize

$$\begin{aligned} p^0 &= 0 \\ w^0 &= -\mathtt{D}_1\mathtt{A}^{-1}\mathbf{f}_1 - \mathtt{D}_2\mathtt{A}^{-1}\mathbf{f}_2 \end{aligned}$$

2. Iterate

$$\begin{aligned} y^n &= \tilde{M}^{-1}w^n \\ z^n &= y^n - \sum_{i=0}^{n-1} \frac{y^{nT} S z^i}{z^{iT} S z^i} z^i \\ \eta^n &= \frac{z^{nT} w^n}{z^{nT} S z^n} \\ p^{n+1} &= p^n + \eta^n z^n \\ w^{n+1} &= w^n - \eta^n S p^n \end{aligned}$$

Note that the inversion of two Laplace operators, $\mathtt{A}$, is required at each pressure iteration when the S matrix operates on a vector and therefore a nested elliptic iteration solver must be used. Furthermore, the initialization step also involves the inverse of the $\mathtt{A}$ matrix. Since this operation is symmetric, positive definite it is also solved using a conjugate gradient iteration solver. Clearly the reduction of the number of iterations of the nested elliptic iterations reduces significantly the computational cost. The FETI solver is ideal to reduce the cost of the inner iterations. Herein it is proposed to use the FETI solver as well as a reconjugaison method for these inversions.

The advantages of Uzawa approach are three-fold: first, a CG algorithm can be used to solve the pressure problem, second the outer iterations are independent of the discretization size (h) [3], third, an excellent preconditioner ,$\tilde{M}$, for the pressure system is this pressure mass matrix [13]. Furthermore, it was shown in [19] that the number of Uzawa outer iterations is always less than other iterative domain decomposition methods. Therefore the Uzawa algorithm is the baseline that is improved in this work.

4 Review of the FETI method

4.1 FETI and its preconditioners

A domain decomposition is applied for the Laplace operator. Therefore, at each iteration, the inversion of this operator is performed using the FETI method. The resulting system, at each iteration, is written in a generic block form as:

$$\begin{bmatrix} \mathtt{A} & \mathtt{B}^T \\ \mathtt{B} & 0 \end{bmatrix} \begin{bmatrix} \xi \\ \Lambda \end{bmatrix} = \begin{bmatrix} g \\ 0 \end{bmatrix} \tag{29}$$

Eliminating the field solution (ξ) for the above equation leads to the interface problem for Λ also known as the dual Schur complement problem. To be more precise:

$$\xi^{(s)} = A^{(s)+} \left(g^{(s)} - B^{(s)^T} \Lambda \right) - R^{(s)} \alpha^{(s)} \tag{30}$$

where $A^{(s)^+}$ is a generalized inverse of $A^{(s)}$. For all semi-indefinite sub-systems, $R^{(s)}\alpha^{(s)}$ is added to each component of velocity where $R^{(s)}$ are the associated floating modes of that sub-system s and $\alpha^{(s)}$ are the amplitudes. The additional unknowns $\alpha^{(s)}$ are determined such that the interface fluxes are in equilibrium with the right-hand side $g^{(s)}$,i.e., such that the subdomain equilibrium problem is well-posed $R^{(s)^T} \left(g^{(s)} - B^{(s)^T} \Lambda \right) = 0$. Note that $A^{(s)}$ has a null space $R^{(s)}$ equal to 1 corresponding to the constant floating mode. This latest fact reduces the computational cost significantly because $R^{(s)}$ is not calculated nor stored.

$$\begin{bmatrix} \mathtt{F}_I & \mathtt{G}_I \\ \mathtt{G}_I{}^T & 0 \end{bmatrix} \begin{bmatrix} \Lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} d \\ e \end{bmatrix} \tag{31}$$

$$\mathtt{F}_I = \sum_{s=1}^{N_{sub}} B^{(s)} A^{(s)^+} B^{(s)^T} \tag{32}$$

$$\mathtt{G}_I = \left[B^{(1)} R^{(1)} \ \ \ldots \ \ B^{(N_{sub})} R^{(N_{sub})} \right] \tag{33}$$

$$\begin{aligned} \alpha &= \left[\alpha^{(1)} \ \ \dots \ \ \alpha^{(N_{sub})}\right] && (34) \\ d &= \sum_{s=1}^{N_{sub}} B^{(s)} A^{(s)^+} g^{(s)} && (35) \\ e &= \left[R^{(1)^T} g^{(1)}\right] \ \ \dots \ \ \left[R^{(N_{sub})^T} g^{(N_{sub})}\right] && (36) \end{aligned}$$

The constraint $G^T \mathtt{\Lambda} = e$ ensures that $g^{(s)} \in$ range $A^{(s)}$ for all $s = 1, ..., N_{sub}$.

The construction of this problem is motivated by the use of a conjugate gradient (CG) algorithm, which allows for an iterative solution. A projector is applied to force Λ to belong to the range of $\mathtt{A}$ for all iterations.

The FETI method iterates on $\mathtt{\Lambda}$ to solve the interface problem (31). In brief the method is a Preconditioned Conjugate Gradient solver with two projector steps at each iteration to handle the self-equilibrium constraints. The projector is orthogonal to Ker$(\mathtt{G}_I^T)$. It is defined by

$$P = I - \mathtt{G}_I (\mathtt{G}_I^T \mathtt{G}_I)^{-1} \mathtt{G}_I^T \qquad (37)$$

such that at each iteration $\mathtt{G}_I^T (\mathtt{\Lambda}^n - \mathtt{\Lambda}^{n-1}) = 0$.

To satisfy the original constraint which is $\mathtt{G}_I^T \mathtt{\Lambda} = e$, the Lagrange multiplier $\mathtt{\Lambda}$ is split into two components

$$\mathtt{\Lambda} = \mathtt{\Lambda}^0 + P \mathtt{\Lambda}^n \qquad (38)$$

where $\mathtt{\Lambda}^0$ satisfies the $\mathtt{G}_I^T \mathtt{\Lambda}^0 = e$ constraint and $\mathtt{\Lambda}^n$ is the solution of a symmetric positive semi-definite problem

$$P^T \mathtt{F}_I P \mathtt{\Lambda}^n = P^T (d - \mathtt{F}_I \mathtt{\Lambda}^0). \qquad (39)$$

More information about the original FETI algorithm is found abundantly in the literature [8, 7, 18].

This algorithm can be summarized as:

1. Initialize

$$\begin{aligned} \mathtt{\Lambda}^0 &= \mathtt{G}_I (\mathtt{G}_I^T \mathtt{G}_I)^{-1} e \\ w^0 &= P^T (d - \mathtt{F}_I \mathtt{\Lambda}^0) \end{aligned}$$

2. Iterate

$$\begin{aligned} y^n &= P \tilde{F}_I^{-1} w^n \\ z^n &= y^n - \sum_{i=0}^{n-1} \frac{y^{n^T} \mathtt{F}_I z^i}{z^{i^T} \mathtt{F}_I z^i} z^i \\ \eta^n &= \frac{z^{n^T} w^n}{z^{n^T} \mathtt{F}_I z^n} \\ \mathtt{\Lambda}^{n+1} &= \mathtt{\Lambda}^n + \eta^n z^n \\ w^{n+1} &= w^n - \eta^n P^T \mathtt{F}_I p^n \end{aligned}$$

The above algorithm includes a preconditioner of the interface system to ensure scalability when the number of elements and/or the number of subdomains is increased. A brief description of the preconditioner of the interface compatibility follows.

Two preconditioners have been introduced and studied to precondition the dual operator $\mathtt{F}_I$ [8]. The first preconditioner is based on the subdomain Schur complement which requires to solve local Dirichlet problems. Each subdomain viscous operator matrix is first partitioned as

$$A^{(s)} = \begin{bmatrix} A_{ii}^{(s)} & A_{ib}^{(s)} \\ A_{ib}^{(s)} & A_{bb}^{(s)} \end{bmatrix} \tag{40}$$

where the subscript i and b denote the interior and subdomain boundary degrees-of-freedom, respectively. The subdomain Schur complement can be written as:

$$\mathtt{S}_{bb}^{(s)} = A_{bb}^{(s)} - A_{ib}^{(s)T} A_{ii}^{(s)+} A_{ib}^{(s)} \tag{41}$$

The resulting Dirichlet preconditioner is given by

$$\tilde{\mathtt{F}}_D^{-1} = \sum_{i=1}^{i=N} B_i \begin{bmatrix} 0 & 0 \\ 0 & \mathtt{S}_{bb}^{(s)} \end{bmatrix} B_i{}^T \tag{42}$$

To avoid solving the local Dirichlet problems a lumped preconditioner has been introduced. This computationally more efficient preconditioner can be written as

$$\tilde{\mathtt{F}}_L^{-1} = \sum_{i=1}^{i=N} B_i \begin{bmatrix} 0 & 0 \\ 0 & A_{bb}^{(s)} \end{bmatrix} B_i{}^T \tag{43}$$

In this study, only the Dirichlet preconditioner is used.

4.2 FETI for multiple right-hand sides: the reconjugaison

In this work, the FETI method is used at every Uzawa iteration on the pressure to solve the decomposed Laplacian problem. In other words, for a given estimate of the pressure p in the outer Uzawa iteration, the FETI method will iterate on the interface flexibility problem (31) to compute the Lagrange multipliers Λ on the interface. Hence, at every Uzawa outer iteration, the same interface problem (31) must be solved with different right-hand sides.

When factorization techniques are used to solve linear systems, the factorization of the operator must be performed only once. It can then be used to solve for different right-hand sides by forward and backward substitution. When iterative solvers are used such as in the FETI method, information about the inverse of the operator that were obtained during previous iterations can also be used when solving for different right-hand sides.

The FETI method is essentially a preconditioned conjugate gradient procedure on the dual interface problem (39). The conjugate gradient iterations produce search directions that are conjugate, namely which are orthogonal with respect to the system operator. Moreover, the application of the system operator on these directions are computed during the iterations. Therefore, if we store the directions z and $\mathtt{F}_I z$ generated during the iterations in the algorithm described in the previous section, the conjugate gradient algorithm can be started by first searching in the subspace of the orthogonal basis generated earlier. Then the iteration proceeds by ensuring the new search directions constructed are conjugate to the stored ones.

Hence the FETI algorithm for multiple right-hand sides can be summarized as follows. Calling Z and X the matrices containing the previous search directions z and $\mathtt{F}_I z$ respectively,

1. Initialize

$$\begin{aligned}\mathtt{\Lambda}^{00} &= \mathtt{G}_I(\mathtt{G}_I^T\mathtt{G}_I)^{-1}e \\ w^{00} &= P^T(d - \mathtt{F}_I\mathtt{\Lambda}^{00})\end{aligned}$$

2. Projection

$$\begin{aligned}\alpha &= Z^T w^{00} \\ \Lambda^0 &= \mathtt{\Lambda}^{00} + Z\alpha \\ w^0 &= w^{00} - X\alpha\end{aligned}$$

3. Iterate

$$\begin{aligned}y^n &= P\tilde{F}_I^{-1}w^n \\ z^n &= y^n - Z(X^T y^n) \\ \eta^n &= \frac{z^{nT}w^n}{z^{nT}\mathtt{F}_I z^n} \\ \mathtt{\Lambda}^{n+1} &= \mathtt{\Lambda}^n + \eta^n z^n \\ w^{n+1} &= w^n - \eta^n P^T\mathtt{F}_I p^n \\ &\ \text{store } z^n \text{ in } Z \text{ and } \mathtt{F}_I z^n \text{ in } X\end{aligned}$$

The part of the solution that is in the subspace of the previous search directions Z is computed at nearly no cost at the beginning of the iteration. The conjugate gradient procedure will iterate only in the complementary subspace and thus converge to the solution faster. As a matter of fact, if the previous search directions significantly contribute to the new solution, convergence will be achieved within a few iterations. This procedure is known as a *reconjugaison* or *projection and re-orthogonalization* method.

The additional cost incurred by the projection and re-orthogonalization steps in the algorithm for multiple right-hand sides is small as long as the number of stored directions is small compared to the problem size. It is also possible to fix the number of stored directions altough this approach is not considered here. Hence, this technique will be very cost effective if all the solutions can be well represented by a limited number of Krylov vectors generated in the iterations. Further discussion of this method and of its application to multiple right-hand sides can be found in [21, 5, 20, 17].

5 Results

In this section the convergence behavior of the Uzawa and the FETI method for Stokes flow in a driven cavity is reported. The computational domain is a unit cavity with Dirichlet boundary conditions for velocity. The velocity in the x direction is equal to one on one side and is equal to zero everywhere else. The number of subdomains, N_{sub}, and the mesh size, h, are related to a uniform triangular mesh as shown in Figure 1. The following stopping criterion is used for the FETI method

$$\frac{\|w^n\|_2}{\|d\|_2} \le 10^{-8} \tag{44}$$

where $\|w^n\|_2$ is the projected residual at the nth iteration and $\|d\|_2$ is the right-hand side of the FETI interface problem. The convergence criteria for the Uzawa solver is

$$\frac{\|w^n\|_2}{\|w^0\|_2} \leq 10^{-6} \tag{45}$$

where $\|w^n\|_2$ is the pressure residual at the nth iteration. All results are obtained in MATLAB 6.0 environment.

The subsections that follow will numerically demonstrate that the number of outer and inner iterations are numerically scalable with respect to the mesh size, h, and the subdomain size, H. Results are tabulated to show the number of outer iterations, the total number of inner iterations for the basic FETI method and the total number of inner iterations when reusing the previously calculated search directions.

5.1 Effect of the mesh size h

The effect of the mesh size h is investigated for a fixed number of subdomains. The number of elements per side is increased from 8 to 48 leading to an increase in the ratio H/h, an increase in the size of the interface problem and an increase in the size of each local problem. Table 1 reports the number of iterations for a fixed number of subdomains, $N_{sub} = 16$. The column "Outer Uzawa iterations" reports the number of iterations of the solution of the dual pressure problem. As expected, the number of iterations is constant for all problem sizes because the Uzawa convergence is independent of the problem size. The next column shows the total number of inner iterations for the basic FETI method and the last column shows the total number of inner FETI iterations with the reconjugaison method. Note that the total inner iterations is the sum of all iterations over the 19 outer Uzawa iterations. A Dirichlet preconditioner $\tilde{\mathtt{F}}_D^{-1}$ is used for all FETI iterations. A small increase in the total number of inner iterations is observed in both cases, nevertheless, this increase is smaller when the reconjugaison is used.

From this table, the following conclusions can be made: first, the Uzawa solver in very scalable; second, the FETI solver is almost numerically scalable; third the reconjugaison method cuts by more than half the number of total iterations.

5.2 Effect of the subdomain size H

To numerically measure scalability with respect to the subdomain size, the number of subdomains is increased from 9 to 225 while keeping the size of the problem fixed. Clearly the number of outer iterations stays the same because the number of pressure d.o.f. is the same. From Table 2, the reader may appreciate that the total number of inner iterations varies slightly but still demonstrates scalability for large number of subdomains. Note that the reconjugaison method is more efficient in reducing the number of inner iterations for a smaller number of subdomains.

In the next set of simulations the number of subdomains and the problem size is increased proportionally to investigate a scaled speed up. Results from this simulation are reported in Table 3. As noticed before, the number of outer iterations in the Uzawa solver does not change with the increase of problem size. Furthermore, the FETI solver requires the same total number of iterations (293). The reconjugaison method requires slightly more iterations as the problem size increases.

H	h	$\frac{H}{h}$	Outer Uzawa it.	Total FETI inner it.	Reconj. Total FETI inner it.
$\frac{1}{4}$	$\frac{1}{8}$	2	18	217	95
$\frac{1}{4}$	$\frac{1}{12}$	3	19	252	107
$\frac{1}{4}$	$\frac{1}{16}$	4	19	269	109
$\frac{1}{4}$	$\frac{1}{20}$	5	19	271	113
$\frac{1}{4}$	$\frac{1}{28}$	7	19	291	115
$\frac{1}{4}$	$\frac{1}{32}$	8	19	291	118
$\frac{1}{4}$	$\frac{1}{36}$	9	19	291	118
$\frac{1}{4}$	$\frac{1}{40}$	10	19	293	119
$\frac{1}{4}$	$\frac{1}{48}$	12	19	311	119

Table 1: Number of outer Uzawa iterations and inner FETI iterations with and without the reconjugaison method for a fixed number of subdomains.

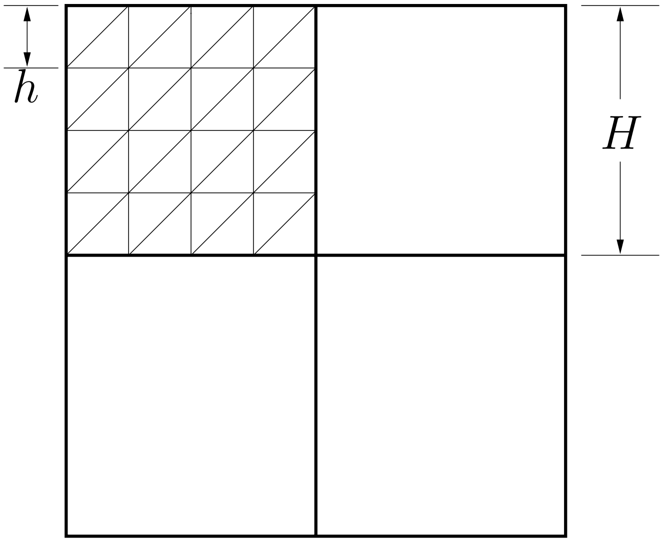

Figure 1:

H	h	$\frac{H}{h}$	Outer Uzawa it.	Total FETI inner it.	Reconj. Total FETI inner it.
$\frac{1}{3}$	$\frac{1}{60}$	20	19	247	85
$\frac{1}{4}$	$\frac{1}{60}$	15	19	312	121
$\frac{1}{5}$	$\frac{1}{60}$	12	19	341	155
$\frac{1}{6}$	$\frac{1}{60}$	10	19	335	190
$\frac{1}{10}$	$\frac{1}{60}$	6	19	308	246
$\frac{1}{12}$	$\frac{1}{60}$	5	19	293	245
$\frac{1}{15}$	$\frac{1}{60}$	4	19	272	240

Table 2: Number of outer Uzawa iterations and inner FETI iterations for a fixed problem size.

H	h	$\frac{H}{h}$	Outer Uzawa it.	Total FETI inner it.	Reconj. Total FETI inner it.
$\frac{1}{4}$	$\frac{1}{20}$	5	19	271	113
$\frac{1}{5}$	$\frac{1}{25}$	5	19	304	146
$\frac{1}{6}$	$\frac{1}{30}$	5	19	293	177
$\frac{1}{7}$	$\frac{1}{35}$	5	19	293	209
$\frac{1}{8}$	$\frac{1}{40}$	5	19	293	225
$\frac{1}{9}$	$\frac{1}{45}$	5	19	293	233
$\frac{1}{10}$	$\frac{1}{50}$	5	19	293	239
$\frac{1}{12}$	$\frac{1}{60}$	5	19	293	245

Table 3: Number of outer Uzawa iterations and inner FETI iterations for a fixed $\frac{H}{h}$.

6 Conclusion

The method proposed herein improves a commonly used strategy, the Uzawa algorithm, for solving the Stokes problem. The Uzawa's algorithm decouples the velocity and the pressure associated with the Stokes problem. This algorithm requires the inversion of a Laplacian for each velocity direction. A domain-decomposition method, the FETI method enhanced with a reconjugaison approach is utilized to perform the inversions of the Laplacian operator. An advantage of this approach is that pressure is approximated with continuous elements while all calculations are based on the FETI domain decomposition approach. The above mentioned FETI approach has been shown to be numerically scalable, i.e. coarse mesh problems require similar number of iterations as fine mesh problems as well as small number of subdomains require similar number of iterations as large number of subdomains. Numerical experiments for a simple driven cavity flow in a square show that the FETI method is indeed numerically scalable and that the reconjugaison approach where the FETI search directions are reused can reduce by three the number of inner iterations for small number of subdomains. This approach can also be applied to unsteady flows where Stokes like problems are solved at each time step. This will be investigated in future papers.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] C. Calgaro and J. Laminie. On the domain decomposition method for the generalized Stokes problem with continuous pressure. *Num. Meth. Part. Dif. Eq.*, 16(1):84–106, January 2000.

[2] M. Dryja and Olof B. Widlund. Domain decomposition algorithms with small overlap. *SIAM J. Sci. Comput.*, 15(3):604–620, 1994.

[3] H. Elman. Multigrid and Krylov subspace methods for discrete Stokes equations. *International journal for numerical methods in fluids*, 22:755–770, 1996.

[4] H. Elman and G.H. Golub. Inexact and preconditioned Uzawa algorithms for saddle point problems. *SIAM J. Numer. Anal.*, 31(6):1645–1661, 1994.

[5] C. Farhat, L. Crivelli, and F.X. Roux. Extending substructures based iteratives solvers to multiple load and repeated analyses. *Comput. Meth. Appl. Mech. Engrg.*, 117:195–209, 1994.

[6] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. FETI-DP: a dual-primal unified FETI method - part i: a faster alternative to the two-level FETI method. *Int. J. Numer. Methods. Engrg.*, 50(7):1523–1544, 2001.

[7] C. Farhat and F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *Comput. Methods. Appl. Mech. Engrg.*, 32:1205–1227, 1991.

[8] C. Farhat and F.-X. Roux. Implicit parallel processing in structural mechanics. *Computational Mechanics Advances*, 2(1):1–124, 1994.

[9] Paul F. Fisher. An overlapping Schwarz method for spectral element solution of the incompressible Navier-Stokes equations. *J. Comp. Phys.*, 133:84–101, 1997.

[10] P. Gosselet, Ch. Rey, F. Lene, and P. Dasset. A domain decomposition method for quasi-incompressible formulations with discontinuous pressure fields. *Rev. Eur. El. Fin.*, 2002. submitted.

[11] A. Klawonn and L.F. Pavarino. Overlapping Schwarz methods for elasticity and Stokes problems. *Comput. Methods Appl. Mech. Engrg.*, 165(1-4):233–245, 1998.

[12] Jing Li. A dual-primal FETI method for incompressible Stokes equations. Technical Report TR2001-816, Courant Institute, New-York, July 2001.

[13] Y. Maday, D. Meiron, A.T. Patera, and E.M. Ronquist. Analysis of iterative methods for the steady and the unsteady Stokes problem: Aplication to spectral element discretizations. *SIAM J. Sci. Comput.*, 14:310–337, 1993.

[14] J. Mandel. Balancing domain decomposition. *Comm. Appl. Num. Meth. Engrg.*, 9:233–241, 1993.

[15] A. Patra. Newton-Krylov domain decomposition solvers for adaptive *hp* approximations of the steady incompressible Navier-Stokes equations with discontinuous pressure fields. In P.E. Bjørstad, M. S. Espedal, and D.E. Keyes, editors, *Ninth international conference on domain decomposition methods*. DDM.org, 1998.

[16] Luca Pavarino and Olof Widlund. Balancing Neumann-Neumann methods for incompressible Stokes equations. Technical Report TR2001-813, Courant Institute, New-York, March 2001.

[17] F. Risler and C. Rey. Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale non linear problems. *Numerical Algorithms*, 23:1–30, 2000.

[18] D. Rixen. Extended preconditioners for FETI method applied to constrained problems. *Internat. J. Num. Meth. Engin.*, 54(1):1–26, 2002.

[19] E. Ronquist. Domain decomposition methods for the steady Stokes equations. In C.-H. Lai, P.E. Bjorstad, M. Cross, and O.B. Widlund, editors, *Eleventh international conference on domain decomposition methods*. DDM.org, 1999.

[20] F.-X. Roux. Parallel implementation of a domain decomposition method for non-linear elasticity problems. In D. Keyes, Y. Saad, and D. Truhlar, editors, *Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering*, pages 161–175. SIAM, 1995.

[21] Y. Saad. On the Lanczos method for solving symmetric linear systems with several right-hand sides. *Math. Comput.*, 48:651–662, 1987.

[22] P. Le Tallec and A. Patra. Non-overlapping domain decomposition methods for adaptive *hp* approximations of the Stokes problem with discontinuous pressure field. *Comput. Methods Appl. Mech. Engrg.*, 145:361–379, 1997.

[23] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element method. *Computers and Fluids*, 1:73–100, 1973.