Introduction to Saddle Point Problems

Michele Benzi

Department of Mathematics and Computer Science
Emory University
Atlanta, GA

Woudschoten, 4 October 2012

Motivation and goals

Let A be a real, symmetric, $n \times n$ matrix and let $f \in \mathbf{R}^n$ be given.

Let $\langle \cdot, \cdot \rangle$ denote the standard inner product in $\mathbf{R}^n$.

Consider the following two problems:

1. Solve $Au = f$
2. Minimize the function $J(u) = \frac{1}{2} \langle Au, u \rangle - \langle f, u \rangle$

Motivation and goals

Let A be a real, symmetric, $n \times n$ matrix and let $f \in \mathbf{R}^n$ be given.

Let $\langle \cdot, \cdot \rangle$ denote the standard inner product in $\mathbf{R}^n$.

Consider the following two problems:

1. Solve $Au = f$
2. Minimize the function $\quad J(u) = \frac{1}{2} \langle Au, u \rangle - \langle f, u \rangle$

Note that $\nabla J(u) = Au - f$. Hence, if A is positive definite (SPD), the two problems are equivalent, and there exists a unique solution $u^* = A^{-1} f$.

Motivation and goals

Let A be a real, symmetric, $n \times n$ matrix and let $f \in \mathbf{R}^n$ be given.

Let $\langle \cdot, \cdot \rangle$ denote the standard inner product in $\mathbf{R}^n$.

Consider the following two problems:

1. Solve $Au = f$
2. Minimize the function $\quad J(u) = \frac{1}{2} \langle Au, u \rangle - \langle f, u \rangle$

Note that $\nabla J(u) = Au - f$. Hence, if A is positive definite (SPD), the two problems are equivalent, and there exists a unique solution $u^* = A^{-1} f$.

Many algorithms exist for solving SPD linear systems: Cholesky, Preconditioned Conjugate Gradients, AMG, etc.

Motivation and goals (cont.)

Now we add a set of linear constraints:

Motivation and goals (cont.)

Now we add a set of linear constraints:

Minimize $\quad J(u) = \frac{1}{2}\langle Au, u\rangle - \langle f, u\rangle$

subject to $\quad Bu = g$

where

- A is $n \times n$, symmetric
- B is $m \times n$, with $m < n$
- $f \in \mathbf{R}^n$, $g \in \mathbf{R}^m$ are given (either f or g could be 0, but not both)

Motivation and goals (cont.)

Now we add a set of linear constraints:

Minimize $\quad J(u) = \frac{1}{2}\langle Au, u\rangle - \langle f, u\rangle$

subject to $\quad Bu = g$

where

- A is $n \times n$, symmetric
- B is $m \times n$, with $m < n$
- $f \in \mathbf{R}^n$, $g \in \mathbf{R}^m$ are given (either f or g could be 0, but not both)

Standard approach: Introduce Lagrange multipliers, $p \in \mathbf{R}^m$

- Lagrangian $\mathcal{L}(u, p) = \frac{1}{2}\langle Au, u\rangle - \langle f, u\rangle + \langle p, Bu - g\rangle$
- First-order optimality conditions: $\nabla_u \mathcal{L} = 0$, $\nabla_p \mathcal{L} = 0$

Motivation and goals (cont.)

- Optimality conditions:

$$\nabla_u \mathcal{L} = Au + B^T p - f = 0, \quad \nabla_p \mathcal{L} = Bu - g = 0$$

Motivation and goals (cont.)

- Optimality conditions:

$$\nabla_u \mathcal{L} = Au + B^T p - f = 0, \quad \nabla_p \mathcal{L} = Bu - g = 0$$

or,

$$\begin{pmatrix} A & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \tag{1}$$

Motivation and goals (cont.)

- Optimality conditions:

$$\nabla_u \mathcal{L} = Au + B^T p - f = 0, \quad \nabla_p \mathcal{L} = Bu - g = 0$$

or,

$$\begin{pmatrix} A & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \tag{1}$$

System (1) is a saddle point problem. Its solutions (u^*, p^*) are saddle points for the Lagrangian $\mathcal{L}(u, p)$:

$$\min_u \max_p \mathcal{L}(u, p) = \mathcal{L}(u^*, p^*) = \max_u \min_p \mathcal{L}(u, p)$$

Motivation and goals (cont.)

- Optimality conditions:

$$\nabla_u \mathcal{L} = Au + B^T p - f = 0, \quad \nabla_p \mathcal{L} = Bu - g = 0$$

or,

$$\begin{pmatrix} A & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \tag{1}$$

System (1) is a saddle point problem. Its solutions (u^*, p^*) are saddle points for the Lagrangian $\mathcal{L}(u, p)$:

$$\min_u \max_p \mathcal{L}(u, p) = \mathcal{L}(u^*, p^*) = \max_u \min_p \mathcal{L}(u, p)$$

Also called a KKT system (Karush–Kuhn–Tucker), or equilibrium equations.

Motivation and goals (cont.)

- Optimality conditions:

$$\nabla_u \mathcal{L} = Au + B^T p - f = 0, \quad \nabla_p \mathcal{L} = Bu - g = 0$$

or,

$$\begin{pmatrix} A & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \tag{1}$$

System (1) is a saddle point problem. Its solutions (u^*, p^*) are saddle points for the Lagrangian $\mathcal{L}(u, p)$:

$$\min_u \max_p \mathcal{L}(u, p) = \mathcal{L}(u^*, p^*) = \max_u \min_p \mathcal{L}(u, p)$$

Also called a KKT system (Karush–Kuhn–Tucker), or equilibrium equations.

Gil Strang calls (1) “the fundamental problem of scientific computing.”

Motivation and goals (cont.)

Saddle point problems do occur frequently, e.g.:

Motivation and goals (cont.)

Saddle point problems do occur frequently, e.g.:

- Incompressible flow problems (Stokes, linearized Navier–Stokes)
- Linear elasticity
- Mixed FEM formulations of 2nd- and 4th-order elliptic PDEs
- PDE-constrained optimization (e.g., variational data assimilation)
- SQP and IP methods for nonlinear constrained optimization
- Structural analysis
- Resistive networks, power network analysis
- Image processing (e.g., image registration)

Motivation and goals (cont.)

Saddle point problems do occur frequently, e.g.:

- Incompressible flow problems (Stokes, linearized Navier–Stokes)
- Linear elasticity
- Mixed FEM formulations of 2nd- and 4th-order elliptic PDEs
- PDE-constrained optimization (e.g., variational data assimilation)
- SQP and IP methods for nonlinear constrained optimization
- Structural analysis
- Resistive networks, power network analysis
- Image processing (e.g., image registration)

Comprehensive survey: M. Benzi, G. Golub and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica 14 (2005), pp. 1–137.

Motivation and goals (cont.)

Saddle point problems do occur frequently, e.g.:

- Incompressible flow problems (Stokes, linearized Navier–Stokes)
- Linear elasticity
- Mixed FEM formulations of 2nd- and 4th-order elliptic PDEs
- PDE-constrained optimization (e.g., variational data assimilation)
- SQP and IP methods for nonlinear constrained optimization
- Structural analysis
- Resistive networks, power network analysis
- Image processing (e.g., image registration)

Comprehensive survey: M. Benzi, G. Golub and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica 14 (2005), pp. 1–137.

The bibliography in this paper contains 535 items.

Motivation and goals (cont.)

Saddle point problems do occur frequently, e.g.:

- Incompressible flow problems (Stokes, linearized Navier–Stokes)
- Linear elasticity
- Mixed FEM formulations of 2nd- and 4th-order elliptic PDEs
- PDE-constrained optimization (e.g., variational data assimilation)
- SQP and IP methods for nonlinear constrained optimization
- Structural analysis
- Resistive networks, power network analysis
- Image processing (e.g., image registration)

Comprehensive survey: M. Benzi, G. Golub and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica 14 (2005), pp. 1–137.

The bibliography in this paper contains 535 items.

Google Scholar reports over 700 citations to date.

Motivation and goals (cont.)

The aim of these lectures:

Motivation and goals (cont.)

The aim of these lectures:

- To review the basic properties of saddle point systems

Motivation and goals (cont.)

The aim of these lectures:

- To review the basic properties of saddle point systems
- To give a broad overview of “standard” solution algorithms

Motivation and goals (cont.)

The aim of these lectures:

- To review the basic properties of saddle point systems
- To give a broad overview of “standard” solution algorithms
- To present some recent developments

Motivation and goals (cont.)

The aim of these lectures:

- To review the basic properties of saddle point systems
- To give a broad overview of "standard" solution algorithms
- To present some recent developments

The emphasis of this lecture (and the next) will be on iterative solvers for large, sparse saddle point problems, with a focus on our own recent work on preconditioners for incompressible flow problems.

Motivation and goals (cont.)

The aim of these lectures:

- To review the basic properties of saddle point systems
- To give a broad overview of "standard" solution algorithms
- To present some recent developments

The emphasis of this lecture (and the next) will be on iterative solvers for large, sparse saddle point problems, with a focus on our own recent work on preconditioners for incompressible flow problems.

The ultimate goal: to develop robust preconditioners that perform uniformly well independently of discretization details and problem parameters.

Motivation and goals (cont.)

The aim of these lectures:

- To review the basic properties of saddle point systems
- To give a broad overview of “standard” solution algorithms
- To present some recent developments

The emphasis of this lecture (and the next) will be on iterative solvers for large, sparse saddle point problems, with a focus on our own recent work on preconditioners for incompressible flow problems.

The ultimate goal: to develop robust preconditioners that perform uniformly well independently of discretization details and problem parameters.

For flow problems, we would like to have solvers that converge fast regardless of mesh size, viscosity, etc. Moreover, the cost per iteration should be linear in the number of unknowns.

Outline

Outline

Outline

Outline

Solvability of saddle point problems

The following result establishes necessary and sufficient conditions for the unique solvability of the saddle point problem (1).

Solvability of saddle point problems

The following result establishes necessary and sufficient conditions for the unique solvability of the saddle point problem (1).

Theorem. Assume that

- A is symmetric positive semidefinite $n \times n$
- B has full rank: $\text{rank}\,(B) = m$

Solvability of saddle point problems

The following result establishes necessary and sufficient conditions for the unique solvability of the saddle point problem (1).

Theorem. Assume that

- A is symmetric positive semidefinite $n \times n$
- B has full rank: $\text{rank}\,(B) = m$

Then the coefficient matrix

$$\mathcal{A} = \begin{pmatrix} A & B^T \\ B & O \end{pmatrix}$$

is nonsingular $\Leftrightarrow$ $\text{Null}\,(A) \cap \text{Null}\,(B) = \{0\}$.

Solvability of saddle point problems

The following result establishes necessary and sufficient conditions for the unique solvability of the saddle point problem (1).

Theorem. Assume that

- A is symmetric positive semidefinite $n \times n$
- B has full rank: $\text{rank}\,(B) = m$

Then the coefficient matrix

$$\mathcal{A} = \begin{pmatrix} A & B^T \\ B & O \end{pmatrix}$$

is nonsingular $\Leftrightarrow$ $\text{Null}\,(A) \cap \text{Null}\,(B) = \{0\}$.

Furthermore, $\mathcal{A}$ is indefinite, with n positive and m negative eigenvalues.

Solvability of saddle point problems

The following result establishes necessary and sufficient conditions for the unique solvability of the saddle point problem (1).

Theorem. Assume that

- A is symmetric positive semidefinite $n \times n$
- B has full rank: $\operatorname{rank}(B) = m$

Then the coefficient matrix

$$\mathcal{A} = \begin{pmatrix} A & B^T \\ B & O \end{pmatrix}$$

is nonsingular $\Leftrightarrow$ $\operatorname{Null}(A) \cap \operatorname{Null}(B) = \{0\}$.

Furthermore, $\mathcal{A}$ is indefinite, with n positive and m negative eigenvalues.

In particular, $\mathcal{A}$ is invertible if A is SPD and B has full rank ("standard case").

Generalizations, I

In some cases, a stabilization (or regularization) term needs to be added in the (2,2) position, leading to linear systems of the form

Generalizations, I

In some cases, a stabilization (or regularization) term needs to be added in the (2,2) position, leading to linear systems of the form

$$\begin{pmatrix} A & B^T \\ B & -\beta C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \tag{2}$$

where $\beta > 0$ is a small parameter and the $m \times m$ matrix C is symmetric positive semidefinite, and often singular, with $\|C\|_2 = 1$.

Generalizations, I

In some cases, a stabilization (or regularization) term needs to be added in the (2,2) position, leading to linear systems of the form

$$\begin{pmatrix} A & B^T \\ B & -\beta C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \qquad (2)$$

where $\beta > 0$ is a small parameter and the $m \times m$ matrix C is symmetric positive semidefinite, and often singular, with $\|C\|_2 = 1$.

This type of system arises, for example, from the stabilization of FEM pairs that do not satisfy the LBB ('inf-sup') condition.

Generalizations, I

In some cases, a stabilization (or regularization) term needs to be added in the (2,2) position, leading to linear systems of the form

$$\begin{pmatrix} A & B^T \\ B & -\beta C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \tag{2}$$

where $\beta > 0$ is a small parameter and the $m \times m$ matrix C is symmetric positive semidefinite, and often singular, with $\|C\|_2 = 1$.

This type of system arises, for example, from the stabilization of FEM pairs that do not satisfy the LBB ('inf-sup') condition.

Another important example is the discretization of the Reissner–Mindlin plate model in linear elasticity. In this case β is related to the thickness of the plate; the limit case $\beta = 0$ can be seen as a reformulation of the biharmonic problem.

Generalizations, II

In other cases, the matrix A is not symmetric: $A \neq A^T$. In this case, the saddle point system does not arise from a constrained minimization problem.

Generalizations, II

In other cases, the matrix A is not symmetric: $A \neq A^T$. In this case, the saddle point system does not arise from a constrained minimization problem.

The most important examples of this case are linear systems arising from the Picard and Newton linearizations of the steady incompressible Navier–Stokes equations. The following result is applicable to the Picard linearization (Oseen problem):

Generalizations, II

In other cases, the matrix A is not symmetric: $A \neq A^T$. In this case, the saddle point system does not arise from a constrained minimization problem.

The most important examples of this case are linear systems arising from the Picard and Newton linearizations of the steady incompressible Navier–Stokes equations. The following result is applicable to the Picard linearization (Oseen problem):

Theorem. Assume that

- $H = \frac{1}{2}(A + A^T)$ is symmetric positive semidefinite $n \times n$
- B has full rank: $\text{rank}\,(B) = m$

Generalizations, II

In other cases, the matrix A is not symmetric: $A \neq A^T$. In this case, the saddle point system does not arise from a constrained minimization problem.

The most important examples of this case are linear systems arising from the Picard and Newton linearizations of the steady incompressible Navier–Stokes equations. The following result is applicable to the Picard linearization (Oseen problem):

Theorem. Assume that

- $H = \frac{1}{2}(A + A^T)$ is symmetric positive semidefinite $n \times n$
- B has full rank: $\operatorname{rank}(B) = m$

Then

- $\mathrm{Null}(H) \cap \mathrm{Null}(B) = \{0\} \Rightarrow \mathcal{A}$ invertible
- $\mathcal{A}$ invertible $\Rightarrow \mathrm{Null}(A) \cap \mathrm{Null}(B) = \{0\}$.

Nonsymmetric, positive definite form

Consider the following equivalent formulation:

Nonsymmetric, positive definite form

Consider the following equivalent formulation:

$$\begin{pmatrix} A & B^T \\ -B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ -g \end{pmatrix}$$

Nonsymmetric, positive definite form

Consider the following equivalent formulation:

$$\begin{pmatrix} A & B^T \\ -B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ -g \end{pmatrix}$$

Theorem. Assume B has full rank. If $H = \frac{1}{2}(A + A^T)$ is positive definite, then the spectrum of

$$\mathcal{A}_- := \begin{pmatrix} A & B^T \\ -B & O \end{pmatrix}$$

lies entirely in the open right-half plane $Re(z) > 0$. Moreover, if A is SPD and the following condition holds:

$$\lambda_{\min}(A) > 4\,\lambda_{\max}(S) \quad \text{where } S = BA^{-1}B^T \quad (\text{“Schur complement”}),$$

then $\mathcal{A}_-$ is diagonalizable with real positive eigenvalues.

Nonsymmetric, positive definite form

Consider the following equivalent formulation:

$$\begin{pmatrix} A & B^T \\ -B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ -g \end{pmatrix}$$

Theorem. Assume B has full rank. If $H = \frac{1}{2}(A + A^T)$ is positive definite, then the spectrum of

$$\mathcal{A}_- := \begin{pmatrix} A & B^T \\ -B & O \end{pmatrix}$$

lies entirely in the open right-half plane $Re(z) > 0$. Moreover, if A is SPD and the following condition holds:

$$\lambda_{\min}(A) > 4\,\lambda_{\max}(S) \quad \text{where } S = BA^{-1}B^T \quad (\text{“Schur complement”}),$$

then $\mathcal{A}_-$ is diagonalizable with real positive eigenvalues. In this case, there exists a non-standard inner product on $\mathbf{R}^{n+m}$ in which $\mathcal{A}_-$ is self-adjoint and positive definite, and a corresponding conjugate gradient method (B./Simoncini, NM 2006).

Outline

Example 1: Mixed formulation of the Poisson equation

Let Ω be a domain in $\mathbf{R}^d$ and consider the system

Example 1: Mixed formulation of the Poisson equation

Let Ω be a domain in $\mathbf{R}^d$ and consider the system

$$\mathbf{u} - \nabla p = \mathbf{f} \quad \text{in} \;\; \Omega,$$

Example 1: Mixed formulation of the Poisson equation

Let Ω be a domain in $\mathbf{R}^d$ and consider the system

$$\mathbf{u} - \nabla p = \mathbf{f} \quad \text{in } \Omega,$$

$$\operatorname{div} \mathbf{u} = g \quad \text{in } \Omega,$$

Example 1: Mixed formulation of the Poisson equation

Let Ω be a domain in $\mathbf{R}^d$ and consider the system

$$\mathbf{u} - \nabla p = \mathbf{f} \quad \text{in} \;\; \Omega,$$

$$\operatorname{div} \mathbf{u} = g \quad \text{in} \;\; \Omega,$$

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on} \;\; \partial\Omega.$$

Example 1: Mixed formulation of the Poisson equation

Let Ω be a domain in $\mathbf{R}^d$ and consider the system

$$\mathbf{u} - \nabla p = \mathbf{f} \quad \text{in } \Omega,$$

$$\operatorname{div} \mathbf{u} = g \quad \text{in } \Omega,$$

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega.$$

Eliminating the vector field $\mathbf{u}$, the scalar field p must satisfy

Example 1: Mixed formulation of the Poisson equation

Let Ω be a domain in $\mathbf{R}^d$ and consider the system

$$\mathbf{u} - \nabla p = \mathbf{f} \quad \text{in } \Omega,$$

$$\operatorname{div} \mathbf{u} = g \quad \text{in } \Omega,$$

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega.$$

Eliminating the vector field $\mathbf{u}$, the scalar field p must satisfy

$$-\Delta p = \operatorname{div} \mathbf{f} - g \quad \text{in } \Omega, \qquad \frac{\partial p}{\partial \mathbf{n}} = -\mathbf{f} \cdot \mathbf{n} \quad \text{on } \partial\Omega.$$

Example 1: Mixed formulation of the Poisson equation

Let Ω be a domain in $\mathbf{R}^d$ and consider the system

$$\mathbf{u} - \nabla p = \mathbf{f} \quad \text{in} \ \ \Omega,$$

$$\operatorname{div} \mathbf{u} = g \quad \text{in} \ \ \Omega,$$

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on} \ \ \partial\Omega.$$

Eliminating the vector field $\mathbf{u}$, the scalar field p must satisfy

$$-\Delta p = \operatorname{div} \mathbf{f} - g \quad \text{in} \ \ \Omega, \qquad \frac{\partial p}{\partial \mathbf{n}} = -\mathbf{f} \cdot \mathbf{n} \quad \text{on} \ \ \partial\Omega.$$

Weak formulation: Find $(\mathbf{u}, p) \in (L^2(\Omega))^d \times H^1(\Omega) \cap L^2_0(\Omega)$ such that

Example 1: Mixed formulation of the Poisson equation

Let Ω be a domain in $\mathbf{R}^d$ and consider the system

$$\mathbf{u} - \nabla p = \mathbf{f} \quad \text{in } \Omega,$$

$$\operatorname{div} \mathbf{u} = g \quad \text{in } \Omega,$$

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega.$$

Eliminating the vector field $\mathbf{u}$, the scalar field p must satisfy

$$-\Delta p = \operatorname{div} \mathbf{f} - g \quad \text{in } \Omega, \qquad \frac{\partial p}{\partial \mathbf{n}} = -\mathbf{f} \cdot \mathbf{n} \quad \text{on } \partial\Omega.$$

Weak formulation: Find $(\mathbf{u}, p) \in (L^2(\Omega))^d \times H^1(\Omega) \cap L^2_0(\Omega)$ such that

$$\langle \mathbf{u}, \mathbf{v} \rangle - \langle \nabla p, \mathbf{v} \rangle = \langle \mathbf{f}, \mathbf{v} \rangle, \quad \mathbf{v} \in (L^2(\Omega))^d,$$

Example 1: Mixed formulation of the Poisson equation

Let Ω be a domain in $\mathbf{R}^d$ and consider the system

$$\mathbf{u} - \nabla p = \mathbf{f} \quad \text{in } \Omega,$$

$$\operatorname{div} \mathbf{u} = g \quad \text{in } \Omega,$$

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega.$$

Eliminating the vector field $\mathbf{u}$, the scalar field p must satisfy

$$-\Delta p = \operatorname{div} \mathbf{f} - g \quad \text{in } \Omega, \qquad \frac{\partial p}{\partial \mathbf{n}} = -\mathbf{f} \cdot \mathbf{n} \quad \text{on } \partial\Omega.$$

Weak formulation: Find $(\mathbf{u}, p) \in (L^2(\Omega))^d \times H^1(\Omega) \cap L^2_0(\Omega)$ such that

$$\langle \mathbf{u}, \mathbf{v} \rangle - \langle \nabla p, \mathbf{v} \rangle = \langle \mathbf{f}, \mathbf{v} \rangle, \quad \mathbf{v} \in (L^2(\Omega))^d,$$

$$-\langle \mathbf{u}, \nabla q \rangle = \langle g, q \rangle, \quad q \in H^1(\Omega) \cap L^2_0(\Omega),$$

where $\langle \cdot, \cdot \rangle$ denotes the L^2 inner product.

Example 1: Mixed formulation of the Poisson equation (cont.)

Discretization using LBB-stable finite element pairs leads to an algebraic saddle point problem:

Example 1: Mixed formulation of the Poisson equation (cont.)

Discretization using LBB-stable finite element pairs leads to an algebraic saddle point problem:

$$\begin{pmatrix} M & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$$

Example 1: Mixed formulation of the Poisson equation (cont.)

Discretization using LBB-stable finite element pairs leads to an algebraic saddle point problem:

$$\begin{pmatrix} M & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$$

Here M is a mass matrix, B the discrete divergence, and B^T the discrete (negative) gradient.

Example 1: Mixed formulation of the Poisson equation (cont.)

Discretization using LBB-stable finite element pairs leads to an algebraic saddle point problem:

$$\begin{pmatrix} M & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$$

Here M is a mass matrix, B the discrete divergence, and B^T the discrete (negative) gradient.

For this problem, several optimal preconditioned iterative solvers have been developed.

Example 1: Mixed formulation of the Poisson equation (cont.)

Discretization using LBB-stable finite element pairs leads to an algebraic saddle point problem:

$$\begin{pmatrix} M & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$$

Here M is a mass matrix, B the discrete divergence, and B^T the discrete (negative) gradient.

For this problem, several optimal preconditioned iterative solvers have been developed.

By optimal we mean the following:

Example 1: Mixed formulation of the Poisson equation (cont.)

Discretization using LBB-stable finite element pairs leads to an algebraic saddle point problem:

$$\begin{pmatrix} M & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$$

Here M is a mass matrix, B the discrete divergence, and B^T the discrete (negative) gradient.

For this problem, several optimal preconditioned iterative solvers have been developed.

By optimal we mean the following:

- The convergence rate is independent of discretization parameters

Example 1: Mixed formulation of the Poisson equation (cont.)

Discretization using LBB-stable finite element pairs leads to an algebraic saddle point problem:

$$\begin{pmatrix} M & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$$

Here M is a mass matrix, B the discrete divergence, and B^T the discrete (negative) gradient.

For this problem, several optimal preconditioned iterative solvers have been developed.

By optimal we mean the following:

- The convergence rate is independent of discretization parameters
- The cost of each iteration is linear in the number of unknowns

Example 2: the generalized Stokes problem

Let Ω be a domain in $\mathbf{R}^d$ and let $\alpha \geq 0$, $\nu > 0$. Consider the system

Example 2: the generalized Stokes problem

Let Ω be a domain in $\mathbf{R}^d$ and let $\alpha \geq 0$, $\nu > 0$. Consider the system

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega,$$

Example 2: the generalized Stokes problem

Let Ω be a domain in $\mathbf{R}^d$ and let $\alpha \geq 0$, $\nu > 0$. Consider the system

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega,$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in } \Omega,$$

Example 2: the generalized Stokes problem

Let Ω be a domain in $\mathbf{R}^d$ and let $\alpha \geq 0$, $\nu > 0$. Consider the system

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega,$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in } \Omega,$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \partial\Omega.$$

Example 2: the generalized Stokes problem

Let Ω be a domain in $\mathbf{R}^d$ and let $\alpha \geq 0$, $\nu > 0$. Consider the system

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega,$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in } \Omega,$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \partial\Omega.$$

Weak formulation: Find $(\mathbf{u}, p) \in (H_0^1(\Omega))^d \times L_0^2(\Omega)$ such that

Example 2: the generalized Stokes problem

Let Ω be a domain in $\mathbf{R}^d$ and let $\alpha \geq 0$, $\nu > 0$. Consider the system

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega,$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in } \Omega,$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \partial\Omega.$$

Weak formulation: Find $(\mathbf{u}, p) \in (H_0^1(\Omega))^d \times L_0^2(\Omega)$ such that

$$\alpha\langle \mathbf{u}, \mathbf{v}\rangle + \nu\langle \nabla \mathbf{u}, \nabla \mathbf{v}\rangle - \langle p, \operatorname{div} \mathbf{v}\rangle = \langle \mathbf{f}, \mathbf{v}\rangle, \quad \mathbf{v} \in (H_0^1(\Omega))^d,$$

Example 2: the generalized Stokes problem

Let Ω be a domain in $\mathbf{R}^d$ and let $\alpha \geq 0$, $\nu > 0$. Consider the system

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega,$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in } \Omega,$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \partial\Omega.$$

Weak formulation: Find $(\mathbf{u}, p) \in (H_0^1(\Omega))^d \times L_0^2(\Omega)$ such that

$$\alpha\langle \mathbf{u}, \mathbf{v}\rangle + \nu\langle \nabla \mathbf{u}, \nabla \mathbf{v}\rangle - \langle p, \operatorname{div} \mathbf{v}\rangle = \langle \mathbf{f}, \mathbf{v}\rangle, \quad \mathbf{v} \in (H_0^1(\Omega))^d,$$

$$\langle q, \operatorname{div} \mathbf{u}\rangle = 0, \quad q \in L_0^2(\Omega).$$

Example 2: the generalized Stokes problem

Let Ω be a domain in $\mathbf{R}^d$ and let $\alpha \geq 0$, $\nu > 0$. Consider the system

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega,$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in } \Omega,$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \partial\Omega.$$

Weak formulation: Find $(\mathbf{u}, p) \in (H_0^1(\Omega))^d \times L_0^2(\Omega)$ such that

$$\alpha\langle \mathbf{u}, \mathbf{v}\rangle + \nu\langle \nabla \mathbf{u}, \nabla \mathbf{v}\rangle - \langle p, \operatorname{div} \mathbf{v}\rangle = \langle \mathbf{f}, \mathbf{v}\rangle, \quad \mathbf{v} \in (H_0^1(\Omega))^d,$$

$$\langle q, \operatorname{div} \mathbf{u}\rangle = 0, \quad q \in L_0^2(\Omega).$$

The standard Stokes problem is obtained for $\alpha = 0$ (steady case). In this case we can assume $\nu = 1$.

Example 2: the generalized Stokes problem (cont.)

Discretization using LBB-stable finite element pairs or other div-stable scheme leads to an algebraic saddle point problem:

Example 2: the generalized Stokes problem (cont.)

Discretization using LBB-stable finite element pairs or other div-stable scheme leads to an algebraic saddle point problem:

$$\begin{pmatrix} A & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$$

Example 2: the generalized Stokes problem (cont.)

Discretization using LBB-stable finite element pairs or other div-stable scheme leads to an algebraic saddle point problem:

$$\begin{pmatrix} A & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$$

Here A is a discrete reaction-diffusion operator, B^T the discrete gradient, and B the discrete (negative) divergence. For $\alpha = 0$, A is just the discrete vector Laplacian.

Example 2: the generalized Stokes problem (cont.)

Discretization using LBB-stable finite element pairs or other div-stable scheme leads to an algebraic saddle point problem:

$$\begin{pmatrix} A & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$$

Here A is a discrete reaction-diffusion operator, B^T the discrete gradient, and B the discrete (negative) divergence. For $\alpha = 0$, A is just the discrete vector Laplacian.

If an unstable FEM pair is used, then a regularization term $-\beta C$ is added in the $(2, 2)$ block of $\mathcal{A}$. The specific choice of β and C depends on the particular discretization used.

Example 2: the generalized Stokes problem (cont.)

Discretization using LBB-stable finite element pairs or other div-stable scheme leads to an algebraic saddle point problem:

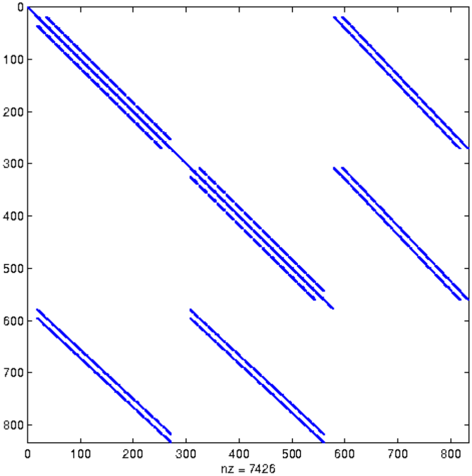
$$\begin{pmatrix} A & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$$

Here A is a discrete reaction-diffusion operator, B^T the discrete gradient, and B the discrete (negative) divergence. For $\alpha = 0$, A is just the discrete vector Laplacian.

If an unstable FEM pair is used, then a regularization term $-\beta C$ is added in the $(2,2)$ block of $\mathcal{A}$. The specific choice of β and C depends on the particular discretization used.

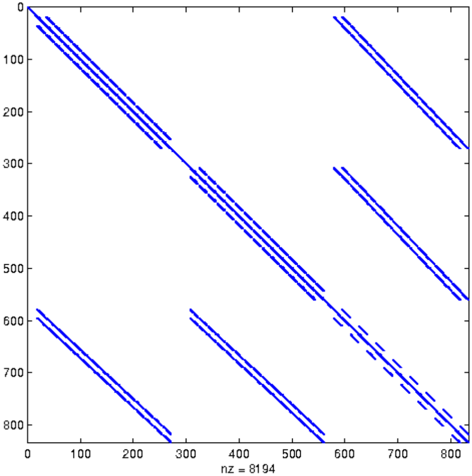
Robust, optimal solvers have been developed for this problem: Cahouet–Chabard for $\alpha > 0$; Silvester–Wathen for $\alpha = 0$.

Sparsity pattern: 2D stokes (Q1-P0)



Without stabilization ($C = O$)

Sparsity pattern: 2D stokes (Q1-P0)



With stabilization ($C \neq O$)

Example 3: the generalized Oseen problem

Let Ω be a domain in $\mathbf{R}^d$ and let $\alpha \geq 0$ and $\nu > 0$. Also, let $\mathbf{w}$ be a divergence-free vector field on Ω. Consider the system

Example 3: the generalized Oseen problem

Let Ω be a domain in $\mathbf{R}^d$ and let $\alpha \geq 0$ and $\nu > 0$. Also, let $\mathbf{w}$ be a divergence-free vector field on Ω. Consider the system

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + (\mathbf{w} \cdot \nabla)\, \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in} \;\; \Omega,$$

Example 3: the generalized Oseen problem

Let Ω be a domain in $\mathbf{R}^d$ and let $\alpha \geq 0$ and $\nu > 0$. Also, let $\mathbf{w}$ be a divergence-free vector field on Ω. Consider the system

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + (\mathbf{w} \cdot \nabla)\, \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega,$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in } \Omega,$$

Example 3: the generalized Oseen problem

Let Ω be a domain in $\mathbf{R}^d$ and let $\alpha \geq 0$ and $\nu > 0$. Also, let $\mathbf{w}$ be a divergence-free vector field on Ω. Consider the system

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + (\mathbf{w} \cdot \nabla)\,\mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \ \Omega,$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in } \ \Omega,$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \ \partial\Omega.$$

Note that for $\mathbf{w} = \mathbf{0}$ we recover the generalized Stokes problem.

Example 3: the generalized Oseen problem

Let Ω be a domain in $\mathbf{R}^d$ and let $\alpha \geq 0$ and $\nu > 0$. Also, let $\mathbf{w}$ be a divergence-free vector field on Ω. Consider the system

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + (\mathbf{w} \cdot \nabla)\, \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \; \Omega,$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in } \; \Omega,$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \; \partial\Omega.$$

Note that for $\mathbf{w} = \mathbf{0}$ we recover the generalized Stokes problem.

Weak formulation: Find $(\mathbf{u}, p) \in (H_0^1(\Omega))^d \times L_0^2(\Omega)$ such that

Example 3: the generalized Oseen problem

Let Ω be a domain in $\mathbf{R}^d$ and let $\alpha \geq 0$ and $\nu > 0$. Also, let $\mathbf{w}$ be a divergence-free vector field on Ω. Consider the system

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + (\mathbf{w} \cdot \nabla)\, \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega,$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in } \Omega,$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \partial\Omega.$$

Note that for $\mathbf{w} = \mathbf{0}$ we recover the generalized Stokes problem.

Weak formulation: Find $(\mathbf{u}, p) \in (H_0^1(\Omega))^d \times L_0^2(\Omega)$ such that

$$\alpha \langle \mathbf{u}, \mathbf{v} \rangle + \nu \langle \nabla \mathbf{u}, \nabla \mathbf{v} \rangle + \langle (\mathbf{w} \cdot \nabla)\, \mathbf{u}, \mathbf{v} \rangle - \langle p, \operatorname{div} \mathbf{v} \rangle = \langle \mathbf{f}, \mathbf{v} \rangle, \quad \mathbf{v} \in (H_0^1(\Omega))^d,$$

Example 3: the generalized Oseen problem

Let Ω be a domain in $\mathbf{R}^d$ and let $\alpha \geq 0$ and $\nu > 0$. Also, let $\mathbf{w}$ be a divergence-free vector field on Ω. Consider the system

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + (\mathbf{w} \cdot \nabla)\, \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega,$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in } \Omega,$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \partial\Omega.$$

Note that for $\mathbf{w} = \mathbf{0}$ we recover the generalized Stokes problem.

Weak formulation: Find $(\mathbf{u}, p) \in (H_0^1(\Omega))^d \times L_0^2(\Omega)$ such that

$$\alpha\langle \mathbf{u}, \mathbf{v}\rangle + \nu\langle \nabla \mathbf{u}, \nabla \mathbf{v}\rangle + \langle (\mathbf{w} \cdot \nabla)\, \mathbf{u}, \mathbf{v}\rangle - \langle p, \operatorname{div} \mathbf{v}\rangle = \langle \mathbf{f}, \mathbf{v}\rangle, \quad \mathbf{v} \in (H_0^1(\Omega))^d,$$

$$\langle q, \operatorname{div} \mathbf{u}\rangle = 0, \quad q \in L_0^2(\Omega).$$

Example 3: the generalized Oseen problem

Let Ω be a domain in $\mathbf{R}^d$ and let $\alpha \geq 0$ and $\nu > 0$. Also, let $\mathbf{w}$ be a divergence-free vector field on Ω. Consider the system

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + (\mathbf{w} \cdot \nabla)\, \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega,$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in } \Omega,$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \partial\Omega.$$

Note that for $\mathbf{w} = \mathbf{0}$ we recover the generalized Stokes problem.

Weak formulation: Find $(\mathbf{u}, p) \in (H_0^1(\Omega))^d \times L_0^2(\Omega)$ such that

$$\alpha\langle \mathbf{u}, \mathbf{v}\rangle + \nu\langle \nabla \mathbf{u}, \nabla \mathbf{v}\rangle + \langle (\mathbf{w} \cdot \nabla)\, \mathbf{u}, \mathbf{v}\rangle - \langle p, \operatorname{div} \mathbf{v}\rangle = \langle \mathbf{f}, \mathbf{v}\rangle, \quad \mathbf{v} \in (H_0^1(\Omega))^d,$$

$$\langle q, \operatorname{div} \mathbf{u}\rangle = 0, \quad q \in L_0^2(\Omega).$$

The standard Oseen problem is obtained for $\alpha = 0$ (steady case).

Example 3: the generalized Oseen problem (cont.)

Discretization using LBB-stable finite element pairs or other div-stable scheme leads to an algebraic saddle point problem:

Example 3: the generalized Oseen problem (cont.)

Discretization using LBB-stable finite element pairs or other div-stable scheme leads to an algebraic saddle point problem:

$$\begin{pmatrix} A & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$$

Example 3: the generalized Oseen problem (cont.)

Discretization using LBB-stable finite element pairs or other div-stable scheme leads to an algebraic saddle point problem:

$$\begin{pmatrix} A & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$$

Now A is a discrete reaction-convection-diffusion operator. For $\alpha = 0$, A is just a discrete vector convection-diffusion operator. Note that now $A \neq A^T$.

Example 3: the generalized Oseen problem (cont.)

Discretization using LBB-stable finite element pairs or other div-stable scheme leads to an algebraic saddle point problem:

$$\begin{pmatrix} A & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$$

Now A is a discrete reaction-convection-diffusion operator. For $\alpha = 0$, A is just a discrete vector convection-diffusion operator. Note that now $A \neq A^T$.

The Oseen problem arises from Picard iteration applied to the steady incompressible Navier–Stokes equations, and from fully implicit schemes applied to the unsteady NSE. The 'wind' $\mathbf{w}$ represents an approximation of the solution $\mathbf{u}$ obtained from the previous Picard step, or from time-lagging.

Example 3: the generalized Oseen problem (cont.)

Discretization using LBB-stable finite element pairs or other div-stable scheme leads to an algebraic saddle point problem:

$$\begin{pmatrix} A & B^T \\ B & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$$

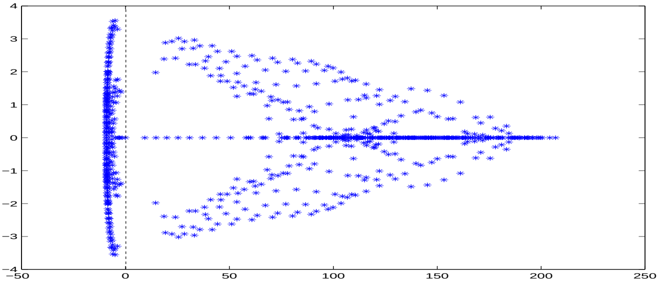
Now A is a discrete reaction-convection-diffusion operator. For $\alpha = 0$, A is just a discrete vector convection-diffusion operator. Note that now $A \neq A^T$.

The Oseen problem arises from Picard iteration applied to the steady incompressible Navier–Stokes equations, and from fully implicit schemes applied to the unsteady NSE. The 'wind' $\mathbf{w}$ represents an approximation of the solution $\mathbf{u}$ obtained from the previous Picard step, or from time-lagging.

As we will see, this problem can be very challenging to solve, especially for small values of the viscosity ν and on stretched meshes.

Eigenvalues of discrete Oseen problem ($\nu = 0.01$), indefinite form

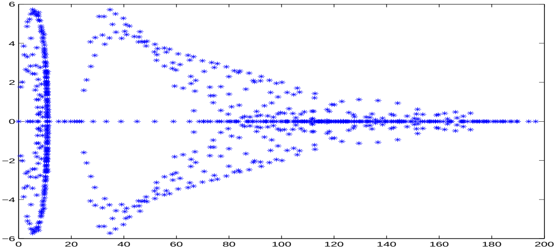
Eigenvalues of Oseen matrix $\mathcal{A} = \begin{pmatrix} A & B^T \\ B & O \end{pmatrix}$, MAC discretization.



Note the different scales in the x and y axes.

Eigenvalues of discrete Oseen problem ($\nu = 0.01$), positive definite form

Eigenvalues of Oseen matrix $\mathcal{A}_- = \begin{pmatrix} A & B^T \\ -B & O \end{pmatrix}$, MAC discretization.



Note the different scales in the x and y axes.

Outline

Overview of available solvers

Two main classes of solvers exist:

1. Direct methods: based on factorization of $\mathcal{A}$

Overview of available solvers

Two main classes of solvers exist:

1. Direct methods: based on factorization of $\mathcal{A}$
    - High-quality software exists (Duff et al.; Demmel et al.)

Overview of available solvers

Two main classes of solvers exist:

1. Direct methods: based on factorization of $\mathcal{A}$
   - High-quality software exists (Duff et al.; Demmel et al.)
   - Quite popular in some areas

Overview of available solvers

Two main classes of solvers exist:

1. Direct methods: based on factorization of $\mathcal{A}$
    - High-quality software exists (Duff et al.; Demmel et al.)
    - Quite popular in some areas
    - Stability issues (indefiniteness)

Overview of available solvers

Two main classes of solvers exist:

1. Direct methods: based on factorization of $\mathcal{A}$
   - High-quality software exists (Duff et al.; Demmel et al.)
   - Quite popular in some areas
   - Stability issues (indefiniteness)
   - Large amounts of fill-in

Overview of available solvers

Two main classes of solvers exist:

1. Direct methods: based on factorization of $\mathcal{A}$
    - High-quality software exists (Duff et al.; Demmel et al.)
    - Quite popular in some areas
    - Stability issues (indefiniteness)
    - Large amounts of fill-in
    - Not feasible for 3D problems

Overview of available solvers

Two main classes of solvers exist:

1. Direct methods: based on factorization of $\mathcal{A}$
   - High-quality software exists (Duff et al.; Demmel et al.)
   - Quite popular in some areas
   - Stability issues (indefiniteness)
   - Large amounts of fill-in
   - Not feasible for 3D problems
   - Difficult to parallelize

Overview of available solvers

Two main classes of solvers exist:

1. Direct methods: based on factorization of $\mathcal{A}$
   - High-quality software exists (Duff et al.; Demmel et al.)
   - Quite popular in some areas
   - Stability issues (indefiniteness)
   - Large amounts of fill-in
   - Not feasible for 3D problems
   - Difficult to parallelize
2. Krylov subspace methods (MINRES, GMRES, Bi-CGSTAB,...)

Overview of available solvers

Two main classes of solvers exist:

1. Direct methods: based on factorization of $\mathcal{A}$
   - High-quality software exists (Duff et al.; Demmel et al.)
   - Quite popular in some areas
   - Stability issues (indefiniteness)
   - Large amounts of fill-in
   - Not feasible for 3D problems
   - Difficult to parallelize
2. Krylov subspace methods (MINRES, GMRES, Bi-CGSTAB,...)
   - Appropriate for large, sparse problems

Overview of available solvers

Two main classes of solvers exist:

1. Direct methods: based on factorization of $\mathcal{A}$
   - High-quality software exists (Duff et al.; Demmel et al.)
   - Quite popular in some areas
   - Stability issues (indefiniteness)
   - Large amounts of fill-in
   - Not feasible for 3D problems
   - Difficult to parallelize
2. Krylov subspace methods (MINRES, GMRES, Bi-CGSTAB,...)
   - Appropriate for large, sparse problems
   - Tend to converge slowly

Overview of available solvers

Two main classes of solvers exist:

1. Direct methods: based on factorization of $\mathcal{A}$
   - High-quality software exists (Duff et al.; Demmel et al.)
   - Quite popular in some areas
   - Stability issues (indefiniteness)
   - Large amounts of fill-in
   - Not feasible for 3D problems
   - Difficult to parallelize
2. Krylov subspace methods (MINRES, GMRES, Bi-CGSTAB,...)
   - Appropriate for large, sparse problems
   - Tend to converge slowly
   - Number of iterations increases as problem size grows

Overview of available solvers

Two main classes of solvers exist:

1. Direct methods: based on factorization of $\mathcal{A}$
   - High-quality software exists (Duff et al.; Demmel et al.)
   - Quite popular in some areas
   - Stability issues (indefiniteness)
   - Large amounts of fill-in
   - Not feasible for 3D problems
   - Difficult to parallelize
2. Krylov subspace methods (MINRES, GMRES, Bi-CGSTAB,...)
   - Appropriate for large, sparse problems
   - Tend to converge slowly
   - Number of iterations increases as problem size grows
   - Effective preconditioners a must

Overview of available solvers

Two main classes of solvers exist:

1. Direct methods: based on factorization of $\mathcal{A}$
   - High-quality software exists (Duff et al.; Demmel et al.)
   - Quite popular in some areas
   - Stability issues (indefiniteness)
   - Large amounts of fill-in
   - Not feasible for 3D problems
   - Difficult to parallelize
2. Krylov subspace methods (MINRES, GMRES, Bi-CGSTAB,...)
   - Appropriate for large, sparse problems
   - Tend to converge slowly
   - Number of iterations increases as problem size grows
   - Effective preconditioners a must

Much effort has been put into developing preconditioners, with optimality and robustness w.r.t. parameters as the ultimate goals. Parallelizability also needs to be taken into account.

Preconditioners

Preconditioning: Find an invertible matrix $\mathcal{P}$ such that Krylov methods applied to the preconditioned system

$$\mathcal{P}^{-1}\mathcal{A}\mathbf{x} = \mathcal{P}^{-1}\mathbf{b}$$

will converge rapidly (possibly, independently of the discretization parameter h).

Preconditioners

Preconditioning: Find an invertible matrix $\mathcal{P}$ such that Krylov methods applied to the preconditioned system

$$\mathcal{P}^{-1}\mathcal{A}\mathbf{x} = \mathcal{P}^{-1}\mathbf{b}$$

will converge rapidly (possibly, independently of the discretization parameter h).

In practice, fast convergence is typically observed when the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ are clustered away from zero. However, it is not an easy matter to characterize the rate of convergence, in general.

Preconditioners

Preconditioning: Find an invertible matrix $\mathcal{P}$ such that Krylov methods applied to the preconditioned system

$$\mathcal{P}^{-1}\mathcal{A}\mathbf{x} = \mathcal{P}^{-1}\mathbf{b}$$

will converge rapidly (possibly, independently of the discretization parameter h).

In practice, fast convergence is typically observed when the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ are clustered away from zero. However, it is not an easy matter to characterize the rate of convergence, in general.

To be effective, a preconditioner must significantly reduce the total amount of work:

Preconditioners

Preconditioning: Find an invertible matrix $\mathcal{P}$ such that Krylov methods applied to the preconditioned system

$$\mathcal{P}^{-1}\mathcal{A}\mathbf{x} = \mathcal{P}^{-1}\mathbf{b}$$

will converge rapidly (possibly, independently of the discretization parameter h).

In practice, fast convergence is typically observed when the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ are clustered away from zero. However, it is not an easy matter to characterize the rate of convergence, in general.

To be effective, a preconditioner must significantly reduce the total amount of work:

- Setting up $\mathcal{P}$ must be inexpensive

Preconditioners

Preconditioning: Find an invertible matrix $\mathcal{P}$ such that Krylov methods applied to the preconditioned system

$$\mathcal{P}^{-1}\mathcal{A}\mathbf{x} = \mathcal{P}^{-1}\mathbf{b}$$

will converge rapidly (possibly, independently of the discretization parameter h).

In practice, fast convergence is typically observed when the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ are clustered away from zero. However, it is not an easy matter to characterize the rate of convergence, in general.

To be effective, a preconditioner must significantly reduce the total amount of work:

- Setting up $\mathcal{P}$ must be inexpensive
- Evaluating $\mathbf{z} = \mathcal{P}^{-1}\mathbf{r}$ must be inexpensive

Preconditioners

Preconditioning: Find an invertible matrix $\mathcal{P}$ such that Krylov methods applied to the preconditioned system

$$\mathcal{P}^{-1}\mathcal{A}\mathbf{x} = \mathcal{P}^{-1}\mathbf{b}$$

will converge rapidly (possibly, independently of the discretization parameter h).

In practice, fast convergence is typically observed when the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ are clustered away from zero. However, it is not an easy matter to characterize the rate of convergence, in general.

To be effective, a preconditioner must significantly reduce the total amount of work:

- Setting up $\mathcal{P}$ must be inexpensive
- Evaluating $\mathbf{z} = \mathcal{P}^{-1}\mathbf{r}$ must be inexpensive
- Convergence must be rapid

Preconditioners

Options include:

Preconditioners

Options include:

1. ILU preconditioners

Preconditioners

Options include:

1. ILU preconditioners

2. Coupled multigrid methods (geometric and algebraic; Vanka-type)

Preconditioners

Options include:

1. ILU preconditioners
2. Coupled multigrid methods (geometric and algebraic; Vanka-type)
3. Schur complement-based methods ('segregated' approach)

Preconditioners

Options include:

1. ILU preconditioners

2. Coupled multigrid methods (geometric and algebraic; Vanka-type)

3. Schur complement-based methods (‘segregated’ approach)
    - Block diagonal preconditioning

Preconditioners

Options include:

1. ILU preconditioners

2. Coupled multigrid methods (geometric and algebraic; Vanka-type)

3. Schur complement-based methods ('segregated' approach)
    - Block diagonal preconditioning
    - Block triangular preconditioning (Elman, Silvester, Wathen et al.)

Preconditioners

Options include:

1. ILU preconditioners

2. Coupled multigrid methods (geometric and algebraic; Vanka-type)

3. Schur complement-based methods ('segregated' approach)
    - Block diagonal preconditioning
    - Block triangular preconditioning (Elman, Silvester, Wathen et al.)
    - SIMPLE and its variants (ur Rehman, Segal, Vuik)

Preconditioners

Options include:

1. ILU preconditioners
2. Coupled multigrid methods (geometric and algebraic; Vanka-type)
3. Schur complement-based methods ('segregated' approach)
   - Block diagonal preconditioning
   - Block triangular preconditioning (Elman, Silvester, Wathen et al.)
   - SIMPLE and its variants (ur Rehman, Segal, Vuik)
4. Constraint preconditioning ('null space methods')

Preconditioners

Options include:

1. ILU preconditioners
2. Coupled multigrid methods (geometric and algebraic; Vanka-type)
3. Schur complement-based methods ('segregated' approach)
   - Block diagonal preconditioning
   - Block triangular preconditioning (Elman, Silvester, Wathen et al.)
   - SIMPLE and its variants (ur Rehman, Segal, Vuik)
4. Constraint preconditioning ('null space methods')
5. Schilders factorization

Preconditioners

Options include:

1. ILU preconditioners
2. Coupled multigrid methods (geometric and algebraic; Vanka-type)
3. Schur complement-based methods ('segregated' approach)
    - Block diagonal preconditioning
    - Block triangular preconditioning (Elman, Silvester, Wathen et al.)
    - SIMPLE and its variants (ur Rehman, Segal, Vuik)
4. Constraint preconditioning ('null space methods')
5. Schilders factorization
6. Augmented Lagrangian-based techniques (AL)

Preconditioners

Options include:

1. ILU preconditioners
2. Coupled multigrid methods (geometric and algebraic; Vanka-type)
3. Schur complement-based methods ('segregated' approach)
   - Block diagonal preconditioning
   - Block triangular preconditioning (Elman, Silvester, Wathen et al.)
   - SIMPLE and its variants (ur Rehman, Segal, Vuik)
4. Constraint preconditioning ('null space methods')
5. Schilders factorization
6. Augmented Lagrangian-based techniques (AL)

The choice of an appropriate preconditioner is highly problem-dependent.

Preconditioners (cont.)

Example: The Silvester–Wathen preconditioner for the Stokes problem is

Preconditioners (cont.)

Example: The Silvester–Wathen preconditioner for the Stokes problem is

$$\mathcal{P} = \begin{pmatrix} \widehat{A} & O \\ O & \widehat{M}_p \end{pmatrix}$$

where $\widehat{A}^{-1}$ is given by a multigrid V-cycle applied to linear systems with coefficient matrix A and $\widehat{M}_p$ is the diagonal of the pressure mass matrix.

This preconditioner is provably optimal:

Preconditioners (cont.)

Example: The Silvester–Wathen preconditioner for the Stokes problem is

$$\mathcal{P} = \begin{pmatrix} \widehat{A} & O \\ O & \widehat{M}_p \end{pmatrix}$$

where $\widehat{A}^{-1}$ is given by a multigrid V-cycle applied to linear systems with coefficient matrix A and $\widehat{M}_p$ is the diagonal of the pressure mass matrix.

This preconditioner is provably optimal:

- MINRES preconditioned with $\mathcal{P}$ converges at a rate independent of the mesh size h

Preconditioners (cont.)

Example: The Silvester–Wathen preconditioner for the Stokes problem is

$$\mathcal{P} = \begin{pmatrix} \widehat{A} & O \\ O & \widehat{M}_p \end{pmatrix}$$

where $\widehat{A}^{-1}$ is given by a multigrid V-cycle applied to linear systems with coefficient matrix A and $\widehat{M}_p$ is the diagonal of the pressure mass matrix.

This preconditioner is provably optimal:

- MINRES preconditioned with $\mathcal{P}$ converges at a rate independent of the mesh size h
- Each preconditioned MINRES iteration costs $O(n+m)$ flops

Preconditioners (cont.)

Example: The Silvester–Wathen preconditioner for the Stokes problem is

$$\mathcal{P} = \begin{pmatrix} \widehat{A} & O \\ O & \widehat{M}_p \end{pmatrix}$$

where $\widehat{A}^{-1}$ is given by a multigrid V-cycle applied to linear systems with coefficient matrix A and $\widehat{M}_p$ is the diagonal of the pressure mass matrix.

This preconditioner is provably optimal:

- MINRES preconditioned with $\mathcal{P}$ converges at a rate independent of the mesh size h
- Each preconditioned MINRES iteration costs $O(n+m)$ flops
- Efficient parallelization is possible

Example: Silvester-Wathen preconditioner

Number of MINRES iterations needed for 10^{-6} reduction in residual for locally stabilized $Q1 - P0$ mixed finite elements for Stokes flow in a cavity. Block diagonal preconditioner: $\widehat{A}$ is one multigrid V-cycle with 1,1 relaxed Jacobi smoothing and $\widehat{M}_p$ is the diagonal pressure mass matrix. The cpu time (in seconds) is on a Sun sparcv9 502 MHz processor with 1024 Mb of memory. The cpu time is also given for a sparse direct solve (UMFPACK in MATLAB).

grid	n	m	iterations	cpu time	sparse direct cpu
64×64	8450	4096	38	14.3	6.8
128×128	33282	16384	37	37.7	48.0
256×256	132098	65536	36	194.6	897
512×512	526339	263169	35	6903	out of memory

Note: the lack of scalability for finer meshes is caused by memory management issues.

But what about more difficult problems?

Block preconditioners

If A is invertible, $\mathcal{A}$ has the block LU factorization

Block preconditioners

If A is invertible, $\mathcal{A}$ has the block LU factorization

$$\mathcal{A} = \begin{pmatrix} A & B^T \\ B & O \end{pmatrix} = \begin{pmatrix} I & O \\ BA^{-1} & I \end{pmatrix} \begin{pmatrix} A & B^T \\ O & S \end{pmatrix},$$

where $S = -BA^{-1}B^T$ (Schur complement).

Block preconditioners

If A is invertible, $\mathcal{A}$ has the block LU factorization

$$\mathcal{A} = \begin{pmatrix} A & B^T \\ B & O \end{pmatrix} = \begin{pmatrix} I & O \\ BA^{-1} & I \end{pmatrix} \begin{pmatrix} A & B^T \\ O & S \end{pmatrix},$$

where $S = -BA^{-1}B^T$ (Schur complement).
Let

$$\mathcal{P}_D = \begin{pmatrix} A & O \\ O & S \end{pmatrix}, \quad \mathcal{P}_T = \begin{pmatrix} A & B^T \\ O & S \end{pmatrix},$$

then

- The spectrum of $\mathcal{P}_D^{-1}\mathcal{A}$ is $\sigma(\mathcal{P}_D^{-1}\mathcal{A}) = \left\{1, \frac{1\pm\sqrt{5}}{2}\right\}$

Block preconditioners

If A is invertible, $\mathcal{A}$ has the block LU factorization

$$\mathcal{A} = \begin{pmatrix} A & B^T \\ B & O \end{pmatrix} = \begin{pmatrix} I & O \\ BA^{-1} & I \end{pmatrix} \begin{pmatrix} A & B^T \\ O & S \end{pmatrix},$$

where $S = -BA^{-1}B^T$ (Schur complement).
Let

$$\mathcal{P}_D = \begin{pmatrix} A & O \\ O & S \end{pmatrix}, \quad \mathcal{P}_T = \begin{pmatrix} A & B^T \\ O & S \end{pmatrix},$$

then

- The spectrum of $\mathcal{P}_D^{-1}\mathcal{A}$ is $\sigma(\mathcal{P}_D^{-1}\mathcal{A}) = \left\{1, \frac{1\pm\sqrt{5}}{2}\right\}$
- The spectrum of $\mathcal{P}_T^{-1}\mathcal{A}$ is $\sigma(\mathcal{P}_T^{-1}\mathcal{A}) = \{1\}$

Block preconditioners

If A is invertible, $\mathcal{A}$ has the block LU factorization

$$\mathcal{A} = \begin{pmatrix} A & B^T \\ B & O \end{pmatrix} = \begin{pmatrix} I & O \\ BA^{-1} & I \end{pmatrix} \begin{pmatrix} A & B^T \\ O & S \end{pmatrix},$$

where $S = -BA^{-1}B^T$ (Schur complement).
Let

$$\mathcal{P}_D = \begin{pmatrix} A & O \\ O & S \end{pmatrix}, \quad \mathcal{P}_T = \begin{pmatrix} A & B^T \\ O & S \end{pmatrix},$$

then

- The spectrum of $\mathcal{P}_D^{-1}\mathcal{A}$ is $\sigma(\mathcal{P}_D^{-1}\mathcal{A}) = \left\{1, \frac{1\pm\sqrt{5}}{2}\right\}$
- The spectrum of $\mathcal{P}_T^{-1}\mathcal{A}$ is $\sigma(\mathcal{P}_T^{-1}\mathcal{A}) = \{1\}$
- GMRES converges in three iterations with $\mathcal{P}_D$, and in two iterations with $\mathcal{P}_T$.

Block preconditioners (cont.)

In practice, it is necessary to replace A and S with easily invertible approximations:

Block preconditioners (cont.)

In practice, it is necessary to replace A and S with easily invertible approximations:

$$\mathcal{P}_D = \begin{pmatrix} \widehat{A} & O \\ O & \widehat{S} \end{pmatrix}, \quad \mathcal{P}_T = \begin{pmatrix} \widehat{A} & B^T \\ O & \widehat{S} \end{pmatrix}$$

Block preconditioners (cont.)

In practice, it is necessary to replace A and S with easily invertible approximations:

$$\mathcal{P}_D = \begin{pmatrix} \widehat{A} & O \\ O & \widehat{S} \end{pmatrix}, \quad \mathcal{P}_T = \begin{pmatrix} \widehat{A} & B^T \\ O & \widehat{S} \end{pmatrix}$$

- $\widehat{A}$ should be spectrally equivalent to A: that is, we want $\mathrm{cond}(\widehat{A}^{-1}A) \leq c$ for some constant c independent of h

Block preconditioners (cont.)

In practice, it is necessary to replace A and S with easily invertible approximations:

$$\mathcal{P}_D = \begin{pmatrix} \widehat{A} & O \\ O & \widehat{S} \end{pmatrix}, \quad \mathcal{P}_T = \begin{pmatrix} \widehat{A} & B^T \\ O & \widehat{S} \end{pmatrix}$$

- $\widehat{A}$ should be spectrally equivalent to A: that is, we want $\mathrm{cond}(\widehat{A}^{-1}A) \leq c$ for some constant c independent of h
- Often a small, fixed number of multigrid V-cycles will do

Block preconditioners (cont.)

In practice, it is necessary to replace A and S with easily invertible approximations:

$$\mathcal{P}_D = \begin{pmatrix} \widehat{A} & O \\ O & \widehat{S} \end{pmatrix}, \quad \mathcal{P}_T = \begin{pmatrix} \widehat{A} & B^T \\ O & \widehat{S} \end{pmatrix}$$

- $\widehat{A}$ should be spectrally equivalent to A: that is, we want $\mathrm{cond}(\widehat{A}^{-1}A) \leq c$ for some constant c independent of h
- Often a small, fixed number of multigrid V-cycles will do
- Approximating S is more involved, except in special situations; for example, in the case of Stokes we can use the pressure mass matrix ($\widehat{S} = M_p$) or its diagonal, assuming the LBB condition holds. This is the Silvester–Wathen preconditioner.

Block preconditioners (cont.)

In practice, it is necessary to replace A and S with easily invertible approximations:

$$\mathcal{P}_D = \begin{pmatrix} \widehat{A} & O \\ O & \widehat{S} \end{pmatrix}, \quad \mathcal{P}_T = \begin{pmatrix} \widehat{A} & B^T \\ O & \widehat{S} \end{pmatrix}$$

- $\widehat{A}$ should be spectrally equivalent to A: that is, we want $\mathrm{cond}(\widehat{A}^{-1}A) \leq c$ for some constant c independent of h
- Often a small, fixed number of multigrid V-cycles will do
- Approximating S is more involved, except in special situations; for example, in the case of Stokes we can use the pressure mass matrix ($\widehat{S} = M_p$) or its diagonal, assuming the LBB condition holds. This is the Silvester–Wathen preconditioner.
- For the Oseen problem this does not work, except for very small Reynolds.

Block preconditioners (cont.)

Recall that $S = -BA^{-1}B^T$ is a discretization of the operator

$$\mathcal{S} = \mathrm{div}(-\nu\Delta + \mathbf{w} \cdot \nabla)^{-1}\nabla$$

Block preconditioners (cont.)

Recall that $S = -BA^{-1}B^T$ is a discretization of the operator

$$\mathcal{S} = \mathrm{div}(-\nu\Delta + \mathbf{w}\cdot\nabla)^{-1}\nabla$$

A plausible (if non-rigorous) approximation of the inverse of this operator is

$$\widehat{\mathcal{S}}^{-1} := \Delta^{-1}(-\nu\Delta + \mathbf{w}\cdot\nabla)_p$$

where the subscript p indicated that the convection-diffusion operator acts on the pressure space. Hence, the action of S^{-1} can be approximated by a matrix-vector multiply with a discrete pressure convection-diffusion operator, followed by a Poisson solve.

Block preconditioners (cont.)

Recall that $S = -BA^{-1}B^T$ is a discretization of the operator

$$\mathcal{S} = \text{div}(-\nu\Delta + \mathbf{w}\cdot\nabla)^{-1}\nabla$$

A plausible (if non-rigorous) approximation of the inverse of this operator is

$$\widehat{\mathcal{S}}^{-1} := \Delta^{-1}(-\nu\Delta + \mathbf{w}\cdot\nabla)_p$$

where the subscript p indicated that the convection-diffusion operator acts on the pressure space. Hence, the action of S^{-1} can be approximated by a matrix-vector multiply with a discrete pressure convection-diffusion operator, followed by a Poisson solve.

This is known as the pressure convection-diffusion preconditioner (PCD), introduced and analyzed by Kay, Loghin, and Wathen (SISC, 2001).

Block preconditioners (cont.)

Recall that $S = -BA^{-1}B^T$ is a discretization of the operator

$$\mathcal{S} = \text{div}(-\nu\Delta + \mathbf{w}\cdot\nabla)^{-1}\nabla$$

A plausible (if non-rigorous) approximation of the inverse of this operator is

$$\widehat{\mathcal{S}}^{-1} := \Delta^{-1}(-\nu\Delta + \mathbf{w}\cdot\nabla)_p$$

where the subscript p indicated that the convection-diffusion operator acts on the pressure space. Hence, the action of S^{-1} can be approximated by a matrix-vector multiply with a discrete pressure convection-diffusion operator, followed by a Poisson solve.

This is known as the pressure convection-diffusion preconditioner (PCD), introduced and analyzed by Kay, Loghin, and Wathen (SISC, 2001).

This preconditioner performs well for small or moderate Reynolds numbers.

Results for Kay, Loghin and Wathen preconditioner

Test problems: steady Oseen, homogeneous Dirichlet BCs, two choices of the wind function.

Results for Kay, Loghin and Wathen preconditioner

Test problems: steady Oseen, homogeneous Dirichlet BCs, two choices of the wind function.

- A constant wind problem: $\mathbf{w} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

Results for Kay, Loghin and Wathen preconditioner

Test problems: steady Oseen, homogeneous Dirichlet BCs, two choices of the wind function.

- A constant wind problem: $\mathbf{w} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$
- A recirculating flow (vortex) problem: $\mathbf{w} = \begin{pmatrix} 4(2y-1)(1-x)x \\ -4(2x-1)(1-y)y \end{pmatrix}$

Results for Kay, Loghin and Wathen preconditioner

Test problems: steady Oseen, homogeneous Dirichlet BCs, two choices of the wind function.

- A constant wind problem: $\mathbf{w} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$
- A recirculating flow (vortex) problem: $\mathbf{w} = \begin{pmatrix} 4(2y-1)(1-x)x \\ -4(2x-1)(1-y)y \end{pmatrix}$

Uniform FEM discretizations: isoP2-P0 and isoP2-P1. These discretizations satisfy the inf-sup condition: no pressure stabilization is needed.
SUPG stabilization is used for the velocities.

Results for Kay, Loghin and Wathen preconditioner

Test problems: steady Oseen, homogeneous Dirichlet BCs, two choices of the wind function.

- A constant wind problem: $\mathbf{w} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$
- A recirculating flow (vortex) problem: $\mathbf{w} = \begin{pmatrix} 4(2y-1)(1-x)x \\ -4(2x-1)(1-y)y \end{pmatrix}$

Uniform FEM discretizations: isoP2-P0 and isoP2-P1. These discretizations satisfy the inf-sup condition: no pressure stabilization is needed.
SUPG stabilization is used for the velocities.

The Krylov subspace method used is Bi-CGSTAB. This method requires two matrix-vector multiplies with $\mathcal{A}$ and two applications of the preconditioner at each iteration.

Results for Kay, Loghin and Wathen preconditioner

Test problems: steady Oseen, homogeneous Dirichlet BCs, two choices of the wind function.

- A constant wind problem: $\mathbf{w} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$
- A recirculating flow (vortex) problem: $\mathbf{w} = \begin{pmatrix} 4(2y-1)(1-x)x \\ -4(2x-1)(1-y)y \end{pmatrix}$

Uniform FEM discretizations: isoP2-P0 and isoP2-P1. These discretizations satisfy the inf-sup condition: no pressure stabilization is needed.
SUPG stabilization is used for the velocities.

The Krylov subspace method used is Bi-CGSTAB. This method requires two matrix-vector multiplies with $\mathcal{A}$ and two applications of the preconditioner at each iteration.

A preconditioning step requires two convection-diffusion solves (three in 3D) and one Poisson solve at each iteration, plus some mat-vecs.

Results for Kay, Loghin, and Wathen preconditioner (cont.)

Results for KLW preconditioner.

mesh size h	viscosity ν				
	1	0.1	0.01	0.001	0.0001
constant wind					
1/16	6 / 12	8 / 16	12 / 24	30 / 34	100 / 80
1/32	6 / 10	10 / 16	14 / 24	24 / 28	86 / 92
1/64	6 / 10	8 / 14	16 / 24	22 / 32	64 / 66
1/128	6 / 10	8 / 12	16 / 26	24 / 36	64 / 58
rotating vortex					
1/16	6 / 8	10 / 12	30 / 40	$>$ 400 / 188	
1/32	6 / 8	10 / 12	30 / 40	$>$ 400 / 378	
1/64	4 / 6	8 / 12	26 / 40	$>$ 400 / $>$ 400	
1/128	4 / 6	8 / 10	22 / 44	228 / $>$ 400	

Number of Bi-CGSTAB iterations
(**Note**: exact solves used throughout. Stopping criterion: $\|\mathbf{b} - \mathcal{A}\mathbf{x}_k\|_2 < 10^{-6}\|\mathbf{b}\|_2$).

Results with Vanka-type MG preconditioner

Results for Vanka-MG-BiCGStab approach, isoP2-P0 FEM.

mesh size h	viscosity $= \nu$				
	1	0.1	0.01	10^{-3}	10^{-4}
constant wind					
$1/16$	4	4	3	4	5
$1/32$	4	4	3	4	4
$1/64$	4	4	4	3	5
$1/128$	4	4	4	3	4
rotating vortex					
$1/16$	4	5	5	8	14
$1/32$	4	4	6	9	17
$1/64$	4	4	5	8	40
$1/128$	4	4	4	9	> 400

Better than KLW preconditioner, but still not robust for low viscosity. Can we do better than this?

References

1. M. Benzi, G. H. Golub and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1–137.
2. H. C. Elman, D. J. Silvester and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Oxford University Press, 2005.
3. K. A. Mardal and R. Winther, *Preconditioning discretizations of systems of partial differential equations*, Numer. Linear Algebra Appl., 18 (2011), pp. 1–40.