Acknowledgments

- NSF (Computational Mathematics)
- Maxim Olshanskii (Mech-Math, Moscow State U.)
- Zhen Wang (PhD student, Emory)
- Thanks also to Valeria Simoncini (U. of Bologna, Italy)
Let $A$ be a real, symmetric, $n \times n$ matrix and let $f \in \mathbb{R}^n$ be given.

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

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 $f \in \mathbb{R}^n$ be given.

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

Consider the following two problems:

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

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 $u^* = A^{-1}f$. 
Let $A$ be a real, symmetric, $n \times n$ matrix and let $f \in \mathbb{R}^n$ be given.

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

Consider the following two problems:

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

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.
Now we add a set of linear constraints:
Motivation and goals (cont.)

Now we add a set of linear constraints:

Minimize $J(u) = \frac{1}{2} \langle Au, u \rangle - \langle f, u \rangle$
subject to $Bu = g$

where

- $A$ is $n \times n$, symmetric
- $B$ is $m \times n$, with $m < n$
- $f \in \mathbb{R}^n$, $g \in \mathbb{R}^m$ 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 \[ J(u) = \frac{1}{2} \langle Au, u \rangle - \langle f, u \rangle \]

subject to \[ Bu = g \]

where

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

Standard approach: Introduce Lagrange multipliers, \( p \in \mathbb{R}^m \)

- 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: \( \nabla_u \mathcal{L} = 0 \), \( \nabla_p \mathcal{L} = 0 \)
Optimality conditions:

\[ \nabla_u \mathcal{L} = Au + B^T p - f = 0, \quad \nabla_p \mathcal{L} = Bu - g = 0 \]
Optimality conditions:

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

or,

\[
\begin{pmatrix}
A & B^T \\
B & 0
\end{pmatrix}
\begin{pmatrix}
u \\
p
\end{pmatrix}
=
\begin{pmatrix}
f \\
g
\end{pmatrix}
\]  

(1)
Motivation and goals (cont.)

- Optimality conditions:

\[ \nabla_u L = Au + B^T p - f = 0, \quad \nabla_p L = Bu - g = 0 \]

or,

\[
\begin{pmatrix}
A & B^T
\end{pmatrix}
\begin{pmatrix}
u \\
p
\end{pmatrix}
= 
\begin{pmatrix}
f \\
g
\end{pmatrix}
\]

(1)

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

\[
\min_u \max_p L(u, p) = L(u^*, p^*) = \max_u \min_p L(u, p)
\]
Optimality conditions:

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

or,

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

(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:
  \[ \nabla_u L = Au + B^T p - f = 0, \quad \nabla_p L = Bu - g = 0 \]

  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 \(L(u, p)\):

\[
\min_u \max_p L(u, p) = L(u^*, p^*) = \max_u \min_p L(u, p)
\]

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

Gil Strang calls (1) “the fundamental problem of scientific computing.”
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)
- Mixed FEM formulations of 2nd- and 4th-order elliptic PDEs
- Time-harmonic Maxwell equations
- PDE-constrained optimization (e.g., variational data assimilation)
- SQP and IP methods for nonlinear constrained optimization
- Structural analysis
- Resistive networks, power network analysis
- Certain economic models
Saddle point problems do occur frequently, e.g.:

- **Incompressible flow problems** (Stokes, linearized Navier–Stokes)
- Mixed FEM formulations of 2nd- and 4th-order elliptic PDEs
- Time-harmonic Maxwell equations
- PDE-constrained optimization (e.g., variational data assimilation)
- SQP and IP methods for nonlinear constrained optimization
- Structural analysis
- Resistive networks, power network analysis
- Certain economic models

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

- **Incompressible flow problems** (Stokes, linearized Navier–Stokes)
- Mixed FEM formulations of 2nd- and 4th-order elliptic PDEs
- Time-harmonic Maxwell equations
- PDE-constrained optimization (e.g., variational data assimilation)
- SQP and IP methods for nonlinear constrained optimization
- Structural analysis
- Resistive networks, power network analysis
- Certain economic models


The bibliography in this paper contains 535 items.
Saddle point problems do occur frequently, e.g.:

- **Incompressible flow problems** (Stokes, linearized Navier–Stokes)
- Mixed FEM formulations of 2nd- and 4th-order elliptic PDEs
- Time-harmonic Maxwell equations
- PDE-constrained optimization (e.g., variational data assimilation)
- SQP and IP methods for nonlinear constrained optimization
- Structural analysis
- Resistive networks, power network analysis
- Certain economic models


The bibliography in this paper contains 535 items.

*Google Scholar* reports 480 citations to date.
The aim of this talk:

- Briefly review the basic properties of saddle point systems.
- Give an overview of solution algorithms for specific problems.
- Point out some current challenges and recent developments.

The emphasis of the talk will be on preconditioned 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.
Motivation and goals (cont.)

The aim of this talk:

- To briefly review the basic properties of saddle point systems
The aim of this talk:

- To briefly review the basic properties of saddle point systems
- To give an overview of solution algorithms for specific problems

The emphasis of the talk will be on preconditioned 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.
The aim of this talk:

- To briefly review the basic properties of saddle point systems
- To give an overview of solution algorithms for specific problems
- To point out some current challenges and recent developments

The emphasis of the talk will be on preconditioned 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.
The aim of this talk:

- To briefly review the basic properties of saddle point systems
- To give an overview of solution algorithms for specific problems
- To point out some current challenges and recent developments

The emphasis of the talk will be on **preconditioned iterative solvers** for large, sparse saddle point problems, with a focus on our own recent work on preconditioners for **incompressible flow problems**.
The aim of this talk:

- To briefly review the basic properties of saddle point systems
- To give an overview of solution algorithms for specific problems
- To point out some current challenges and recent developments

The emphasis of the talk will be on preconditioned 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.
The aim of this talk:

- To briefly review the basic properties of saddle point systems
- To give an overview of solution algorithms for specific problems
- To point out some current challenges and recent developments

The emphasis of the talk will be on preconditioned 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

1. Properties of saddle point matrices
Outline

1. Properties of saddle point matrices
2. Examples: Incompressible flow problems
Outline

1. Properties of saddle point matrices
2. Examples: Incompressible flow problems
3. Some solution algorithms
Outline

1. Properties of saddle point matrices
2. Examples: Incompressible flow problems
3. Some solution algorithms
4. The Augmented Lagrangian (AL) approach
Fast Iterative Solution of Saddle Point Problems

Outline

1. Properties of saddle point matrices
2. Examples: Incompressible flow problems
3. Some solution algorithms
4. The Augmented Lagrangian (AL) approach
5. The modified Augmented Lagrangian-based preconditioner
Outline

1. Properties of saddle point matrices
2. Examples: Incompressible flow problems
3. Some solution algorithms
4. The Augmented Lagrangian (AL) approach
5. The modified Augmented Lagrangian-based preconditioner
6. Conclusions
Outline

1. Properties of saddle point matrices
2. Examples: Incompressible flow problems
3. Some solution algorithms
4. The Augmented Lagrangian (AL) approach
5. The modified Augmented Lagrangian-based preconditioner
6. Conclusions
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: $\text{rank} (B) = m$

Furthermore, $A$ is indefinite, with $n$ positive and $m$ negative eigenvalues. In particular, $A$ is invertible if $A$ is SPD and $B$ has full rank ("standard case").
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 $\iff \text{Null}(A) \cap \text{Null}(B) = \{0\}$.

Furthermore, $\mathcal{A}$ is indefinite, with $n$ positive and $m$ negative eigenvalues.
The following result establishes necessary and sufficient conditions for the \textbf{unique solvability} of the saddle point problem (1).

**Theorem.** Assume that

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

Then the coefficient matrix

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

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

Furthermore, $\mathcal{A}$ is \textbf{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").
In some cases, a stabilization (or regularization) term needs to be added in the (2,2) position, leading to linear systems of the form
In some cases, a stabilization (or regularization) term needs to be added in the (2,2) position, leading to linear systems of the form

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

where $\beta > 0$ is a small parameter and the $m \times m$ matrix $C$ is symmetric positive semidefinite, and often singular, with $\|C\|_2 = 1.$
In some cases, a stabilization (or regularization) term needs to be added in the (2,2) position, leading to linear systems of the form

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

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.
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 \( \beta \) is related to the thickness of the plate; the limit case \( \beta = 0 \) can be seen as a reformulation of the biharmonic problem.
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.
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):
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: $\text{rank}(B) = m$
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: $\text{rank}(B) = m$

Then

- $\text{Null}(H) \cap \text{Null}(B) = \{0\} \Rightarrow \mathcal{A}$ invertible
- $\mathcal{A}$ invertible $\Rightarrow \text{Null}(A) \cap \text{Null}(B) = \{0\}$. 
Nonsymmetric, positive definite form

Consider the following equivalent formulation:
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:

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

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

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

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

1 Properties of saddle point matrices

2 Examples: Incompressible flow problems

3 Some solution algorithms

4 The Augmented Lagrangian (AL) approach

5 The modified Augmented Lagrangian-based preconditioner

6 Conclusions
Example 1: the generalized Stokes problem

Let $\Omega$ be a domain in $\mathbb{R}^d$ and let $\alpha \geq 0$, $\nu > 0$. Consider the system
Example 1: the generalized Stokes problem

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

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

where \( \langle \cdot, \cdot \rangle \) denotes the \( L^2 \) inner product.
Example 1: the generalized Stokes problem

Let \( \Omega \) be a domain in \( \mathbb{R}^d \) and let \( \alpha \geq 0, \nu > 0 \). Consider the system

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

\[
\text{div} \ \mathbf{u} = 0 \quad \text{in} \quad \Omega,
\]
Example 1: the generalized Stokes problem

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

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = f \quad \text{in } \Omega,$$
$$\text{div} \mathbf{u} = 0 \quad \text{in } \Omega,$$
$$\mathbf{u} = 0 \quad \text{on } \partial \Omega.$$
Example 1: the generalized Stokes problem

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

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

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

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

**Weak formulation:** Find $(\mathbf{u}, p) \in (H^1_0(\Omega))^d \times L^2_0(\Omega)$ such that
Example 1: the generalized Stokes problem

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

$$\alpha u - \nu \Delta u + \nabla p = f \quad \text{in } \Omega,$$
$$\text{div } u = 0 \quad \text{in } \Omega,$$
$$u = 0 \quad \text{on } \partial \Omega.$$

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

$$\alpha \langle u, v \rangle + \langle \nabla u, \nabla v \rangle - \langle p, \text{div } v \rangle = \langle f, v \rangle, \quad v \in (H^1_0(\Omega))^d,$$
Example 1: the generalized Stokes problem

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

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

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

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

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

$$\alpha \langle u, v \rangle + \langle \nabla u, \nabla v \rangle - \langle p, \text{div} \, v \rangle = \langle f, v \rangle, \quad v \in (H^1_0(\Omega))^d,$$

$$\langle q, \text{div} \, u \rangle = 0, \quad q \in L^2_0(\Omega),$$

where $\langle \cdot, \cdot \rangle$ denotes the $L^2$ inner product.
Example 1: the generalized Stokes problem

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

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

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

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

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

$$
\alpha \langle \mathbf{u}, \mathbf{v} \rangle + \langle \nabla \mathbf{u}, \nabla \mathbf{v} \rangle - \langle p, \text{div} \mathbf{v} \rangle = \langle \mathbf{f}, \mathbf{v} \rangle, \quad \mathbf{v} \in (H^1_0(\Omega))^d,
$$

$$
\langle q, \text{div} \mathbf{u} \rangle = 0, \quad q \in L^2_0(\Omega),
$$

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

The standard Stokes problem is obtained for $\alpha = 0$ (steady case). In this case we can assume $\nu = 1$. 
Discretization using LBB-stable finite element pairs or other div-stable scheme leads to an algebraic saddle point problem:
Example 1: 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 1: 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.
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 \(A\). The specific choice of \(\beta\) and \(C\) depends on the particular discretization used.
Example 1: 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 \(A\). The specific choice of \(\beta\) and \(C\) depends on the particular discretization used.

Robust, optimal solvers have been developed for this problem: \textbf{Cahouet–Chabard} for \(\alpha > 0\); \textbf{Silvester–Wathen} for \(\alpha = 0\).
Sparsity pattern: 2D stokes (Q1-P0)

Without stabilization ($C = O$)
Fast Iterative Solution of Saddle Point Problems
Examples: Incompressible flow problems

Sparsity pattern: 2D stokes (Q1-P0)

With stabilization ($C \neq O$)
Example 2: the generalized Oseen problem

Let $\Omega$ be a domain in $\mathbb{R}^d$ and let $\alpha \geq 0$ and $\nu > 0$. Also, let $w$ be a divergence-free vector field on $\Omega$. Consider the system
Example 2: the generalized Oseen problem

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

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

and

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

$$u = 0 \quad \text{on } \partial \Omega.$$
Example 2: the generalized Oseen problem

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

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

$$
div u = 0 \quad \text{in } \Omega,
$$
Example 2: the generalized Oseen problem

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

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

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

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

Note that for $w = 0$ we recover the generalized Stokes problem.
Example 2: the generalized Oseen problem

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

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

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

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

Note that for $w = 0$ we recover the generalized Stokes problem.

Weak formulation: Find $(u, p) \in (H^1_0(\Omega))^d \times L^2(\Omega)$ such that
Example 2: the generalized Oseen problem

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

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

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

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

Note that for $w = 0$ we recover the generalized Stokes problem.

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

$$\alpha \langle u, v \rangle + \nu \langle \nabla u, \nabla v \rangle + \langle (w \cdot \nabla) u, v \rangle - \langle p, \text{div } v \rangle = \langle f, v \rangle, \quad v \in (H^1_0(\Omega))^d,$$
Let $\Omega$ be a domain in $\mathbb{R}^d$ and let $\alpha \geq 0$ and $\nu > 0$. Also, let $w$ be a divergence-free vector field on $\Omega$. Consider the system

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

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

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

Note that for $w = 0$ we recover the generalized Stokes problem.

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

$$\alpha \langle u, v \rangle + \nu \langle \nabla u, \nabla v \rangle + \langle (w \cdot \nabla) u, v \rangle - \langle p, \text{div } v \rangle = \langle f, v \rangle, \quad v \in (H^1_0(\Omega))^d,$$

$$\langle q, \text{div } u \rangle = 0, \quad q \in L^2_0(\Omega).$$
Example 2: the generalized Oseen problem

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

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

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

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

Note that for $w = 0$ we recover the generalized Stokes problem.

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

$$\alpha \langle u, v \rangle + \nu \langle \nabla u, \nabla v \rangle + \langle (w \cdot \nabla) u, v \rangle - \langle p, \text{div} \ v \rangle = \langle f, v \rangle, \quad v \in (H^1_0(\Omega))^d,$$

$$\langle q, \text{div} \ u \rangle = 0, \quad q \in L^2_0(\Omega).$$

The standard Oseen problem is obtained for $\alpha = 0$ (steady case).
Example 2: 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 2: the generalized Oseen problem (cont.)

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

\[
\begin{pmatrix}
A & B^T \\
B & O
\end{pmatrix}
\begin{pmatrix}
u \\
p
\end{pmatrix}
=
\begin{pmatrix}
f \\
0
\end{pmatrix}
\]
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 2: 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’ \(w\) represents an approximation of the solution \(u\) obtained from the previous Picard step, or from time-lagging.
Example 2: 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’ $w$ represents an approximation of the solution $u$ 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 $\nu$ and on stretched meshes.
Eigenvalues of discrete Oseen problem ($\nu = 0.01$), indefinite form

$$\mathbb{A} = \begin{pmatrix} A & B^T \\ B & O \end{pmatrix}, \text{ MAC discretization.}$$

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 $A_- = \begin{pmatrix} A & B^T \\ -B & O \end{pmatrix}$, MAC discretization.

Note the different scales in the x and y axes.
Fast Iterative Solution of Saddle Point Problems
Some solution algorithms

Outline

1. Properties of saddle point matrices
2. Examples: Incompressible flow problems
3. Some solution algorithms
   - The Augmented Lagrangian (AL) approach
   - The modified Augmented Lagrangian-based preconditioner
4. Conclusions
Overview of available solvers

Two main classes of solvers exist:

- **Direct methods**: based on factorization of $A$

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

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.
Overview of available solvers

Two main classes of solvers exist:

- **Direct methods**: based on factorization of $A$
  - High-quality software exists (Duff et al.; Demmel et al.)
Overview of available solvers

Two main classes of solvers exist:

- **Direct methods**: based on factorization of $A$
  - 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 $A$
   - High-quality software exists (Duff et al.; Demmel et al.)
   - Quite popular in some areas
   - Stability issues (indefiniteness)
Two main classes of solvers exist:

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

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.
Overview of available solvers

Two main classes of solvers exist:

1. **Direct methods**: based on factorization of $A$
   - 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 $A$
   - 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 are 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.
Overview of available solvers

Two main classes of solvers exist:

1. **Direct methods**: based on factorization of \( A \)
   - 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 $A$
   - 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 $A$
   - High-quality software exists (Duff et al.; Demmel et al.)
   - Quite popular in some areas
   - Stability issues (indefinite)
   - 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 $A$
   - 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
Two main classes of solvers exist:

1. **Direct methods:** based on factorization of $A$
   - 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 $A$
   - 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*

$$\mathcal{P}^{-1}Ax = \mathcal{P}^{-1}b$$

will converge rapidly (possibly, independently of the discretization parameter $h$).
Preconditioning: Find an invertible matrix $\mathcal{P}$ such that Krylov methods applied to the preconditioned system

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

will converge rapidly (possibly, independently of the discretization parameter $h$). In practice, fast convergence is typically observed when the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}A$ 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

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

will converge rapidly (possibly, independently of the discretization parameter $h$). In practice, fast convergence is typically observed when the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}A$ 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

$$\mathcal{P}^{-1}Ax = \mathcal{P}^{-1}b$$

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

In practice, fast convergence is typically observed when the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}A$ 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 $P$ such that Krylov methods applied to the *preconditioned system*

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

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

In practice, **fast convergence** is typically observed when the eigenvalues of the preconditioned matrix $P^{-1}A$ 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 $P$ must be inexpensive
- Evaluating $\mathbf{z} = P^{-1}\mathbf{r}$ must be inexpensive
Preconditioners

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

$$\mathcal{P}^{-1}Ax = \mathcal{P}^{-1}b$$

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

In practice, fast convergence is typically observed when the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}A$ 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 $z = \mathcal{P}^{-1}r$ must be inexpensive
- Convergence must be rapid
Preconditioners

Options include:

1. ILU preconditioners
2. Coupled multigrid methods (geometric and algebraic; Vanka-type)
3. Schur complement-based methods ('segregated' approach)
4. Block diagonal preconditioning
5. Block triangular preconditioning (Elman et al.)
6. Uzawa, SIMPLE,...
7. Constraint preconditioning ('null space methods')
8. Augmented Lagrangian-based techniques (AL)

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

Options include:

- ILU preconditioners
Preconditioners

Options include:

1. ILU preconditioners

2. Coupled multigrid methods (geometric and algebraic; Vanka-type)
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 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 et al.)
   - Uzawa, SIMPLE,...
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 et al.)
   - Uzawa, SIMPLE,...

4. **Constraint preconditioning** (‘null space methods’)

The choice of an appropriate preconditioner is highly problem-dependent.
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 et al.)
   - Uzawa, SIMPLE,...

4. Constraint preconditioning (‘null space methods’)

5. 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 et al.)
   - Uzawa, SIMPLE,...

4. Constraint preconditioning (‘null space methods’)

5. 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

\[ P = b A O O b M p \]

where \( b A^{-1} \) is given by a multigrid V-cycle applied to linear systems with coefficient matrix \( A \) and \( b M p \) is the diagonal of the pressure mass matrix. This preconditioner is provably optimal: MINRES preconditioned with \( P \) converges at a rate independent of the mesh size \( h \). Each preconditioned MINRES iteration costs \( O(n + m) \) flops. Efficient parallelization is possible. But what about more difficult problems?
Example: The Silvester–Wathen preconditioner for the Stokes problem is

\[ P = \begin{pmatrix} \hat{A} & O \\ O & \hat{M}_p \end{pmatrix} \]

where \( \hat{A}^{-1} \) is given by a multigrid V-cycle applied to linear systems with coefficient matrix \( A \) and \( \hat{M}_p \) 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

\[ P = \begin{pmatrix} \hat{A} & O \\ O & \hat{M}_p \end{pmatrix} \]

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

This preconditioner is **provably optimal**:

- MINRES preconditioned with \( P \) converges at a rate independent of the mesh size \( h \).
Example: The Silvester–Wathen preconditioner for the Stokes problem is

\[ P = \begin{pmatrix} \hat{A} & O \\ O & \hat{M}_p \end{pmatrix} \]

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

This preconditioner is provably optimal:

- MINRES preconditioned with \( P \) converges at a rate independent of the mesh size \( h \)
- Each preconditioned MINRES iteration costs \( O(n + m) \) flops
Example: The Silvester–Wathen preconditioner for the Stokes problem is

\[ P = \begin{pmatrix} \hat{A} & O \\ O & \hat{M}_p \end{pmatrix} \]

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

This preconditioner is provably optimal:

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

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

\[ \mathcal{P} = \begin{pmatrix} \hat{A} & O \\ O & \hat{M}_p \end{pmatrix} \]

where \( \hat{A}^{-1} \) is given by a multigrid V-cycle applied to linear systems with coefficient matrix \( A \) and \( \hat{M}_p \) 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

But what about more difficult problems?
Block preconditioners

If $A$ is invertible, $A$ has the block LU factorization

$$A = \begin{bmatrix} A & B \\ B^T & O \end{bmatrix} = \begin{bmatrix} I & O \\ BA^{-1} & I \end{bmatrix} ^{-1} \begin{bmatrix} A & B \\ B^T & O \end{bmatrix} = \begin{bmatrix} -BA^{-1} & B \\ B^T & -I \end{bmatrix} (\text{Schur complement}).$$

Let $P = \begin{bmatrix} A & O \\ O & S \end{bmatrix}$, $P^T = \begin{bmatrix} A & B \\ B^T & O \end{bmatrix}$, then

$$\sigma(P^{-1}D) = \left\{ \begin{array}{c} 1, \\
1 \pm \sqrt{5}^{-1} \end{array} \right\}$$

GMRES converges in three iterations with $P^{-1}D$, and in two iterations with $P^{-1}T$.
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, $A$ has the block LU factorization

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

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

Let

$$P_D = \begin{pmatrix} A & O \\ O & S \end{pmatrix}, \quad P_T = \begin{pmatrix} A & B^T \\ O & S \end{pmatrix},$$

then

- The spectrum of $P_D^{-1}A$ is $\sigma(P_D^{-1}A) = \left\{ 1, \frac{1±\sqrt{5}}{2} \right\}$
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, $A$ has the block LU factorization

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

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

Let

$$
P_D = \begin{pmatrix} A & O \\ O & S \end{pmatrix}, \quad P_T = \begin{pmatrix} A & B^T \\ O & S \end{pmatrix},
$$

then

- The spectrum of $P_D^{-1}A$ is $\sigma(P_D^{-1}A) = \left\{ 1, \frac{1 + \sqrt{5}}{2} \right\}$
- The spectrum of $P_T^{-1}A$ is $\sigma(P_T^{-1}A) = \{1\}$
- GMRES converges in three iterations with $P_D$, and in two iterations with $P_T$. 
In practice, it is necessary to replace $A$ and $S$ with easily invertible approximations:
In practice, it is necessary to replace $A$ and $S$ with easily invertible approximations:

$$P_D = \begin{pmatrix} \hat{A} & O \\ O & \hat{S} \end{pmatrix}, \quad P_T = \begin{pmatrix} \hat{A} & B^T \\ O & \hat{S} \end{pmatrix}$$
Block preconditioners (cont.)

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

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

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


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

$$
\mathcal{P}_D = \begin{pmatrix} \hat{A} & 0 \\ O & \hat{S} \end{pmatrix}, \quad \mathcal{P}_T = \begin{pmatrix} \hat{A} & B^T \\ O & \hat{S} \end{pmatrix}
$$

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

- Often a small, fixed number of multigrid V-cycles will do
In practice, it is necessary to replace $A$ and $S$ with easily invertible approximations:

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

- $\hat{A}$ should be spectrally equivalent to $A$: that is, we want $\text{cond}(\hat{A}^{-1}A) \leq c$ 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 ($\hat{S} = M_p$) 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:

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

- $\hat{A}$ should be spectrally equivalent to $A$: that is, we want $\text{cond}(\hat{A}^{-1}A) \leq c$ 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 ($\hat{S} = M_p$) 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.
Recall that $S = -BA^{-1}B^T$ is a discretization of the operator $S = \text{div}(-\nu \Delta + \mathbf{w} \cdot \nabla)^{-1}\nabla$
Recall that $S = -BA^{-1}B^T$ is a discretization of the operator

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

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

$$\hat{S}^{-1} := \Delta^{-1}(-\nu\Delta + \mathbf{w} \cdot \nabla)\rho$$

where the subscript $\rho$ 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.
Recall that $S = -BA^{-1}B^T$ is a discretization of the operator

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

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

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

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

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

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

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

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.
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: \( w = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \)
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.
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.
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 $A$ 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.)

<table>
<thead>
<tr>
<th>mesh size $h$</th>
<th>viscosity $\nu$</th>
<th>1</th>
<th>0.1</th>
<th>0.01</th>
<th>0.001</th>
<th>0.0001</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>constant wind</strong></td>
<td></td>
<td>6 / 12</td>
<td>8 / 16</td>
<td>12 / 24</td>
<td>30 / 34</td>
<td>100 / 80</td>
</tr>
<tr>
<td>1/16</td>
<td></td>
<td>6 / 10</td>
<td>10 / 16</td>
<td>14 / 24</td>
<td>24 / 28</td>
<td>86 / 92</td>
</tr>
<tr>
<td>1/32</td>
<td></td>
<td>6 / 10</td>
<td>8 / 14</td>
<td>16 / 24</td>
<td>22 / 32</td>
<td>64 / 66</td>
</tr>
<tr>
<td>1/64</td>
<td></td>
<td>6 / 10</td>
<td>8 / 12</td>
<td>16 / 26</td>
<td>24 / 36</td>
<td>64 / 58</td>
</tr>
<tr>
<td>1/128</td>
<td></td>
<td>6 / 10</td>
<td>8 / 12</td>
<td>16 / 26</td>
<td>24 / 36</td>
<td>64 / 58</td>
</tr>
<tr>
<td><strong>rotating vortex</strong></td>
<td></td>
<td>6 / 8</td>
<td>10 / 12</td>
<td>30 / 40</td>
<td>&gt; 400 / 188</td>
<td></td>
</tr>
<tr>
<td>1/16</td>
<td></td>
<td>6 / 8</td>
<td>10 / 12</td>
<td>30 / 40</td>
<td>&gt; 400 / 378</td>
<td></td>
</tr>
<tr>
<td>1/32</td>
<td></td>
<td>4 / 6</td>
<td>8 / 12</td>
<td>26 / 40</td>
<td>&gt; 400 / &gt; 400</td>
<td></td>
</tr>
<tr>
<td>1/64</td>
<td></td>
<td>4 / 6</td>
<td>8 / 10</td>
<td>22 / 44</td>
<td>228 / &gt; 400</td>
<td></td>
</tr>
<tr>
<td>1/128</td>
<td></td>
<td>4 / 6</td>
<td>8 / 10</td>
<td>22 / 44</td>
<td>228 / &gt; 400</td>
<td></td>
</tr>
</tbody>
</table>

Number of Bi-CGSTAB iterations

*Note:* exact solves used throughout. Stopping criterion: $\|b - Ax_k\|_2 < 10^{-6} ||b||_2$.
Outline

1. Properties of saddle point matrices
2. Examples: Incompressible flow problems
3. Some solution algorithms
4. The Augmented Lagrangian (AL) approach
5. The modified Augmented Lagrangian-based preconditioner
6. Conclusions
Consider the equivalent augmented Lagrangian formulation (Fortin, Glowinski, 1982) given by

\[
\begin{pmatrix}
A + \gamma B^T W^{-1} B & B^T \\
B & O
\end{pmatrix}
\begin{pmatrix}
u \\
p
\end{pmatrix}
= \begin{pmatrix}
f + \gamma B^T W^{-1} g \\
g
\end{pmatrix},
\]

where \( \gamma > 0 \) and \( W \) is symmetric positive definite.
Consider the equivalent augmented Lagrangian formulation (Fortin, Glowinski, 1982) given by

\[
\begin{pmatrix}
A + \gamma B^T W^{-1} B & B^T \\
B & O
\end{pmatrix}
\begin{pmatrix}
u \\
p
\end{pmatrix} = 
\begin{pmatrix}
f + \gamma B^T W^{-1} g \\
g
\end{pmatrix},
\]

where \( \gamma > 0 \) and \( W \) is symmetric positive definite.

Letting \( A_\gamma := A + \gamma B^T W^{-1} B \) and \( f_\gamma := f + \gamma B^T W^{-1} g \),

\[
\begin{pmatrix}
A_\gamma & B^T \\
B & O
\end{pmatrix}
\begin{pmatrix}
u \\
p
\end{pmatrix} = 
\begin{pmatrix}
f_\gamma \\
g
\end{pmatrix}, \quad \text{or} \quad \hat{A} \hat{x} = \hat{b}.
\]

(4)
Consider the equivalent augmented Lagrangian formulation (Fortin, Glowinski, 1982) given by

\[
\begin{pmatrix}
A + \gamma B^T W^{-1} B & B^T \\
B & O
\end{pmatrix}
\begin{pmatrix}
u \\
p
\end{pmatrix} = 
\begin{pmatrix}
f + \gamma B^T W^{-1} g \\
g
\end{pmatrix},
\] (3)

where \( \gamma > 0 \) and \( W \) is symmetric positive definite.

Letting \( A_\gamma := A + \gamma B^T W^{-1} B \) and \( f_\gamma := f + \gamma B^T W^{-1} g \),

\[
\begin{pmatrix}
A_\gamma & B^T \\
B & O
\end{pmatrix}
\begin{pmatrix}
u \\
p
\end{pmatrix} = 
\begin{pmatrix}
f_\gamma \\
g
\end{pmatrix}, \quad \text{or} \quad \tilde{A} x = \hat{b}.
\] (4)

B. and Olshanskii introduced the following block preconditioner for (4)

\[
\mathcal{P} = 
\begin{pmatrix}
A_\gamma & B^T \\
O & \hat{S}
\end{pmatrix}, \quad \hat{S}^{-1} = -\nu \tilde{M}_p^{-1} - \gamma W^{-1}.
\] (5)
Analysis for the Oseen problem

Theorem (B./Olshanskii, SISC 2006)

Setting $W = M_p$, the preconditioned matrix $P^{-1} \hat{A}$ has the eigenvalue 1 of multiplicity $n$; the remaining $m$ eigenvalues are contained in a rectangle in the right half plane with sides independent of the mesh size $h$, and bounded away from 0. Moreover, for $\gamma = O(\nu^{-1})$ the rectangle does not depend on $\nu$. When $\gamma \rightarrow \infty$, all the eigenvalues tend to 1.
Theorem (B./Olshanskii, SISC 2006)

Setting $W = M_p$, the preconditioned matrix $\mathcal{P}^{-1} \hat{A}$ has the eigenvalue 1 of multiplicity $n$; the remaining $m$ eigenvalues are contained in a rectangle in the right half plane with sides independent of the mesh size $h$, and bounded away from 0. Moreover, for $\gamma = O(\nu^{-1})$ the rectangle does not depend on $\nu$. When $\gamma \to \infty$, all the eigenvalues tend to 1.
Analysis for the Oseen problem (cont.)

Using field of values analysis, we can prove the following stronger parameter-independent convergence result for preconditioned GMRES:

\[ \frac{\|b - Ax_k\|}{\|b - Ax_0\|} \leq q^k \]

where \( q < 1 \) is independent of problem parameters \( h, \nu \) and \( \alpha \).
Using **field of values** analysis, we can prove the following stronger parameter-independent convergence result for preconditioned GMRES:

**Theorem (B./Olshanskii, SINUM 2011)**

For $\nu < 1$, if $\gamma = \|(BA^{-1}B^T)^{-1}M_p\|_M$ the residual norms in GMRES with the original AL preconditioner satisfy

$$\|\hat{b} - \hat{A}x_k\| \leq q^k\|\hat{b} - \hat{A}x_0\|,$$

where $q < 1$ is independent of problem parameters $h, \nu$ and $\alpha$. 
Using field of values analysis, we can prove the following stronger parameter-independent convergence result for preconditioned GMRES:

**Theorem (B./Olshanskii, SINUM 2011)**

For $\nu < 1$, if $\gamma = \|(BA^{-1}B^T)^{-1}M_p\|_M$ the residual norms in GMRES with the original AL preconditioner satisfy

$$\|\hat{b} - \hat{A}x_k\| \leq q^k \|\hat{b} - \hat{A}x_0\|,$$

where $q < 1$ is independent of problem parameters $h$, $\nu$ and $\alpha$.

Recall that the field of values of an $n \times n$ matrix $B$ is the subset of $\mathbb{C}$ defined by

$$\mathcal{F}(B) := \{x^*Bx \mid x \in \mathbb{C}^n, x^*x = 1\}.$$

We proved that $\mathcal{F}(P^{-1}A)$ is bounded and bounded away from 0 for all $h$, $\nu$ and $\alpha$. This implies the above convergence result for GMRES.
Practical considerations

Applying $P^{-1}$ to a vector requires one solve with $A_{\gamma}$ and one with $\hat{S}$. 
Practical considerations

Applying $P^{-1}$ to a vector requires one solve with $A_\gamma$ and one with $\hat{S}$.

- In practice we use $W = \hat{M}_p = \text{diag}(M_p)$
Practical considerations

Applying $P^{-1}$ to a vector requires one solve with $A_\gamma$ and one with $\tilde{S}$.

- In practice we use $W = \tilde{M}_p = \text{diag}(M_p)$

- The solve with $A_\gamma$ can be approximated by a suitable geometric multigrid method for elliptic systems (similar to the one by Schöberl, NM 1999)
Practical considerations

Applying $\mathcal{P}^{-1}$ to a vector requires one solve with $A_\gamma$ and one with $\widehat{S}$.

- In practice we use $W = \widehat{M}_p = \text{diag}(M_p)$

- The solve with $A_\gamma$ can be approximated by a suitable geometric multigrid method for elliptic systems (similar to the one by Schöberl, NM 1999)

- For $\widehat{S}^{-1}$, a few Richardson iterations preconditioned with the diagonal of $M_p$ can be used to solve the linear system with $M_p$. 

Applying $\mathcal{P}^{-1}$ to a vector requires one solve with $A_\gamma$ and one with $\hat{S}$.

- In practice we use $W = \hat{M}_p = \text{diag}(M_p)$

- The solve with $A_\gamma$ can be approximated by a suitable geometric multigrid method for elliptic systems (similar to the one by Schöberl, NM 1999)

- For $\hat{S}^{-1}$, a few Richardson iterations preconditioned with the diagonal of $M_p$ can be used to solve the linear system with $M_p$.

Though the previous theorems suggest that $\gamma = O(\nu^{-1})$, in practice, $\gamma = O(1)$ is sufficient for (near) parameter-independent convergence.
Numerical results

**Table:** Bi-CGSTAB iterations (isoP2-P0 FEM, SUPG, $\gamma = 1$)

<table>
<thead>
<tr>
<th>Viscosity</th>
<th>1</th>
<th>0.1</th>
<th>0.01</th>
<th>0.001</th>
<th>0.0001</th>
</tr>
</thead>
<tbody>
<tr>
<td>mesh size</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/16</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>1/32</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>1/64</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>1/128</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>mesh size</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/16</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>1/32</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>10</td>
<td>21</td>
</tr>
<tr>
<td>1/64</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>9</td>
<td>18</td>
</tr>
<tr>
<td>1/128</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>14</td>
</tr>
</tbody>
</table>

The rate of convergence of Krylov subspace method with this preconditioner is nearly optimal:

- Independent of the grid; almost independent of viscosity
- Cost is $O(n + m)$ per iteration
- Similar results with isoP2-P1 FEM
Outline

1. Properties of saddle point matrices
2. Examples: Incompressible flow problems
3. Some solution algorithms
4. The Augmented Lagrangian (AL) approach
5. The modified Augmented Lagrangian-based preconditioner
6. Conclusions
Motivation: Circumvent sophisticated geometric multigrid techniques in $A_\gamma^{-1}$, so as to be able to handle unstructured grids and more complex geometries.
The modified augmented Lagrangian-based preconditioner

**Motivation:** Circumvent sophisticated geometric multigrid techniques in $A^{-1}$, so as to be able to handle unstructured grids and more complex geometries. From $A = \text{diag}(A_1, A_2)$ and $B = (B_1, B_2)$, we find

$$A_γ = A + γB^TW^{-1}B = \begin{pmatrix} A_1 + γB_1^TW^{-1}B_1 & γB_1^TW^{-1}B_2 \\ γB_2^TW^{-1}B_1 & A_2 + γB_2^TW^{-1}B_2 \end{pmatrix} := \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}. $$
Motivation: Circumvent sophisticated geometric multigrid techniques in $A_\gamma^{-1}$, so as to be able to handle unstructured grids and more complex geometries. From $A = \text{diag}(A_1, A_2)$ and $B = (B_1, B_2)$, we find

$$A_\gamma = A + \gamma B^T W^{-1} B = \begin{pmatrix} A_1 + \gamma B_1^T W^{-1} B_1 & \gamma B_1^T W^{-1} B_2 \\ \gamma B_2^T W^{-1} B_1 & A_2 + \gamma B_2^T W^{-1} B_2 \end{pmatrix} := \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}.$$ 

The modified AL preconditioner (B., Olshanskii and Wang, IJNMF 2010) is defined as

$$\tilde{P} = \begin{pmatrix} \tilde{A}_\gamma & B^T \\ O & \tilde{S} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & B_1^T \\ O & A_{22} & B_2^T \end{pmatrix}.$$  \hfill (6)
Motivation: Circumvent sophisticated geometric multigrid techniques in $A^{-1}_{\gamma}$, so as to be able to handle unstructured grids and more complex geometries. From $A = \text{diag}(A_1, A_2)$ and $B = (B_1, B_2)$, we find

$$A_{\gamma} = A + \gamma B^T W^{-1} B = \begin{pmatrix} A_1 + \gamma B_1^T W^{-1} B_1 & \gamma B_1^T W^{-1} B_2 \\ \gamma B_2^T W^{-1} B_1 & A_2 + \gamma B_2^T W^{-1} B_2 \end{pmatrix} := \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}.$$

The modified AL preconditioner (B., Olshanskii and Wang, IJNMF 2010) is defined as

$$\tilde{P} = \begin{pmatrix} \tilde{A}_{\gamma} & B^T \\ O & \tilde{S} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & B_1^T \\ O & A_{22} & B_2^T \\ O & O & \tilde{S} \end{pmatrix}$$

$$(6)$$

- $A_{ii} = A_i + \gamma B_i^T W^{-1} B_i$  ($i = 1, 2$) can be interpreted as discrete scalar anisotropic convection-diffusion operators with anisotropy ratio $\approx 1 + \frac{\gamma}{\nu}$.
The modified augmented Lagrangian-based preconditioner

**Motivation:** Circumvent sophisticated geometric multigrid techniques in $A_{\gamma}^{-1}$, so as to be able to handle unstructured grids and more complex geometries. From $A = \text{diag}(A_1, A_2)$ and $B = (B_1, B_2)$, we find

$$A_{\gamma} = A + \gamma B^T W^{-1} B = \begin{pmatrix} A_1 + \gamma B_1^T W^{-1} B_1 & \gamma B_1^T W^{-1} B_2 \\ \gamma B_2^T W^{-1} B_1 & A_2 + \gamma B_2^T W^{-1} B_2 \end{pmatrix} := \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}. $$

The modified AL preconditioner (B., Olshanskii and Wang, IJNMF 2010) is defined as

$$\tilde{\mathcal{P}} = \begin{pmatrix} \tilde{A}_{\gamma} & B^T \\ O & \tilde{S} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & B_1^T \\ O & A_{22} & B_2^T \\ O & O & \tilde{S} \end{pmatrix}$$ (6)

- $A_{ii} = A_i + \gamma B_i^T W^{-1} B_i$ ($i = 1, 2$) can be interpreted as discrete scalar anisotropic convection-diffusion operators with anisotropy ratio $\approx 1 + \frac{\gamma}{\nu}$.
- They can be solved with standard algebraic multigrid (AMG) methods, in particular, parallel AMG solvers.
Note that

$$\hat{\mathcal{A}} \hat{\mathcal{P}}^{-1} = \begin{pmatrix} I_{n/2} & O \\ \ast & I_{n/2} - D \end{pmatrix} \begin{pmatrix} O \\ E \end{pmatrix}.$$  

The eigenvalues of $\hat{\mathcal{A}} \hat{\mathcal{P}}^{-1}$ are $\lambda = 1$ of multiplicity $n/2$, plus the eigenvalues of

$$\begin{pmatrix} I_{n/2} - D & E \\ F & I_m - G \end{pmatrix} = I_{n/2+m} - \begin{pmatrix} D & -E \\ -F & G \end{pmatrix}.$$  

In general, the multiplicity of $\lambda = 1$ is only $n/2$.  

Analysis

Note that

$$\widehat{AP}^{-1} = \begin{pmatrix}
\frac{I_{n/2}}{*} & O & O \\
* & \frac{I_{n/2}}{F} - D & E \\
* & F & I_m - G
\end{pmatrix}.$$

The eigenvalues of $\widehat{AP}^{-1}$ are $\lambda = 1$ of multiplicity $n/2$, plus the eigenvalues of

$$\left( \frac{I_{n/2}}{F} - D \quad E \right) = I_{n/2+m} - \begin{pmatrix} D & -E \\ -F & G \end{pmatrix}.$$

In general, the multiplicity of $\lambda = 1$ is only $n/2$.
However, letting $\hat{S}^{-1} = -\gamma W^{-1}$, the matrix on the right-hand side is rank deficient by $n/2$, so $\widehat{AP}^{-1}$ has the eigenvalue $\lambda = 1$ of multiplicity at least $n$.

Using field of values analysis, we can prove that the convergence rate of GMRES with the modified AL preconditioner is $h$-independent, with a moderate dependence on $\nu$. 
The choice of $\gamma$

The value of the augmentation parameter $\gamma$ is determined by local Fourier analysis (LFA):
The choice of $\gamma$

The value of the augmentation parameter $\gamma$ is determined by local Fourier analysis (LFA):

1. ‘discretize’ the diffusion and (frozen) convection terms in $A$ by centered differences, assuming periodic BCs;

Note that $\gamma$ only depends on $h$ and $\nu$. Hence it can be pre-computed, so no overhead is imposed. The discretizations are symbolic. Details in B. and Wang (SISC, 2011).
The choice of $\gamma$

The value of the augmentation parameter $\gamma$ is determined by local Fourier analysis (LFA):

1. ‘discretize’ the diffusion and (frozen) convection terms in $A$ by centered differences, assuming periodic BCs;
2. ‘discretize’ the gradient and divergence by one-sided differences;
The choice of $\gamma$

The value of the augmentation parameter $\gamma$ is determined by local Fourier analysis (LFA):

1. ‘discretize’ the diffusion and (frozen) convection terms in $A$ by centered differences, assuming periodic BCs;
2. ‘discretize’ the gradient and divergence by one-sided differences;
3. note that $W = \hat{M}_p$ scales as $h^2$;
The choice of $\gamma$

The value of the augmentation parameter $\gamma$ is determined by local Fourier analysis (LFA):

1. 'discretize' the diffusion and (frozen) convection terms in $A$ by centered differences, assuming periodic BCs;
2. 'discretize' the gradient and divergence by one-sided differences;
3. note that $W = \hat{M}_p$ scales as $h^2$;
4. express $\widetilde{P}$ and $\hat{A}$ in terms of "Fourier eigenvalues", and find the $\gamma$ that minimizes the average distance of the non-unit eigenvalues $\lambda(\gamma)$ of the preconditioned matrix $\hat{A}\widetilde{P}^{-1}$ from 1.

Note that $\gamma$ only depends on $h$ and $\nu$. Hence it can be pre-computed, so no overhead is imposed. The discretizations are symbolic.

Details in B. and Wang (SISC, 2011).
Example: a regularized lid driven cavity problem

All 2D experiments are done using IFISS package (Elman, Silvester & Ramage).

In the lid driven cavity problem, the flow is enclosed in a square with $u_1 = 1 - x^4$, $u_2 = 0$ on the top to represent the moving lid.

**Figure:** Regularized lid driven cavity (Q2-Q1, $\nu = 0.001$, stretched $128 \times 128$ grid)
Iteration counts for the lid driven cavity problem

Table: GMRES iterations with modified AL preconditioner (cavity, Q2-Q1, uniform grids)

<table>
<thead>
<tr>
<th>Viscosity</th>
<th>0.1</th>
<th>0.01</th>
<th>0.005</th>
<th>0.001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid</td>
<td>LFA</td>
<td>Opt</td>
<td>LFA</td>
<td>Opt</td>
</tr>
<tr>
<td>16 × 16</td>
<td>9</td>
<td>9</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>32 × 32</td>
<td>10</td>
<td>9</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>64 × 64</td>
<td>9</td>
<td>9</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>128 × 128</td>
<td>9</td>
<td>9</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Observations:
- The number of GMRES iterations with $\gamma$ chosen by LFA is almost the same as for the optimal $\gamma$, especially on the finest grid.
- The iteration counts with both sets of $\gamma$ are independent of grid size and only mildly dependent on viscosity.
Values of $\gamma$

Table: The values of $\gamma$ chosen by LFA and optimal values (cavity, Q2-Q1, uniform grids)

<table>
<thead>
<tr>
<th>Viscosity</th>
<th>0.1</th>
<th>0.01</th>
<th>0.005</th>
<th>0.001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid</td>
<td>LFA</td>
<td>Opt</td>
<td>LFA</td>
<td>Opt</td>
</tr>
<tr>
<td>16 × 16</td>
<td>0.42</td>
<td>0.45</td>
<td>0.075</td>
<td>0.085</td>
</tr>
<tr>
<td>32 × 32</td>
<td>0.29</td>
<td>0.38</td>
<td>0.056</td>
<td>0.050</td>
</tr>
<tr>
<td>64 × 64</td>
<td>0.32</td>
<td>0.32</td>
<td>0.055</td>
<td>0.045</td>
</tr>
<tr>
<td>128 × 128</td>
<td>0.28</td>
<td>0.28</td>
<td>0.036</td>
<td>0.046</td>
</tr>
</tbody>
</table>
Eigenvalues of preconditioned matrices

Figure: Plots of the eigenvalues of the preconditioned Oseen matrix (lid driven cavity, Q2-Q1, 32 × 32 uniform grid, $\nu = 0.01$). Left: with optimal $\gamma$. Right: with $\gamma$ chosen by Fourier analysis.

The two values of $\gamma$ are very close: 0.050 vs. 0.056.

The eigenvalue $\lambda = 1$ has multiplicity $n$ (for all $\gamma$).
Iteration counts with various values of $\gamma$

Figure: GMRES iterations with modified AL preconditioner (cavity, Q2-Q1, uniform grids, $\nu = 0.001$)
Use the same values of $\gamma$ as for Picard linearization.

**Table:** GMRES iterations with modified AL preconditioner (cavity, Q2-Q1, stretched grids, Newton)

<table>
<thead>
<tr>
<th>Viscosity</th>
<th>0.1</th>
<th>0.01</th>
<th>0.005</th>
<th>0.001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid</td>
<td>LFA</td>
<td>Opt</td>
<td>LFA</td>
<td>Opt</td>
</tr>
<tr>
<td>16 $\times$ 16</td>
<td>13</td>
<td>13</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>32 $\times$ 32</td>
<td>14</td>
<td>14</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>64 $\times$ 64</td>
<td>14</td>
<td>14</td>
<td>24</td>
<td>23</td>
</tr>
<tr>
<td>128 $\times$ 128</td>
<td>15</td>
<td>14</td>
<td>26</td>
<td>23</td>
</tr>
</tbody>
</table>

Not quite $h$-independent for small $\nu$. 
Results for stretched grids and comparison with PCD/LSC/mPCD

Table: GMRES with modified AL preconditioner (cavity, Q2-Q1, stretched grids)

<table>
<thead>
<tr>
<th>Viscosity</th>
<th>0.1</th>
<th>0.01</th>
<th>0.005</th>
<th>0.001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid</td>
<td>FA</td>
<td>Opt</td>
<td>FA</td>
<td>Opt</td>
</tr>
<tr>
<td>16 × 16</td>
<td>9</td>
<td>9</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>32 × 32</td>
<td>9</td>
<td>9</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>64 × 64</td>
<td>8</td>
<td>8</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>128 × 128</td>
<td>8</td>
<td>7</td>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>

Table: GMRES iterations with PCD, LSC and mPCD preconditioners of Elman, Silvester, Wathen (cavity, Q2-Q1, stretched grids, $\nu = 0.001$)

<table>
<thead>
<tr>
<th>Grid</th>
<th>PCD</th>
<th>LSC</th>
<th>mPCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 × 16</td>
<td>79</td>
<td>50</td>
<td>81</td>
</tr>
<tr>
<td>32 × 32</td>
<td>105</td>
<td>78</td>
<td>201</td>
</tr>
<tr>
<td>64 × 64</td>
<td>117</td>
<td>117</td>
<td>135</td>
</tr>
<tr>
<td>128 × 128</td>
<td>117</td>
<td>174</td>
<td>144</td>
</tr>
</tbody>
</table>

Note: All methods have similar costs per iteration.
A backward facing step test problem

Figure: Backward facing step problem (Q2-Q1, $\nu = 0.005$, uniform $64 \times 192$ grid)
Table: GMRES iterations with modified AL preconditioner (step, Q2-Q1, uniform grids)

<table>
<thead>
<tr>
<th>Viscosity</th>
<th>0.1</th>
<th>0.01</th>
<th>0.005</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid</td>
<td>LFA</td>
<td>Opt</td>
<td>LFA</td>
</tr>
<tr>
<td>16 × 48</td>
<td>15</td>
<td>12</td>
<td>46</td>
</tr>
<tr>
<td>32 × 96</td>
<td>12</td>
<td>12</td>
<td>24</td>
</tr>
<tr>
<td>64 × 192</td>
<td>12</td>
<td>11</td>
<td>17</td>
</tr>
<tr>
<td>128 × 384</td>
<td>11</td>
<td>11</td>
<td>15</td>
</tr>
</tbody>
</table>
Comparison of exact and inexact solves

In the following Table we present a comparison of modified AL preconditioning with exact and inexact inversion of diagonal blocks $A_{ii}$.

For the ‘exact’ solves we use the sparse LU factorization with column AMD reordering available in Matlab.

For the inexact solves we use one iteration (V-cycle) of AMG using the HSL−MI20 code developed by Boyle, Mihajlovic and Scott (IJNME 2010).

We perform tests for both Picard and Newton linearizations of the lid driven cavity problem discretized with Q2-Q1 elements (Newton is harder), using the same value of $\gamma$ from Fourier analysis in both cases. The viscosity is $\nu = 0.005$.

The experiments are performed in Matlab 7.9.0 on a Sun Microsystems SunFire.

The upshot:

- Using inexact solves does not affect the convergence rates
- Inexact solves result in much faster solution times
**Iteration counts and timings of exact solve and AMG (MI20)**

*Table:* Comparison of exact and inexact inner solvers. GMRES iterations and timings with modified AL preconditioner (cavity, Q2-Q1, uniform grids, $\nu = 0.005$)

<table>
<thead>
<tr>
<th>Grid</th>
<th>Picard</th>
<th>Newton</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Timings</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Exact</td>
<td>MI20</td>
</tr>
<tr>
<td>64 × 64</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>Set-up time</td>
<td>1.93</td>
<td>0.31</td>
</tr>
<tr>
<td>Iter time</td>
<td>0.62</td>
<td>2.76</td>
</tr>
<tr>
<td>Total time</td>
<td>2.55</td>
<td>3.07</td>
</tr>
<tr>
<td>128 × 128</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>Setup time</td>
<td>34.90</td>
<td>1.29</td>
</tr>
<tr>
<td>Iter time</td>
<td>4.44</td>
<td>12.00</td>
</tr>
<tr>
<td>Total time</td>
<td>39.34</td>
<td>13.29</td>
</tr>
<tr>
<td>256 × 256</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>Setup time</td>
<td>856.74</td>
<td>5.86</td>
</tr>
<tr>
<td>Iter time</td>
<td>40.22</td>
<td>58.84</td>
</tr>
<tr>
<td>Total time</td>
<td>896.96</td>
<td>64.70</td>
</tr>
</tbody>
</table>
The modified AL preconditioner for 3D problems

For 3D Oseen problems $A = \text{diag}(A_1, A_2, A_3)$ and $B = (B_1, B_2, B_3)$.

$$A_\gamma = A + \gamma B^T W^{-1} B$$

$$= \begin{pmatrix} A_1 & O & O \\ O & A_2 & O \\ O & O & A_3 \end{pmatrix} + \gamma \begin{pmatrix} B_1^T \\ B_2^T \\ B_3^T \end{pmatrix} W^{-1} \begin{pmatrix} B_1 \\ B_2 \\ B_3 \end{pmatrix}$$

$$= \begin{pmatrix} A_1 + \gamma B_1^T W^{-1} B_1 & \gamma B_1^T W^{-1} B_2 & \gamma B_1^T W^{-1} B_3 \\ \gamma B_2^T W^{-1} B_1 & A_2 + \gamma B_2^T W^{-1} B_2 & \gamma B_2^T W^{-1} B_3 \\ \gamma B_3^T W^{-1} B_1 & \gamma B_3^T W^{-1} B_2 & A_3 + \gamma B_3^T W^{-1} B_3 \end{pmatrix}$$

$$=: \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix},$$

so the modified AL preconditioner is

$$\tilde{P} = \begin{pmatrix} \tilde{A}_\gamma & B^T \\ O & \tilde{S} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & B_{1}^T \\ O & A_{22} & A_{23} & B_{2}^T \\ O & O & A_{33} & B_{3}^T \\ O & O & O & \tilde{S} \end{pmatrix}.$$
Preliminary parallel results

These preliminary runs are done on a small cluster using the Trilinos package (SNL). The subproblems in the modified AL preconditioner are solved inexactly by one AMG iteration (‘ML’ solver in Trilinos). Each node of the cluster has two dual core AMD 2.2 GHz Opteron CPUs, 4 GB RAM and 80 GB drive.

Steady Oseen problem ($\nu = 0.01$), MAC discretization, $128^3$ grid; total number of degrees of freedom is $8M$, for a total of about $100M$ nonzeros in $A$. 

<table>
<thead>
<tr>
<th># of cores</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>26</td>
<td>26</td>
<td>28</td>
<td>30</td>
</tr>
<tr>
<td>Setup time</td>
<td>61.08</td>
<td>37.42</td>
<td>19.41</td>
<td>12.29</td>
</tr>
<tr>
<td>Iter time</td>
<td>209.08</td>
<td>152.4</td>
<td>87.04</td>
<td>45.68</td>
</tr>
<tr>
<td>Total time</td>
<td>270.16</td>
<td>189.82</td>
<td>106.45</td>
<td>57.97</td>
</tr>
</tbody>
</table>

Note: For $64^3$ grid (1M unknowns), total time on 16 cores is 8.67s (30 its).
Preliminary parallel results

These preliminary runs are done on a small cluster using the Trilinos package (SNL). The subproblems in the modified AL preconditioner are solved inexacty by one AMG iteration (‘ML’ solver in Trilinos). Each node of the cluster has two dual core AMD 2.2 GHz Opteron CPUs, 4 GB RAM and and 80 GB drive.

Steady Oseen problem ($\nu = 0.01$), MAC discretization, $128^3$ grid; total number of degrees of freedom is $8.3M$, for a total of about $100M$ nonzeros in $A$. 

<table>
<thead>
<tr>
<th># of cores</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>26</td>
<td>26</td>
<td>28</td>
<td>30</td>
</tr>
<tr>
<td>Setup time</td>
<td>61.08</td>
<td>37.42</td>
<td>19.41</td>
<td>12.29</td>
</tr>
<tr>
<td>Iter time</td>
<td>209.08</td>
<td>152.4</td>
<td>87.04</td>
<td>45.68</td>
</tr>
<tr>
<td>Total time</td>
<td>270.16</td>
<td>189.82</td>
<td>106.45</td>
<td>57.97</td>
</tr>
</tbody>
</table>

Note: For $64^3$ grid (1M unknowns), total time on 16 cores is 8.67s (30 its).
Preliminary parallel results

These preliminary runs are done on a small cluster using the Trilinos package (SNL). The subproblems in the modified AL preconditioner are solved inexactly by one AMG iteration (‘ML’ solver in Trilinos). Each node of the cluster has two dual core AMD 2.2 GHz Opteron CPUs, 4 GB RAM and an 80 GB drive.

Steady Oseen problem ($\nu = 0.01$), MAC discretization, $128^3$ grid; total number of degrees of freedom is $8.3M$, for a total of about $100M$ nonzeros in $A$.

Table: GMRES iterations and timings for modified AL preconditioner (inexact solves, $\gamma = 0.06$.)

<table>
<thead>
<tr>
<th># of cores</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>26</td>
<td>26</td>
<td>28</td>
<td>30</td>
</tr>
<tr>
<td>Setup time</td>
<td>61.08</td>
<td>37.42</td>
<td>19.41</td>
<td>12.29</td>
</tr>
<tr>
<td>Iter time</td>
<td>209.08</td>
<td>152.4</td>
<td>87.04</td>
<td>45.68</td>
</tr>
<tr>
<td>Total time</td>
<td>270.16</td>
<td>189.82</td>
<td>106.45</td>
<td>57.97</td>
</tr>
</tbody>
</table>

Note: For $64^3$ grid (1M unknowns), total time on 16 cores is 8.67s (30 its).
Outline

1. Properties of saddle point matrices
2. Examples: Incompressible flow problems
3. Some solution algorithms
4. The Augmented Lagrangian (AL) approach
5. The modified Augmented Lagrangian-based preconditioner
6. Conclusions
Conclusions and future work

- Large linear systems of saddle point type still pose a significant challenge for modern preconditioned iterative solvers.
Conclusions and future work

- Large linear systems of saddle point type still pose a significant challenge for modern preconditioned iterative solvers.
- Computing steady solutions to incompressible flow problems for small values of the viscosity and on stretched grids is not easy.
Conclusions and future work

- Large linear systems of saddle point type still pose a significant challenge for modern preconditioned iterative solvers.
- Computing steady solutions to incompressible flow problems for small values of the viscosity and on stretched grids is not easy.
- Suitably modified and combined with AMG-type inner solvers, the AL approach results in fairly robust preconditioners.
Conclusions and future work

- Large linear systems of saddle point type still pose a significant challenge for modern preconditioned iterative solvers.
- Computing steady solutions to incompressible flow problems for small values of the viscosity and on stretched grids is not easy.
- Suitably modified and combined with AMG-type inner solvers, the AL approach results in fairly robust preconditioners.
- Both stable and stabilized discretizations can be accommodated.
Conclusions and future work

- Large linear systems of saddle point type still pose a significant challenge for modern preconditioned iterative solvers.
- Computing steady solutions to incompressible flow problems for small values of the viscosity and on stretched grids is not easy.
- Suitably modified and combined with AMG-type inner solvers, the AL approach results in fairly robust preconditioners.
- Both stable and stabilized discretizations can be accommodated.
- Stretched grids do not pose any difficulties to the AL approach.
Conclusions and future work

- Large linear systems of saddle point type still pose a significant challenge for modern preconditioned iterative solvers.
- Computing steady solutions to incompressible flow problems for small values of the viscosity and on stretched grids is not easy.
- Suitably modified and combined with AMG-type inner solvers, the AL approach results in fairly robust preconditioners.
- Both stable and stabilized discretizations can be accommodated.
- Stretched grids do not pose any difficulties to the AL approach.
- The Fourier-based approach gives very good estimates of the optimal parameter $\gamma$.

Current and future work: parallelization; application to real problems.
Conclusions and future work

- Large linear systems of saddle point type still pose a significant challenge for modern preconditioned iterative solvers.
- Computing steady solutions to incompressible flow problems for small values of the viscosity and on stretched grids is not easy.
- Suitably modified and combined with AMG-type inner solvers, the AL approach results in fairly robust preconditioners.
- Both stable and stabilized discretizations can be accommodated.
- Stretched grids do not pose any difficulties to the AL approach.
- The Fourier-based approach gives very good estimates of the optimal parameter $\gamma$.
- Clearly better than competing methods on difficult problems.
Conclusions and future work

- Large linear systems of saddle point type still pose a significant challenge for modern preconditioned iterative solvers.
- Computing steady solutions to incompressible flow problems for small values of the viscosity and on stretched grids is not easy.
- Suitably modified and combined with AMG-type inner solvers, the AL approach results in fairly robust preconditioners.
- Both stable and stabilized discretizations can be accommodated.
- Stretched grids do not pose any difficulties to the AL approach.
- The Fourier-based approach gives very good estimates of the optimal parameter $\gamma$.
- Clearly better than competing methods on difficult problems.
- Current and future work: parallelization; application to real problems.
References


Results for coupled Vanka-multigrid preconditioner

Results for Vanka-MG-Bi-CGSTAB approach, isoP2-P0 FEM.

<table>
<thead>
<tr>
<th>mesh size $h$</th>
<th>viscosity = $\nu$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>constant wind</td>
<td></td>
</tr>
<tr>
<td>1/16</td>
<td>4</td>
</tr>
<tr>
<td>1/32</td>
<td>4</td>
</tr>
<tr>
<td>1/64</td>
<td>4</td>
</tr>
<tr>
<td>1/128</td>
<td>4</td>
</tr>
<tr>
<td>rotating vortex</td>
<td></td>
</tr>
<tr>
<td>1/16</td>
<td>4</td>
</tr>
<tr>
<td>1/32</td>
<td>4</td>
</tr>
<tr>
<td>1/64</td>
<td>4</td>
</tr>
<tr>
<td>1/128</td>
<td>4</td>
</tr>
</tbody>
</table>

Number of preconditioned Bi-CGSTAB iterations

On the ‘easy’ problem the method is perfectly robust, displaying $h$- and $\nu$-independent behavior. For the harder problem, the solver breaks down for very small $\nu$. 