Fast iterative methods for solving the incompressible Navier-Stokes equations

Carlos Echeverría Serur

August 28, 2013

Table of Contents

Motivation

- Except for some simple cases, the analytical solution of the (N-S) equations is impossible.
- The efficient solution of the linearized system of equations is of primary interest.
- Industry: slowdown in solvers (MARIN).

Linear systems everywhere!

Always Simplify:

$$\left.\begin{array}{lr} -\nu\nabla^2\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p = \mathbf{f} & \text{in } \Omega \\ \nabla\cdot\mathbf{u} = 0 & \text{in } \Omega \end{array}\right\} \longrightarrow Ax = b$$

1. **Discretization** done by the Finite Element Method (plus a correct choice of finite elements).
2. **Linearization** performed by the Newton or Picard method.
3. Saddle-Point problem is obtained:

$$\begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$$

Stagnation Phase of SIMPLER

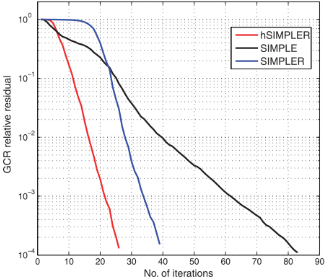


Figure: Convergence plot of SIMPLE-type peconditioners for the Stokes problem. [Vuik et. al. 2009]

The main effort of this thesis is to investigate and, if possible, to eliminate this stagnation behavior, thus providing a more efficient way of solving the N-S system of equations.

Research Questions

- Why is there a stagnation phase in the iterative solution of the SIMPLER-preconditioned Navier-Stokes Algebraic System?
- How do we eliminate it?
- Why does the number of iterations get worse for stretched grids? (MARIN).
- How do we avoid this?

Discretization

The incompressible flow of a Newtonian fluid is governed by the behavior defined by the set of equations:

$$-\nu\nabla^2\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p = \mathbf{f},$$
$$\nabla\cdot\mathbf{u} = 0,$$

posed on a two or three dimensional domain Ω, together with boundary conditions on $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ given by

$$\mathbf{u} = \mathbf{w} \text{ on } \partial\Omega_D, \quad \nu\frac{\partial\mathbf{u}}{\partial n} - \mathbf{n}p = \mathbf{s} \text{ on } \partial\Omega_N.$$

Finite Element Discretization of N-S:

Weak Formulation → Discrete Weak Formulation → Galerkin method → Matrix formulation.

Matrix Formulation:

$$A_d u + N(u) + B^T p = f,$$
$$Bu = g,$$

Here u and p denote the vectors of unknowns.

Linearization

Picard:

The nonlinear term is substituted by an approximation including the velocity vector calculated at distinct time steps:

$$u^{k+1} \cdot \nabla u^{k+1} \approx u^k \cdot \nabla u^{k+1}.$$

Newton:

The velocity field at the new time-step is the sum of the velocity field at the previous time step plus a correction:

$$u^{k+1} = u^k + \delta u^k.$$

After linearizing we obtain an algebraic system of equations:

$$Fu + B^T p = f,$$
$$Bu = g.$$

Saddle-Point Problem

Linear system:

$$\begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \text{ is of the form } Ax = b$$

Any solution (u^*, p^*) of the previous linear system is a saddle point for the Lagrangian:

$$\mathcal{L}(u, p) = \frac{1}{2} u^T A u - f^T u + (Bu - g)^T p.$$

Projection Techniques

Extract an approximate solution of $Ax = b$ from a subspace of $\mathbb{R}^n$:

$$\text{Find } \tilde{x} \in x_0 + \mathcal{K}_m \text{ such that } b - A\tilde{x} \perp \mathcal{L}_m.$$

Krylov Subspaces

A Krylov subspace method is a projection method for which the subspace $\mathcal{K}_m$ is the Krylov subspace:

$$\mathcal{K}_m(A, r_0) = \text{span}\{r_0, Ar_0, A^2 r_0, \ldots, A^{m-1} r_0\}.$$

The different versions of Krylov subspace methods arise from different choices of the subspace $\mathcal{L}_m$ and from the ways in which the system is preconditioned.

Solution Methods

GMRES

The generalized minimal residual method (GMRES) is a projection method based on taking $\mathcal{K} = \mathcal{K}_m$ and $\mathcal{L} = A\mathcal{K}_m$, in which $\mathcal{K}_m$ is the m-th Krylov subspace, with $\|v_1\| = r_0/\|r_0\|_2$.

Preconditioning

It's all about the eigenvalues

$$M^{-1}Ax = M^{-1}b.$$

or

$$AM^{-1}u = b, x = M^{-1}u$$

1. This system, which has the same solution as the original system, is called a *preconditioned system* and M is the *preconditioning matrix* or *preconditioner*.
2. Clustered Spectrum = Fast Convergence.

Deflation

Definition:

Let A be an SPSD matrix. Suppose that $Z \in \mathbb{R}^{n \times k}$ with full rank is given. Then we define the invertible Galerkin matrix, $E \in \mathbb{R}^{k \times k}$, the correction matrix, $Q \in \mathbb{R}^{n \times n}$, and the deflation matrix $P \in \mathbb{R}^{n \times n}$, as follows:

$$P = I - AQ, \quad Q = ZE^{-1}Z^T, \quad E = Z^T AZ \tag{1}$$

In the previous equations, Z is the so-called 'deflation subspace matrix' whose k columns are called the 'deflation vectors' or 'projection vectors'. These vectors remain unspecified for the moment, but they are chosen in such a way that E is nonsingular.

Block-type Preconditioners

Block preconditioners are based on a block factorization of the coefficient matrix:

$$\begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} = LDU = \begin{bmatrix} I & 0 \\ BF^{-1} & I \end{bmatrix} \begin{bmatrix} F & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & F^{-1}B^T \\ 0 & I \end{bmatrix}.$$

$$P_t = DU = \begin{bmatrix} F & B^T \\ 0 & S \end{bmatrix}$$

Algorithm: Preconditioner P_t

1. Solve $Sz_p = r_p$
2. Update $r_u = r_u - B^T z_p$
3. Solve $Fz_u = r_u$

By investigating the following generalized eigenvalue problem, we can determine the eigenvalues of the preconditioned system:

$$\begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \lambda \begin{bmatrix} F & B^T \\ 0 & S \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix}$$

$$-BF^{-1}B^T p = \lambda S p.$$

This shows that whenever $S = -BF^{-1}B^T$ we have $\lambda = 1$ with multiplicity n_p. From this equation, we can see that a good approximation of the Schur complement matrix will dictate the convergence behavior of the preconditioned system with P_t.

SIMPLE preconditioner

The algorithms follows from a block LU decomposition of the original coefficient matrix:

$$\begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} F & 0 \\ B & -BF^{-1}B^T \end{bmatrix} \begin{bmatrix} I & F^{-1}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix},$$

The approximation $F^{-1} = D^{-1} = diag(F)^{-1}$ in the $(2,2)$ and $(1,2)$ block of the L and U block matrices, respectively, leads to the SIMPLE algorithm. Solve recursively the following systems:

$$\begin{bmatrix} F & 0 \\ B & -BD^{-1}B^T \end{bmatrix} \begin{bmatrix} u^* \\ p^* \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix},$$

and

$$\begin{bmatrix} I & D^{-1}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} u^* \\ \delta p \end{bmatrix}.$$

SIMPLER preconditioner

A variant of SIMPLE, known as SIMPLER, is supposed to provide Reynolds-independent convergence. Instead of estimating the pressure p^* in the SIMPLE algorithm, p^* is obtained from solving a subsystem

$$\hat{S}p^* = r_p - BD^{-1}((D - F)u^k + r_u), \quad (2)$$

where u^k is obtained from the prior iteration. In case SIMPLER is used as a preconditioner, u^k is taken equal to zero, therefore:

$$\hat{S}p^* = r_p - BD^{-1}r_u. \quad (3)$$

Algorithm

Algorithm: SIMPLER Preconditioner

1. Solve $\hat{S}p^* = r_p - BD^{-1}r_u$.
2. Solve $Fu^* = r_u - B^T p^*$.
3. Solve $\hat{S}\delta p = r_p - Bu^* - Cp^*$.
4. Update $z_u = u^* - D^{-1}B^T\delta p$.
5. Update $z_p = p^* + \delta p$.

The SIMPLER and other SIMPLE-type algorithms were implemented as preconditioners and numerical simulations were carried out.

How?

- IFISS: is a graphical Matlab package for the interactive numerical study of incompressible flow problems. It includes algorithms for discretization by mixed finite element methods and a posteriori error estimation of the computed solutions.
- The package can also be used as a computational laboratory for experimenting with state-of-the-art preconditioned iterative solvers for the discrete linear equation systems that arise in incompressible flow modeling

Numerical Experiments

In all simulations, the linearization of the nonlinear system of equations was performed via the Picard method. The tolerance of the linear approximation was set to $1e-8$. At each nonlinear iteration, we accept an iterate x_k of the `GMRES` method with right preconditioning as a valid solution when,

$$\frac{\|M^{-1}(B - Ax_k)\|}{\|M^{-1}(B - Ax_0)\|} \leq 1e{-}6. \tag{4}$$

The convergence grpahs show $\log_{10}(\frac{\|M^{-1}(B-Ax_k)\|}{\|M^{-1}(B-Ax_0)\|})$ Vs. k where k is the iteration number.

Backward Facing Step

This example represents the flow over a step of length L.

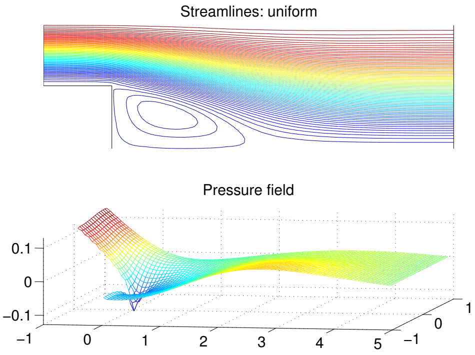


Figure: Solution to the backward facing step with $\mathbf{Q_2} - \mathbf{Q_1}$ elements.

Results

ν	0.1	0.01	0.005
16×48	20 (7)	34 (25)	50 (64)
32×96	29 (7)	38 (23)	56 (70)
64×192	45 (6)	47 (21)	57 (63)
128×384	70 (6)	74 (19)	77 (54)

Table: Iteration numbers for the backward facing step flow using the FGMRES method preconditioned with SIMPLER(WBC). The number of Picard iterations appears in parenthesis.

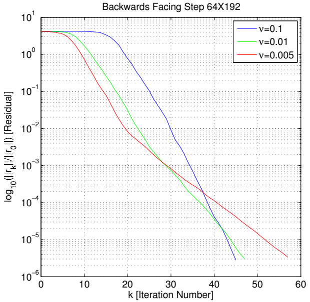
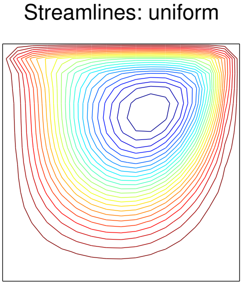
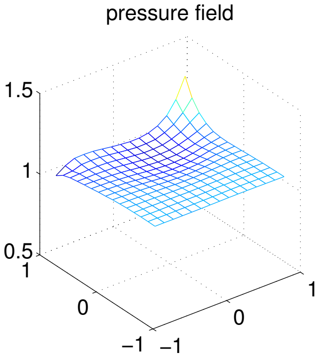


Figure: Convergence plot of FGMRES method preconditioned with SIMPLER(WBC) for the backwards facing step flow.

Driven Cavity Flow

This is the model of flow inside a square cavity.





Results

ν	0.1	0.01	0.005
$h = \frac{1}{32}$	16 (6)	18 (13)	21 (14)
$h = \frac{1}{64}$	24 (5)	23 (12)	24 (13)
$h = \frac{1}{128}$	37 (5)	40 (11)	36 (11)
$h = \frac{1}{256}$	58 (4)	66 (10)	67 (10)

Table: Iteration numbers for the driven cavity flow using the FGMRES method preconditioned with SIMPLER(WBC). The number of Picard iterations appears in parenthesis.

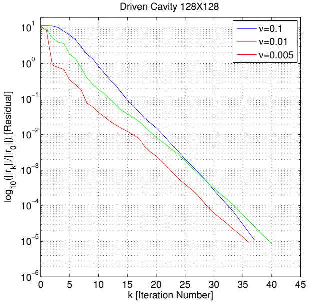


Figure: Convergence plot of FGMRES method preconditioned with SIMPLER(WBC) for the driven cavity flow.

Poiseuille Flow

This example represents the flow inside a channel.

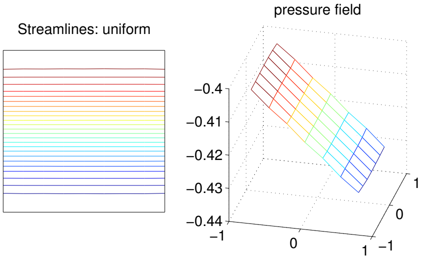


Figure: Solution to the Poiseuille flow problem.

Results

ν	0.1	0.01	0.005
$h = \frac{1}{32}$	17 (1)	22 (1)	30 (1)
$h = \frac{1}{64}$	27 (1)	24 (1)	29 (1)
$h = \frac{1}{128}$	44 (1)	34 (1)	33 (1)
$h = \frac{1}{256}$	72 (1)	66 (1)	54 (1)

Table: Iteration numbers for the Poiseuille flow using the FGMRES method preconditioned with SIMPLER(WBC). The number of Picard iterations appears in parenthesis.

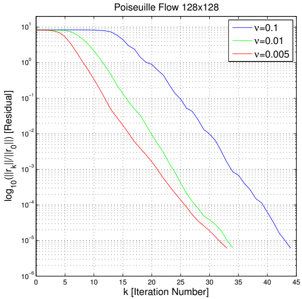


Figure: Convergence plot of FGMRES method preconditioned with SIMPLER(WBC) for the Poiseuille flow.

Side result:

The SIMPLE(R) preconditioner was implemented to work with IFISS. A new version is proposed SIMPLER(WOBC). Stagnation disappears but increases number of iterations.

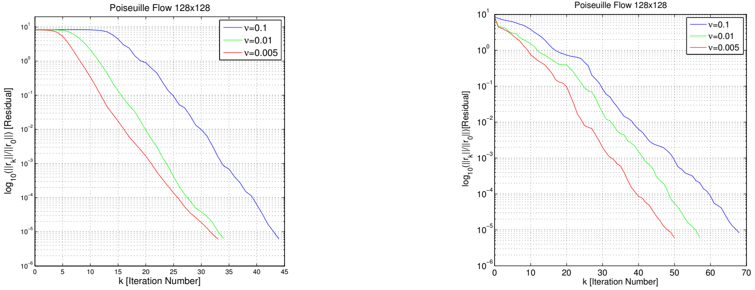


Figure: Convergence plot of FGMRES method preconditioned with SIMPLER(WOBC) for the Poiseuille flow.

Eigenvalue Exploration

An analysis of the spectrum of the preconditioned matrix was carried out in order to find patterns in the eigenvalue positions.

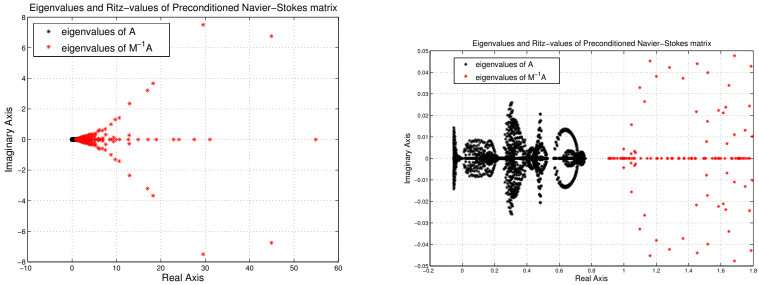
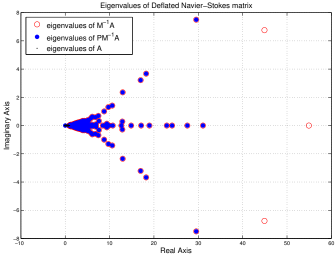


Figure: Spectrum of eigenvalues of the Navier-Stokes system (A) and preconditioned with SIMPLER ($M^{-1}A$) for the Poiseuille flow

The eigenvalue exploration gives the idea of deflating the undesired eigenvalues.



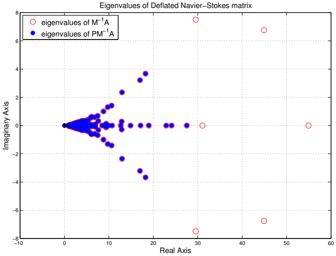
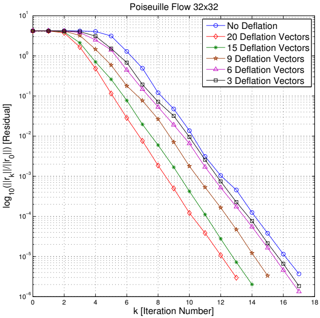


Figure: Comparison of eigenvalues of the Navier-Stokes system preconditioned with SIMPLER ($M^{-1}A$) [red] and of the deflated system [blue] for the Poiseuille flow for different sizes of the deflation matrix

Results after Deflation



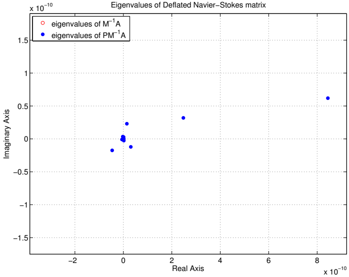


Figure: Convergence History of the SIMPLER(WBC) preconditioned Navier-Stokes system ($M^{-1}A$) and deflated ($PM^{-1}A$) for the Poiseuille Flow for different sizes of the deflation matrix.

Results on stretched grids

It was found that, when SIMPLER(WBC) is used as a preconditioner, the number of iterations for the stretched and un-stretched girds remains constant. This can not be said when SIMPLER(WOBC) is used:

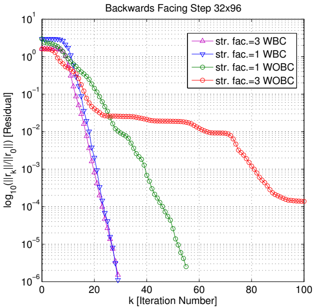


Figure: Backward Facing Step (stretched) convergence plot of FGMRES method.

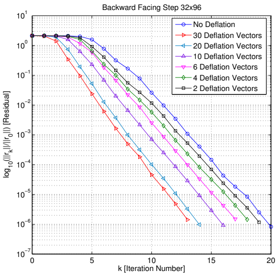


Figure: Convergence History of the SIMPLER(WBC) preconditioned Navier-Stokes system ($M^{-1}A$) and deflated ($PM^{-1}A$) for the Backward Facing Step for different sizes of the deflation matrix with stretched grid factor of 3

Once again, we find a positive reduction of the stagnation behavior when a large deflation matrix is used.

Ongoing Research (Silvester, Liao 2013)

- Single element construction:

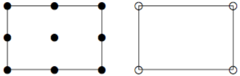


Figure: $\mathbf{Q}_2 - \mathbf{Q}_1$ element (• velocity, ∘ pressure).

- Macroelement construction:

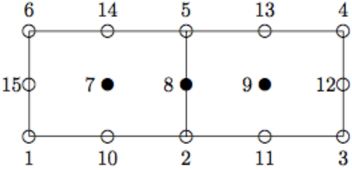


Figure: $\mathbf{Q}_2 - \mathbf{Q}_1$ macroelement numbering for two-element patch $\square_k \bigcup \square_m$.

Research Questions (revisited)-Conclusions

- Why is there a stagnation phase in the iterative solution of the SIMPLER-preconditioned Navier-Stokes Algebraic System?
- Due to the complicated nature of the NS matrix, there exist unfavorable eigenvalues which compromise the performance of the iterative solver (GMRES).
- Why does the number of iterations get worse for stretched grids?
- The LBB condition is no longer satisfied.

A deflation approach is successful in treating the ill-conditioned eigenvalues in both cases (stretched/unstretched).
The success of this approach allows us to suggest the construction of a *two-level preconditioned* `GMRES` *method* for solving the Navier-Stokes equations.

Future Research

A direct continuation of the direction of research of this thesis would focus on the next points:

- Formalize the concepts presented in this thesis and construct a two-level preconditioned SIMPLER algorithm.
- Investigate the effect of using different deflation techniques (different approaches of constructing the deflation matrix) applied to the SIMPLER-preconditioned Navier-Stokes matrix for stretched and non-stretched grids in the efficiency of the solution.
- Implement the macroelement construction in the FEM discretization of the Navier-Stokes system of equations.