An advanced ILU preconditioner for the incompressible Navier-Stokes equations

M. ur Rehman  C. Vuik  A. Segal

Delft Institute of Applied Mathematics, TU delft
The Netherlands

Computational Methods with Applications, August 19 -25, Harrachov 2007, Czech Republic
Introduction
Solution techniques
Preconditioning
Numerical Experiments
Conclusions

Rehman, Vuik and Segal
Solution of the Navier-Stokes problem
The incompressible Navier-Stokes equation

\[-\nu\nabla^2 \bm{u} + \bm{u} \cdot \nabla \bm{u} + \nabla p = f \quad \text{in} \quad \Omega\]

\[\nabla \cdot \bm{u} = 0 \quad \text{in} \quad \Omega.\]

\(\bm{u}\) is the fluid velocity
\(p\) is the pressure field
\(\nu > 0\) is the kinematic viscosity coefficient (\(1/Re\)).
\(\Omega \subset \mathbb{R}^2\) is a bounded domain with the boundary condition:

\[\bm{u} = \bm{w} \quad \text{on} \quad \partial \Omega_D, \quad \nu \frac{\partial \bm{u}}{\partial \bm{n}} - \bm{n}p = 0 \quad \text{on} \quad \partial \Omega_N.\]
 Finite element discretization

Weak formulation

\[ \tilde{X} = (H^1_E(\Omega))^d, \quad X = (H^1_0(\Omega))^d, \quad M = L^2(\Omega) \]

Find \( u \in \tilde{X} \) and \( p \in M \)

\[ \nu \int_\Omega \nabla u : \nabla v d\Omega + \int_\Omega (u.\nabla u).v d\Omega - \int_\Omega p(\nabla . v) d\Omega = \int_\Omega f.v d\Omega, \quad \forall v \in X \]

\[ \int_\Omega q(\nabla . u) d\Omega = 0, \quad \forall q \in M \]
**Finite element discretization**

**Discrete weak formulation**

\[
\tilde{X}_h = (H^1_E(\Omega))^d, \quad X_h = (H^1_0(\Omega))^d, \quad M_h = L^2(\Omega)
\]

Find \( u_h \in \tilde{X}_h \) and \( p_h \in M_h \)

\[
\nu \int_\Omega \nabla u_h : \nabla v_h d\Omega + \int_\Omega (u_h \cdot \nabla u_h) \cdot v_h d\Omega - \int_\Omega p_h (\nabla \cdot v_h) d\Omega = \int_\Omega f \cdot v_h d\Omega, \quad \forall v_h \in X_h,
\]

\[
\int_\Omega q_h (\nabla \cdot u_h) d\Omega = 0 \quad \forall q_h \in M_h.
\]

**Matrix notation**

\[
Au + N(u) + B^T p = f
\]
\[
Bu = 0.
\]
## Linearization

### Stokes problem

\[- \nu \nabla^2 \mathbf{u} + \nabla p = \mathbf{f}\]

\[\nabla \cdot \mathbf{u} = 0\]

### Picard’s method

\[- \nu \Delta \mathbf{u}^{(k+1)} + (\mathbf{u}^{(k)} \cdot \nabla) \mathbf{u}^{(k+1)} + \nabla p^{(k+1)} = \mathbf{f}\]

\[\nabla \cdot \mathbf{u}^{(k+1)} = 0\]

### Newton’s method

\[- \nu \Delta \mathbf{u}^{k+1} + \mathbf{u}^{k+1} \cdot \nabla \mathbf{u}^k + \mathbf{u}^k \cdot \nabla \mathbf{u}^{k+1} + \nabla p^{k+1} = \mathbf{f} + \mathbf{u}^k \cdot \nabla \mathbf{u}^k,\]

\[\nabla \cdot \mathbf{u}^{k+1} = 0.\]
Linear system

Matrix form after linearization

\[
\begin{bmatrix}
F & B^T \\
B & 0
\end{bmatrix}
\begin{bmatrix}
u \\
p
\end{bmatrix} =
\begin{bmatrix}
f \\
0
\end{bmatrix} \quad \text{or } Ax = b
\]

- \( F \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{m \times n}, f \in \mathbb{R}^n \) and \( m \leq n \)
- Sparse linear system, Symmetric (Stokes problem), nonsymmetric indefinite otherwise.
- Saddle point problem having large number of zeros on the main diagonal
Introduction
Solution techniques
Preconditioning
Numerical Experiments
Conclusions

Solution techniques

Direct methods
To solve $Ax = b$,
factorize $A$ into upper $U$ and lower $L$ triangular matrices
($LUx = b$)
First solve $Ly = b$, then $Ux = y$

Classical iterative methods
Methods based on matrix splitting, generates sequence of iterations
$x_{k+1} = M^{-1}(Nx_k + b) = Qx_k + s$
where $A = M - N$
Jacobi, Gauss Seidel, SOR, SSOR
### Solution techniques

#### Krylov subspace methods

Find the approximate solution \( x_n = x_0 + c \), where \( c \) is a linear combination of basis functions of Krylov subspace \( \mathcal{K}_n(A, b) \), where \( \mathcal{K}_n = \langle b, Ab, A^2 b, \ldots, A^{n-1} b \rangle \) of dimension \( n \).

<table>
<thead>
<tr>
<th>Method</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGNR</td>
<td>Paige and Saunders - 1975</td>
</tr>
<tr>
<td>QMR</td>
<td>Freund and Nachtigal - 1991</td>
</tr>
<tr>
<td>CGS</td>
<td>Sonneveld - 1989</td>
</tr>
<tr>
<td>Bi-CGSTAB</td>
<td>van der Vorst - 1992</td>
</tr>
<tr>
<td>GMRES</td>
<td>Saad and Schultz - 1986</td>
</tr>
<tr>
<td>GMRESR</td>
<td>van der Vorst and Vuik - 1994</td>
</tr>
<tr>
<td>GCR</td>
<td>Eisenstat, Elman and Schultz - 1986</td>
</tr>
</tbody>
</table>

- matrix-vector multiplications, good convergence properties, optimal and short recurrence
- Convergence depends strongly on eigenvalues distribution clustered around 1 or away from 0.
Preconditioner for the Navier-Stokes equations

**Definition**

A linear system $Ax = b$ is transformed into $P^{-1}Ax = P^{-1}b$ such that

- Eigenvalues of $P^{-1}A$ are more clustered than $A$
- $P \approx A$
- $Pz = r$ cheap to compute

Several approaches, we will discuss here

- Block triangular preconditioners
- Incomplete LU factorization
Preconditioners for the Navier-Stokes equations

Block triangular preconditioners

\[
\begin{bmatrix}
F & B^T \\
B & 0
\end{bmatrix}
= \begin{bmatrix}
I & 0 \\
BF^{-1} & I
\end{bmatrix}
\begin{bmatrix}
F & 0 \\
0 & S
\end{bmatrix}
\begin{bmatrix}
I & F^{-1}B^T \\
0 & I
\end{bmatrix}
\]

\[
P_t = \begin{bmatrix}
F & B^T \\
0 & S
\end{bmatrix}, \quad S = -BF^{-1}B^T \text{(Schur complement matrix)}
\]

Subsystem solve

\[
Sz_2 = r_2, \quad Fz_1 = r_1 - B^Tz_2
\]

• In practice \(F^{-1}\) and \(S^{-1}\) are expensive.
• \(F^{-1}\) is obtained by an approximate solve
• \(S\) is first approximated and then solved inexactly
Preconditioners for the Navier-Stokes equations

Well-known approximations to Schur complement

- **Pressure convection diffusion (PCD)** [Kay, Login and Wathen, 2002]
  \[ S \approx -A_p F_p^{-1} Q_p \]

- **Least squares commutator (LSC)** [Elman, Howle, Shadid, Silvester and Tuminaro, 2002]
  \[ S \approx -(BQ^{-1} B^T)(BQ^{-1} FQ^{-1} B^T)^{-1}(BQ^{-1} B^T) \]

- **Augmented Lagrangian approach (AL)** [Benzi and Olshanskii, 2006]

- Convergence independent of the mesh size and mildly dependent on Reynolds number
- Require iterative solvers (Multigrid) for the (1,1) and (2,2) blocks
- Require extra operators
Preconditioners for the Navier-Stokes equations

**Incomplete LU preconditioners**

\[ A = LD^{-1}U + R, \]

\((LD^{-1}U)_{i,j} = a_{i,j}\) for \((i,j) \in S\),

where \(R\) consist of dropped entries that are absent in the index set \(S(i,j)\). [Meijerink and van der Vorst, 1977]

- dropping based on position, \(S = \{(i,j) | a_{ij} \neq 0\}\) (positional dropping)
- dropping based on numerical size (Threshold dropping)

- Simple to implement,
- Computation is inexpensive
- *Inaccuracies and instabilities,*
Preconditioners for the Navier-Stokes equations

**Pivoting**
- Prevent zero diagonal, small pivots
- *A priori estimation of the memory required to store the matrix a difficult task*

**A priori reordering/renumbering**
- Improve profile and bandwidth of the matrix
- Minimizes dropped entries in ILU

**Well-known renumbering schemes**
- Cuthill McKee renumbering (CMK) [Cuthill McKee - 1969]
- Sloan renumbering [Sloan - 1986]
- Minimum degree renumbering (MD) [Tinney and Walker - 1967]
Preconditioners for the Navier-Stokes equations

ILUPACK

Developed by Matthias Bollhöfer and his team. Gives robust and stable ILU preconditioner
- Static reordering [RCM, AMD etc]
- Scaling, pivoting
- Inverse triangular factors are kept bounded.
- The above steps are performed recursively
- Krylov method is applied to solve the preconditioned system

Preconditioners for the Navier-Stokes equations

New priori ordering scheme

Two Steps:
- Renumbering of grid points: Grid points are renumbered with Sloan or Cuthill McKee algorithms
- Reordering of unknowns

**p-last ordering**, first all the velocity unknowns are ordered followed by pressure unknowns. Usually it produces a large profile but avoids breakdown of $LU$ decomposition.

**p-last per node ordering**, The velocity unknowns are ordered followed by pressure unknowns per node (Optimal profile but breakdown of ILU may occur, therefore pivoting required)
Preconditioners for the Navier-Stokes equations

**p-last per level reordering**

Levels?

Q2–Q1 finite element subdivision

---

Rehman, Vuik and Segal

Solution of the Navier-Stokes problem
Preconditioners for the Navier-Stokes equations

p-last per level reordering

First we take all the velocities of level 1, then all pressures of level 1. Next we do the same for level 2, and repeat this process for all nodes.

- The profile is hardly enlarged
- Zero pivots becomes nonzero, therefore no pivoting required

**Choice of first level:** The first level may be defined as a point, or even a line in $\mathbb{R}^2$ or a surface in $\mathbb{R}^3$. 
Preconditioners for the Navier-Stokes equations

p-last per level reordering

Remark: The ILU decomposition does not breakdown if there is at least one nonzero connection between a velocity and pressure unknown. In each level, velocity unknowns must be followed by pressure unknowns.
Preconditioners for the Navier-Stokes equations

Some features of SILU preconditioner

1. Fill-in based on the connectivity in the finite element grid
2. Extra-fill in
3. Lumping of positive off-diagonal entries
4. Artificial compressibility
Numerical Experiments

Flow domains

- **Channel flow**  The Poiseuille channel flow in a square domain $(-1, 1)^2$ with a parabolic inflow boundary condition and the natural outflow condition having the analytic solution: $u = 1 - y^2; \ v = 0; \ p = 2\nu x$

- **Backward facing step**

Q2-Q1 finite element discretization [Taylor, Hood - 1973]

Q2-P1 finite element discretization [Crouzeix, Raviart - 1973]
Numerical experiments

Renumbering/Reordering used in direct methods

The reordering methods helps in minimizing storage in band and envelope storage scheme.

- Bandwidth($\mathcal{A}$) = $\max_i \{ \beta_i(\mathcal{A}), 1 \leq i \leq n \}$
- Profile($\mathcal{A}$) = $\sum_{i=1}^{n} \beta_i(\mathcal{A})$

$16 \times 16$ channel flow with Q2-Q1 discretization

Profile = 52195, Bandwidth = 570
p–last ordering with lexicographic numbering

Profile = 31222, Bandwidth = 212
p–last per level ordering with Sloan renumbering

Profile = 47468, Bandwidth = 160
p–last per level ordering with Cuthill McKee renumbering
Numerical experiments

Renumbering/Reordering used in direct methods
Profile and bandwidth reduction in the backward facing step with Q2-Q1 discretization

<table>
<thead>
<tr>
<th>Grid</th>
<th>Profile reduction</th>
<th>Bandwidth reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sloan</td>
<td>Cuthill-McKee</td>
</tr>
<tr>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 × 12</td>
<td>0.37</td>
<td>0.61</td>
</tr>
<tr>
<td>8 × 24</td>
<td>0.28</td>
<td>0.54</td>
</tr>
<tr>
<td>16 × 48</td>
<td>0.26</td>
<td>0.5</td>
</tr>
<tr>
<td>32 × 96</td>
<td>0.25</td>
<td>0.48</td>
</tr>
</tbody>
</table>
Numerical experiments

**Stokes Problem in a square domain with BiCGSTAB, accuracy $= 10^{-6}$, Sloan renumbering**

<table>
<thead>
<tr>
<th>Grid size</th>
<th>$Q2 - Q1$</th>
<th>$Q2 - P1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>p-last</td>
<td>p-last per level</td>
</tr>
<tr>
<td>16 $\times$ 16</td>
<td>36(0.11)</td>
<td>25(0.09)</td>
</tr>
<tr>
<td>32 $\times$ 32</td>
<td>90(0.92)</td>
<td>59(0.66)</td>
</tr>
<tr>
<td>64 $\times$ 64</td>
<td>255(11.9)</td>
<td>135(6.7)</td>
</tr>
<tr>
<td>128 $\times$ 128</td>
<td>472(96)</td>
<td>249(52)</td>
</tr>
</tbody>
</table>
Convergence of the ILU preconconditioned Bi-CGSTAB for the Stokes Problem in a backward facing domain with an accuracy $= 10^{-6}$

<table>
<thead>
<tr>
<th>Grid</th>
<th>$Q2 - Q1$</th>
<th>$Q2 - P1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sloan</td>
<td>Cuthill-McKee</td>
</tr>
<tr>
<td>8x24</td>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>16x48</td>
<td>22</td>
<td>32</td>
</tr>
<tr>
<td>32x96</td>
<td>59</td>
<td>65</td>
</tr>
<tr>
<td>64x192</td>
<td>172</td>
<td>285</td>
</tr>
</tbody>
</table>
Numerical experiments

Effect of grid increase (Left) and Reynolds number (Right) on inner iterations for the Navier-Stokes backward facing step problem with $\textit{accuracy} = 10^{-2}$ using the p-last-level reordering.
Comparison of the preconditioners using MG solver for (1,1),(2,2) blocks of PCD and LSC preconditioner with Bi-CGSTAB and accuracy $= 10^{-4}$ (IFISS)

<table>
<thead>
<tr>
<th>Grid</th>
<th>PCD</th>
<th>SILU</th>
<th>LSC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter.</td>
<td>Mflops</td>
<td>Iter.</td>
</tr>
<tr>
<td>Re=100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 x 24</td>
<td>40</td>
<td>3.7</td>
<td>9</td>
</tr>
<tr>
<td>16 x 48</td>
<td>36</td>
<td>15.3</td>
<td>13</td>
</tr>
<tr>
<td>32 x 96</td>
<td>39</td>
<td>70.9</td>
<td>21</td>
</tr>
<tr>
<td>64 x 192</td>
<td>61</td>
<td>458</td>
<td>55</td>
</tr>
</tbody>
</table>

64 x 192 grid with increasing Re

| Re = 200 | 48 | 259 | 17 | 241 |
| Re = 300 | 50 | 269 | 19 | 270 |
| Re = 400 | 48 | 259 | 29 | 412 |

Extra-fillin: 16 iterations, 155 flops
Numerical experiments

Comparison with ILUPACK-Stokes Problem in a backward facing domain with an accuracy $= 10^{-6}$, Q2-Q1 elements

- ![Graph showing the number of GMRES(20) iterations vs. grid size for different methods.]
- ![Graph showing CPU time vs. grid size for different methods.]

64x192 grid (8 iterations, const. time(s) = 90, solver time(s) = 7, gain factor = 6)

Rehman, Vuik and Segal, Solution of the Navier-Stokes problem
Numerical experiments

Comparison with ILUPACK-Stokes Problem in a backward facing domain with an accuracy $= 10^{-6}$, Q2-Q1 elements

64x192 grid (8 iterations, const. time(s) = 90, solver time(s)=7, gain factor =6)

Rehman, Vuik and Segal Solution of the Navier-Stokes problem
Conclusions

- A new scheme for the renumbering of grid points and reordering of unknowns is introduced that prevents the break down of the ILU preconditioner and leads to faster convergence of Krylov subspace methods.
- Improves profile and bandwidth of a matrix
- Sloan with p-last per level reordering leads to best results for the Taylor Hood and Crouzeix Raviart elements.
- Since the block preconditioners are independent of the grid size and weakly dependent of the Reynolds number there performance can be better than the S ILU preconditioners for large grid sizes and large Reynolds numbers
- Varying stretched grids
- Testing the preconditioner for the problems with high Reynolds number (SUPG implementation)
- Use of SILU preconditioner in 3D
Thank you for your attention!