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

Outline

Introduction

The incompressible Navier-Stokes equation

$$-\nu\nabla^2\mathbf{u} + \mathbf{u}.\nabla\mathbf{u} + \nabla p = f \quad \text{in} \quad \Omega$$

$$\nabla.\mathbf{u} = 0 \quad \text{in} \quad \Omega.$$

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

$$\mathbf{u} = \mathbf{w} \;\; on \;\; \partial\Omega_D, \quad \nu\frac{\partial\mathbf{u}}{\partial\mathbf{n}} - \mathbf{n}p = 0 \;\; on \;\; \partial\Omega_N.$$

Finite element discretization

Weak formulation

$$\tilde{X} = (H_E^1(\Omega))^d, \quad X = (H_0^1(\Omega))^d, \quad M = L^2(\Omega)$$

Find $\mathbf{u} \in \tilde{X}$ and $p \in M$

$$\nu \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v} d\Omega + \int_\Omega (\mathbf{u}.\nabla \mathbf{u}).\mathbf{v} d\Omega - \int_\Omega p(\nabla.\mathbf{v}) d\Omega = \int_\Omega \mathbf{f}.\mathbf{v} d\Omega, \quad \forall \mathbf{v} \in X$$

$$\int_\Omega q(\nabla.\mathbf{u}) d\Omega = 0, \quad \forall q \in M$$

Finite element discretization

Discrete weak formulation

$$\tilde{X}_h = (H_E^1(\Omega))^d, \quad X_h = (H_0^1(\Omega))^d, \quad M_h = L^2(\Omega)$$

Find $\mathbf{u_h} \in \tilde{X}_h$ and $p_h \in M_h$

$$\nu \int_\Omega \nabla \mathbf{u_h} : \nabla \mathbf{v_h} d\Omega + \int_\Omega (\mathbf{u_h}.\nabla \mathbf{u_h}).\mathbf{v_h} d\Omega - \int_\Omega p_h (\nabla.\mathbf{v_h}) d\Omega = \int_\Omega \mathbf{f}.\mathbf{v_h} d\Omega, \quad \forall \mathbf{v_h} \in X_h,$$

$$\int_\Omega q_h (\nabla.\mathbf{u_h}) d\Omega = 0 \quad \forall q_h \in M_h.$$

Matrix notation

$$A\mathbf{u} + N(\mathbf{u}) + B^T p = \mathbf{f}$$

$$B\mathbf{u} = 0.$$

Linearization

Stokes problem

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

Picard's method

$$-\nu\Delta\mathbf{u}^{(k+1)} + (\mathbf{u}^{(k)}.\nabla)\mathbf{u}^{(k+1)} + \nabla p^{(k+1)} = \mathbf{f}$$
$$\nabla.\mathbf{u}^{(k+1)} = 0$$

Newton's method

$$-\nu\Delta\mathbf{u}^{k+1} + \mathbf{u}^{k+1}.\nabla\mathbf{u}^{k} + \mathbf{u}^{k}.\nabla\mathbf{u}^{k+1} + \nabla p^{k+1} = \mathbf{f} + \mathbf{u}^{k}.\nabla\mathbf{u}^{k},$$
$$\nabla.\mathbf{u}^{k+1} = 0.$$

Linear system

Matrix form after linearization

$$\begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} \quad \text{or } \mathcal{A}x = b$$

- $F \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{m \times n}$, $f \in \mathbb{R}^n$ 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

Solution techniques

Direct methods

To solve $\mathcal{A}x = b$,
factorize $\mathcal{A}$ 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 $\mathcal{A} = M - N$
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 $\mathcal{K}_n(\mathcal{A}, b)$, where $\mathcal{K}_n = \langle b, \mathcal{A}b, \mathcal{A}^2 b, ..., \mathcal{A}^{n-1} b\rangle$ of dimension n.

CGNR	[Paige and Saunders - 1975]
QMR	[Freund and Nachtigal - 1991]
CGS	[Sonneveld - 1989]
Bi-CGSTAB	[van der Vorst - 1992]
GMRES	[Saad and Schultz - 1986]
GMRESR	[van der Vorst and Vuik - 1994]
GCR	[Eisenstat, Elman and Schultz - 1986]

- 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 $\mathcal{A}x = b$ is transformed into $P^{-1}\mathcal{A}x = P^{-1}b$ such that

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

Several approaches, we will discuss here

- Block triangular preconditioners
- Incomplete LU factorization

Preconditioners for the Navier-Stokes equations

Block triangular preconditioners

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

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

$$\text{Subsystem solve } Sz_2 = r_2, \quad Fz_1 = r_1 - B^T z_2$$

- 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

$\mathcal{A} = LD^{-1}U + R$,
$(LD^{-1}U)_{i,j} = a_{i,j}$ for $(i,j) \in \mathcal{S}$,
where R consist of dropped entries that are absent in the index set $\mathcal{S}(i,j)$. [Meijerink and van der Vorst, 1977]
- dropping based on position, $\mathcal{S} = \{(i,j) | \; a_{ij} \neq 0\}$ (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]

[Dutto-1993, Benzi-1997, Duff and Meurant-1989, Wille-2004, Chow and Saad - 1997]

Preconditioners for the Navier-Stokes equations

ILUPACK

Devloped by Matthias Bollhöfer and his team. Gives robust and stable ILU preconditioner

- Static reordering [RCM, AMD etc]
- Scaling, pivoting
- Inverse traingular factors are kept bounded.
- The above steps are perfomed recursively
- Krylov method is applied to solve the preconditioned system

Matthias Bollhöfer, Yousef Saad. Multilevel Preconditioners Constructed From Inverse-Based ILUs, *SIAM Journal on Scientific Computing, 27* , 5(2005), 1627-1650

Preconditioners for the Navier-Stokes equations

New priori ordering scheme

Two Steps:

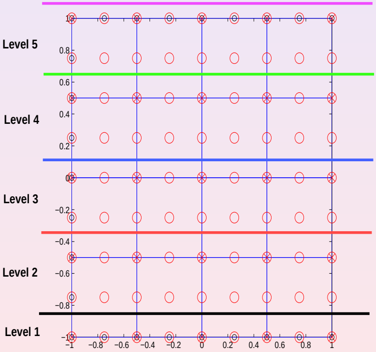
- Renumbering of grid points: Grid points are renumbered with Sloan or Cuthill McKee algorithms
- Reordering of unknowns

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

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.

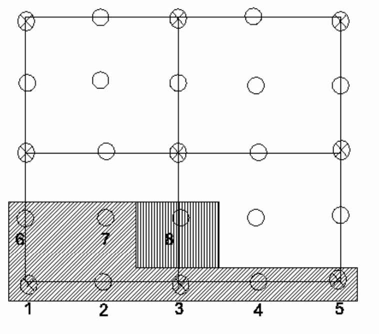
- The profile is hardly enlarged
- Zero pivots becomes nonzero, therefore no pivoting required

Choice of first level: The first level may be defined as a point, or even a line in R^2 or a surface in 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

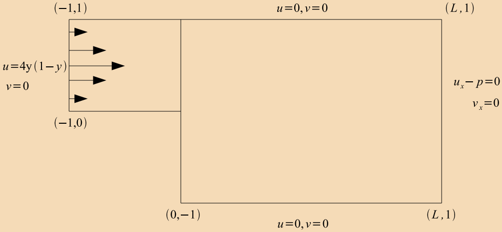
Some features of SILU preconditioner

1. Fill-in based on the connectivity in the finite element grid
2. Extra-fill in
3. Lumping of positve 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: $u = 1 - y^2; \quad v = 0; \quad p = 2\nu x$
- **Backward facing step**



- Q2-Q1 finite element discretization [Taylor, Hood - 1973]
- Q2-P1 finite element discretization [Crouzeix, Raviart - 1973]

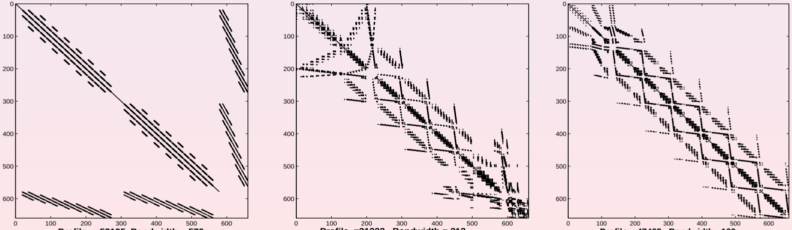
Numerical experiments

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($\mathcal{A}$)=$\sum_{i=1}^{n} \beta_i(\mathcal{A})$

16×16 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

Grid	Profile reduction		Bandwidth reduction	
-	Sloan	Cuthill-McKee	Sloan	Cuthill-McKee
4×12	0.37	0.61	0.18	0.17
8×24	0.28	0.54	0.13	0.08
16×48	0.26	0.5	0.11	0.04
32×96	0.25	0.48	0.06	0.02

Numerical experiments

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

	$Q2-Q1$		$Q2-P1$	
Grid size	p-last	p-last per level	p-last	p-last per level
16×16	36(0.11)	25(0.09)	44(0.14)	34(0.13)
32×32	90(0.92)	59(0.66)	117(1.08)	75(0.80)
64×64	255(11.9)	135(6.7)	265(14)	165(9.0)
128×128	472(96)	249(52)	597(127)	407(86)

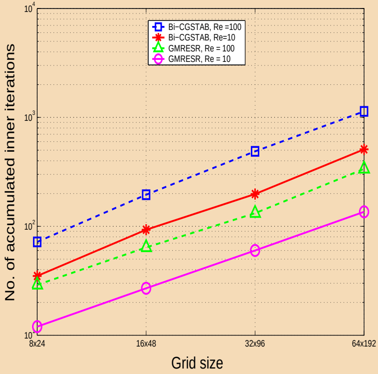
Numerical experiments

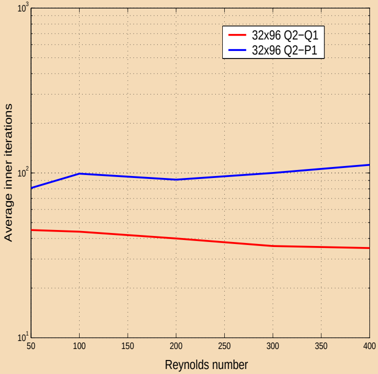
Convergence of the ILU preonconditioned Bi-CGSTAB for the Stokes Problem in a backward facing domain with an *accuracy* $= 10^{-6}$

Grid	$Q2-Q1$		$Q2-P1$	
-	Sloan	Cuthill-McKee	Sloan	Cuthill-McKee
-	Iter.	Iter.	Iter.	Iter.
8x24	9	15	29	97
16x48	22	32	40	288
32x96	59	65	73	1300
64x192	172	285	330	1288

Numerical experiments

Effect of grid increase(Left) and Reynolds number(Right) on inner iterations for the Navier-Stokes backward facing step problem with *accuracy* $= 10^{-2}$ using the p-last-level reordering





Numerical experiments

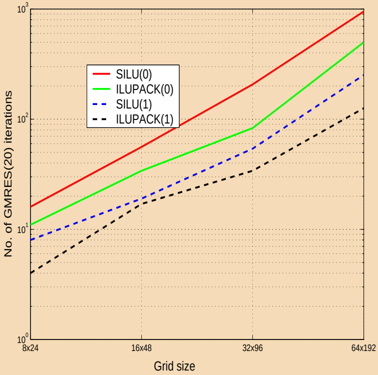
Comparison of the preconditioners using MG solver for (1,1),(2,2) blocks of PCD and LSC preconditioner with Bi-CGSTAB and *accuracy* $= 10^{-4}$ (IFISS)

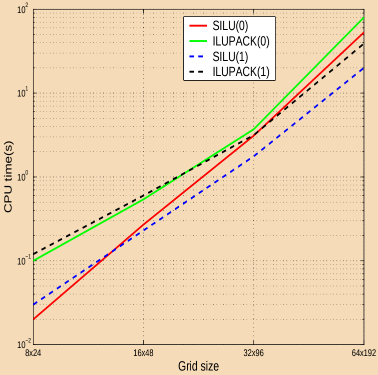
Grid	PCD		SILU		LSC	
Re=100						
	Iter.	Mflops	Iter.	Mflops	Iter.	Mflops
8×24	40	3.7	9	0.6	24	4
16×48	36	15.3	13	3.9	19	14.9
32×96	39	70.9	21	27.5	13	44.4
64×192	61	458	55	297	13	185
64×192 grid with increasing Re						
Re = 200			48	259	17	241
Re = 300			50	269	19	270
Re = 400			48	259	29	412

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

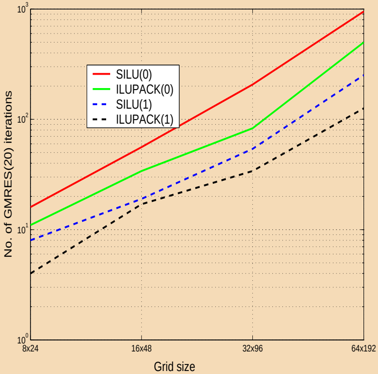


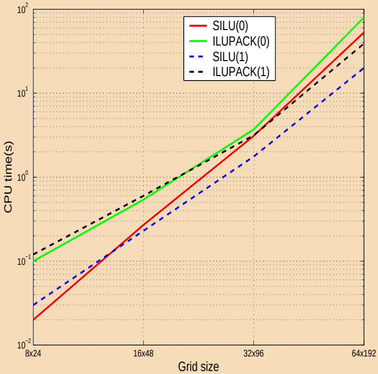


64x192 grid (8 iterations, const. time(s) = 90, solver time(s)=7, gain factor =6)

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)

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 !