Preconditioners for the incompressible Navier-Stokes equations

C. Vuik, M. ur Rehman, and A. Segal

Delft University of Technology

Delft Institute of Applied Mathematics

Delft, The Netherlands.

International Conference On Preconditioning Techniques

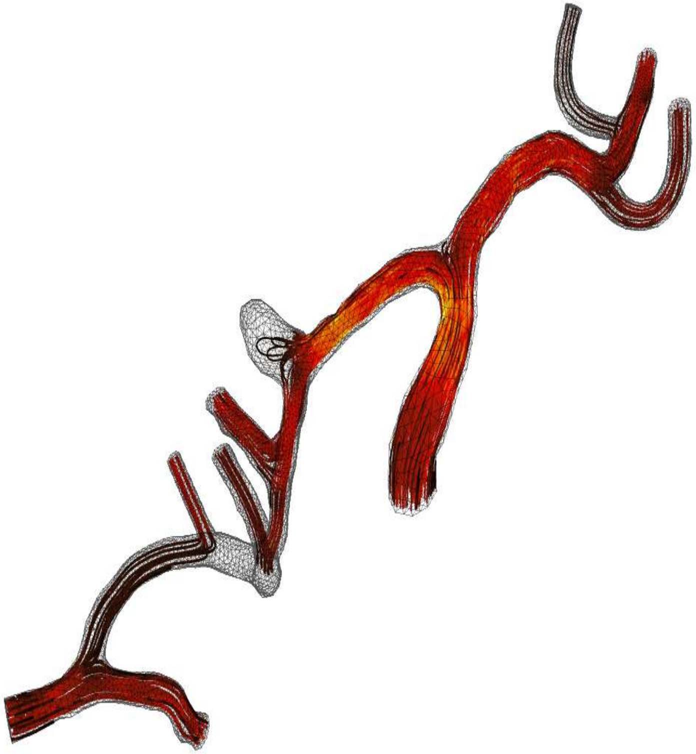
For Scientific And Industrial Applications

August 24-26, 2009 , Hong Kong

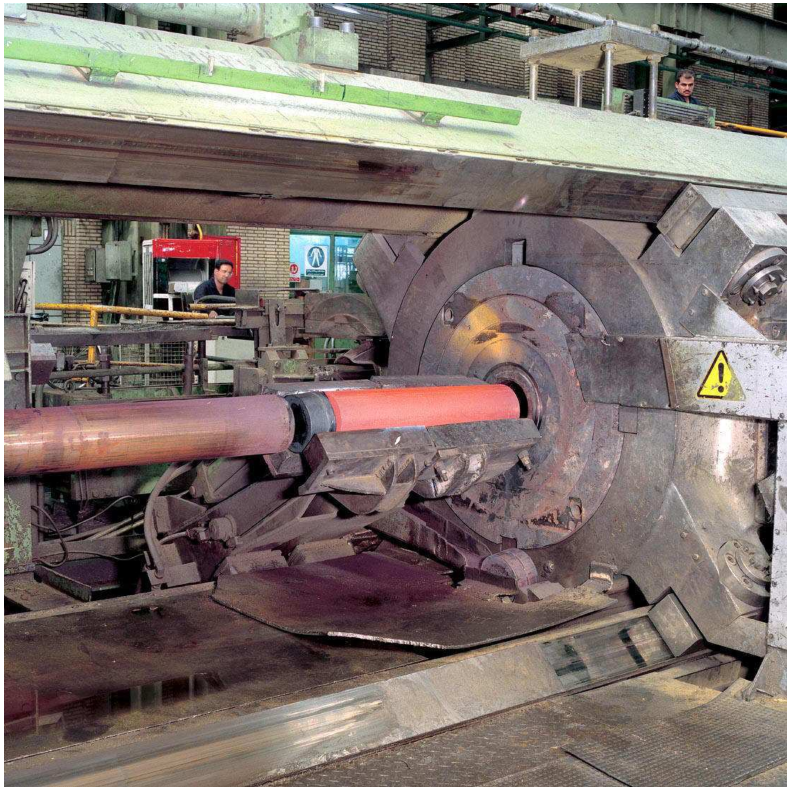
Outline

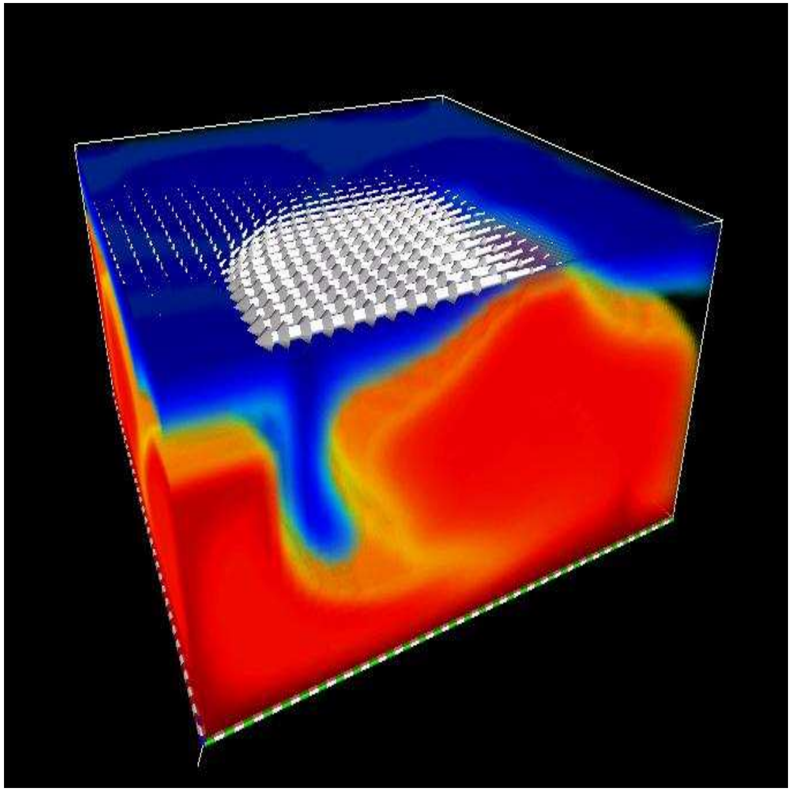
1. Introduction
2. Problem
3. Krylov solvers and preconditioners
4. ILU-type preconditioners
5. Block preconditioners
6. Stokes preconditioners
7. Conclusions

1. Introduction



1. Introduction





2. Problem

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

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

$\mathbf{u}$ is the fluid velocity vector

p is the pressure field

$\nu > 0$ is the kinematic viscosity coefficient ($1/Re$).

$\Omega \subset \mathbf{R}^{2 \text{ or } 3}$ is a bounded domain with the boundary condition:

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

Finite element discretization

Discrete weak formulation:

$$X_h \subset (H_0^1(\Omega))^d, \quad M_h \subset L^2(\Omega)$$

Find $\mathbf{u_h} \in X_h$ 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:

$$\begin{aligned} A\mathbf{u} + N(\mathbf{u}) + B^T p &= \mathbf{f} \\ B\mathbf{u} &= 0. \end{aligned}$$

Finite element discretization

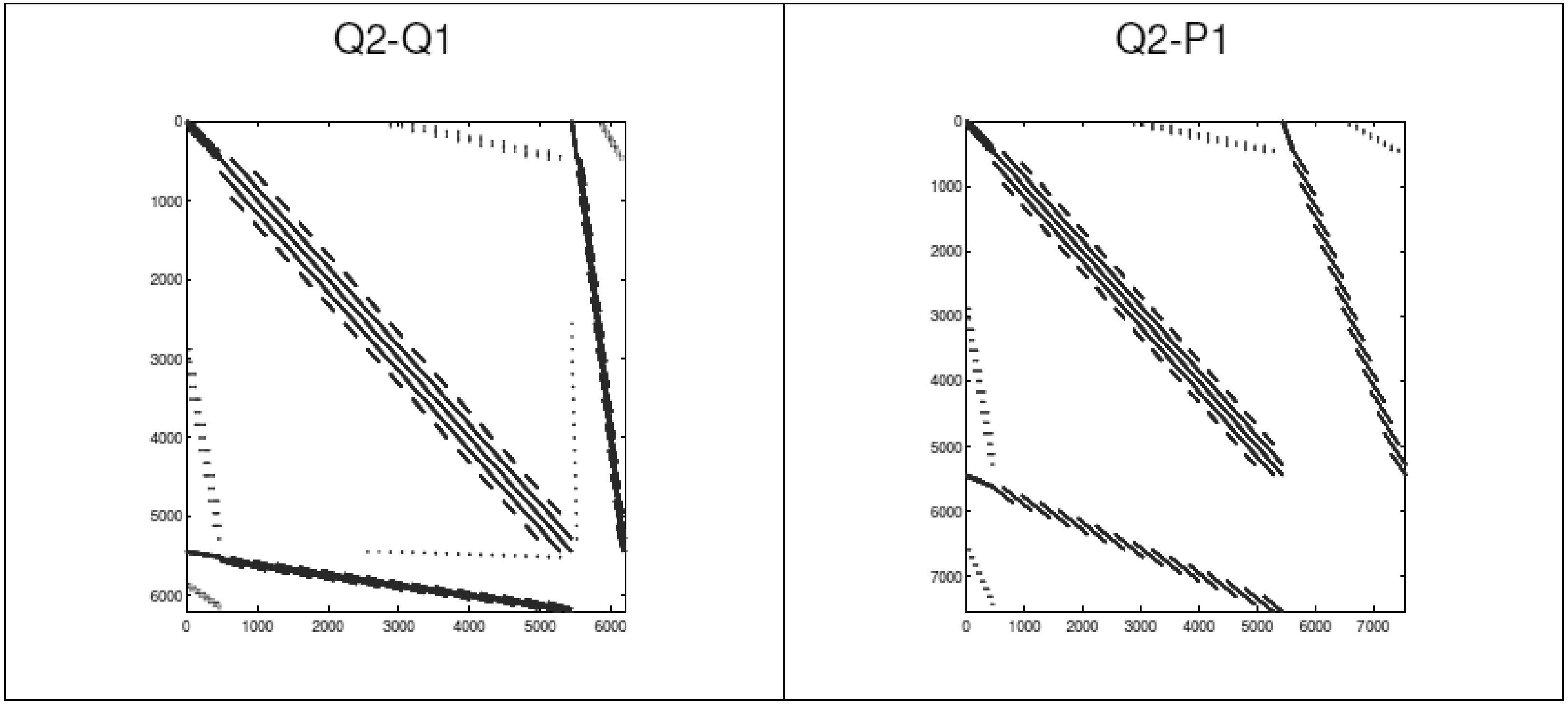
For a unique solution, finite elements must satisfy the LBB condition

$$\inf_{q \in M_h} \sup_{v \in X_h} \frac{(\nabla . \mathbf{v_h}, q_h)}{\|\mathbf{v_h}\|_{X_h} \|q_h\|_{M_h}} \geq \gamma \geq 0.$$

- Taylor-Hood elements [Taylor and Hood, 1973]: Bi-quadratic velocity and bi-linear pressure (Q2-Q1).
- Crouzeix Raviart elements [Crouzeix and Raviart, 1973]: Bi-quadratic velocity, discontinuous linear pressure (Q2-P1).

Finite element discretization

Non-zero pattern of the Stokes matrix with two families of elements



Linear system

Matrix form after linearization:

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

where $F \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{m \times n}$, $f \in \mathbb{R}^n$ and $m \leq n$

- $F = \nu A$ in Stokes problem, A is vector Laplacian matrix
- $F = \nu A + N$ in Picard linearization, N is vector-convection matrix
- $F = \nu A + N + W$ in Newton linearization, W is the Newton derivative matrix
- B is the divergence matrix
- Sparse linear system, Symmetric indefinite (Stokes problem), nonsymmetric otherwise.
- Saddle point problem having large number of zeros on the main diagonal

3. Krylov Solvers and preconditioners

- **Direct method:**

  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 Schemes:**

  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

- **Krylov Subspace Methods:**

  $x_{k+1} = x_k + \alpha_k p_k$

  Some well known methods are

  CGNR[1975], QMR[1991], CGS[1989], Bi-CGSTAB[1992], GMRES[1986], GMRESR[1994], GCR[1986], IDR(s)[2007]

IDR and IDR(s) (Induced Dimension Reduction)

- Sonneveld developed IDR in the 1970's. IDR is a finite termination (Krylov) method for solving nonsymmetric linear systems.
- Analysis showed that IDR can be viewed as Bi-CG combined with linear minimal residual steps.
- This discovery led to the development of first CGS, and later of Bi-CGSTAB (by van der Vorst).

IDR and IDR(s) (continued)

- As a result of these developments the basic IDR-idea was abandoned for the Bi-CG-approach.

- Recently, Sonneveld and van Gijzen discovered that the IDR-approach was abandoned too soon and proposed a generalization of IDR: IDR(s).

- P. Sonneveld and M.B. van Gijzen IDR(s): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations

  SIAM J. Sci. Comput., 31, pp. 1035-1062, 2008

The IDR approach for solving $Ax = b$

Generate residuals $\boldsymbol{r}_n = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_n$ that are in subspaces $\mathcal{G}_j$ of decreasing dimension.

These nested subspaces are related by

$$\mathcal{G}_j = (\boldsymbol{I} - \omega_j \boldsymbol{A})(\mathcal{G}_{j-1} \cap \mathcal{S})$$

where

- $\mathcal{S}$ is a fixed proper subspace of $\mathbb{C}^N$. $\mathcal{S}$ can be taken to be the orthogonal complement of s randomly chosen vectors $\boldsymbol{p}_i, i = 1 \cdots s$.
- The parameters $\omega_j \in \mathbb{C}$ are non-zero scalars.

It can be proved that ultimately $\boldsymbol{r}_n \in \{\boldsymbol{0}\}$ (IDR theorem).

IDR versus Bi-CG

IDR(s) forces the residual to be in an increasingly small subspace, while Bi-CG constructs a residual in an increasingly large subspace. Yet, IDR(s) is closely related to:

- Bi-CGSTAB: IDR(1) and Bi-CGSTAB are mathematically equivalent.
- ML(k)BiCGSTAB (Yeung and Chan, 1999): This method generalizes Bi-CGSTAB using multiple ’shadow residuals’. Mathematically IDR(s) and ML(k)BiCGSTAB differ in the selection of the parameters ω_j.
  IDR(s) uses simpler recurrences, less vector operations and memory than ML(k)BiCGSTAB, and is more flexible (e.g. to avoid break down).

Prototype IDR(s) algorithm.

```
while $\|\boldsymbol{r}_n\| > TOL$ or $n < MAXIT$ do
    for $k = 0$ to $s$ do
        Solve $\boldsymbol{c}$ from $\boldsymbol{P}^H d\boldsymbol{R}_n \boldsymbol{c} = \boldsymbol{P}^H \boldsymbol{r}_n$
        $\boldsymbol{v} = \boldsymbol{r}_n - d\boldsymbol{R}_n \boldsymbol{c}$; $\boldsymbol{t} = \boldsymbol{A}\boldsymbol{v}$;
        if $k = 0$ then
            $\omega = (\boldsymbol{t}^H \boldsymbol{v})/(\boldsymbol{t}^H \boldsymbol{t})$;
        end if
        $d\boldsymbol{r}_n = -d\boldsymbol{R}_n \boldsymbol{c} - \omega \boldsymbol{t}$; $d\boldsymbol{x}_n = -d\boldsymbol{X}_n \boldsymbol{c} + \omega \boldsymbol{v}$;
        $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + d\boldsymbol{r}_n$; $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + d\boldsymbol{x}_n$;
        $n = n + 1$;
        $d\boldsymbol{R}_n = (d\boldsymbol{r}_{n-1} \cdots d\boldsymbol{r}_{n-s})$; $d\boldsymbol{X}_n = (d\boldsymbol{x}_{n-1} \cdots d\boldsymbol{x}_{n-s})$;
    end for
end while
```

More information

More information: http://ta.twi.tudelft.nl/nw/users/gijzen/IDR.html

- P. Sonneveld and M.B. van Gijzen IDR(s): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations

  SIAM J. Sci. Comput., 31, pp. 1035-1062, 2008
- The relation of IDR(s) with Bi-CGSTAB, and how to derive generalizations of Bi-CGSTAB using IDR-ideas can be found in: Bi-CGSTAB as an induced dimension reduction method (with Sleijpen).
- A high quality IDR(s) implementation is described in: An elegant IDR(s) variant that efficiently exploits bi-orthogonality properties.

4. ILU-type Preconditioners

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

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

Several approaches, we will discuss here

- ILU preconditioner
- Preconditioned IDR(s) and Bi-CGSTAB comparison
- Block preconditioners

ILU preconditioners

Incomplete LU preconditioners

$\mathcal{A} = LU - R$, where R consist of dropped entries that are absent in the index set $\mathcal{S}(i,j)$.

$\mathcal{S} = \{(i,j) |\ a_{ij} \neq 0\}$ [Classical ILU by Meijerink and van der Vorst, 1977].

- cheap and easy to implement
- if $\|R\|$ is large, give poor convergence (**reordering**)
- Instability due to large $\|L^{-1}\|$ and $\|U^{-1}\|$
- Pivoting, reordering, scaling and shifted diagonal schemes are used to make ILU factors more stable and effective.

[Elman -1989, Chow and Saad -1997]

SILU preconditioners

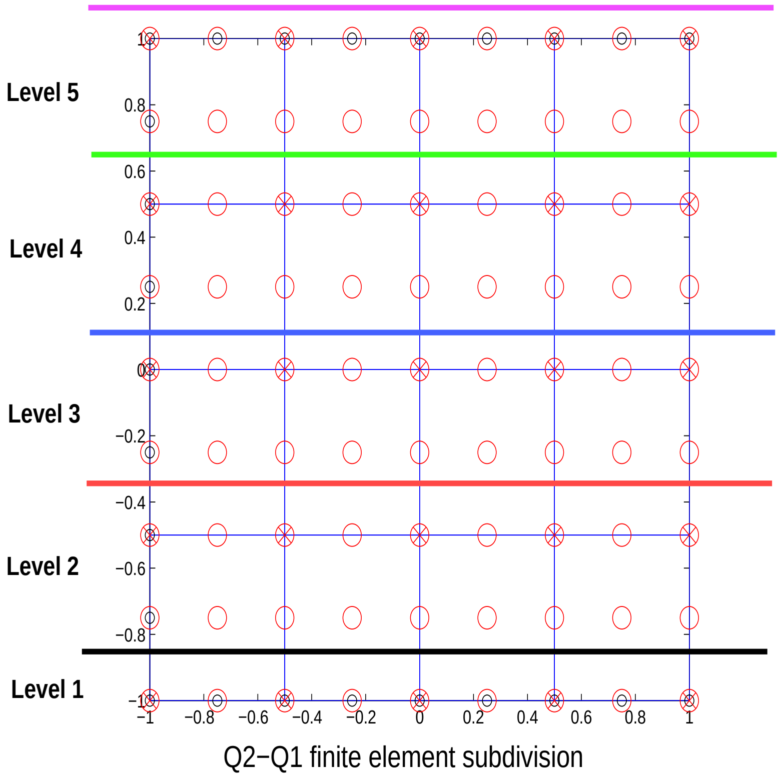
New renumbering Scheme

- Renumbering of grid points:
  - Sloan algorithm [Sloan - 1986]
  - Cuthill McKee algorithms [Cuthill McKee - 1969]
- The unknowns are reordered by p-last or p-last per level methods
  - In **p-last reordering**, first all the velocity unknowns are ordered followed by pressure unknowns. Usually it produces a large profile but avoids breakdown of LU decomposition.
  - In **p-last per level reordering**, unknowns are reordered per level such that at each level, the velocity unknowns are followed by the pressure unknowns.

what are the levels ?

SILU preconditioner

4×4 Q2-Q1 grid



SILU preconditioners

- In direct solver, reordering improves the profile and bandwidth of the matrix.
- Improve the convergence of the ILU preconditioned Krylov subspace method
- Minimizes dropped entries in ILU ($\|\mathcal{A} - \bar{L}\bar{U}\|_F$)
- May give stable factorization ($\|I - \mathcal{A}(\bar{L}\bar{U})^{-1}\|_F$)

[Dutto-1993, Benzi et al 1999, Duff and Meurant-1989, Wille-2004, Bollhöfer and Saad - 2006, Saad -2005]

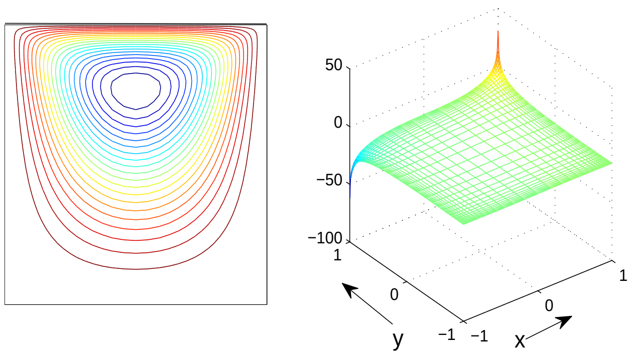
SILU preconditioners

- **Proposition 1:** If we use the p-last ordering and assume that the ILUD decomposition of F exists then the ILUD decomposition exists because every column of B^T contains a non-zero element . Note that if B^T has a zero column then $\mathcal{A}$ is singular.

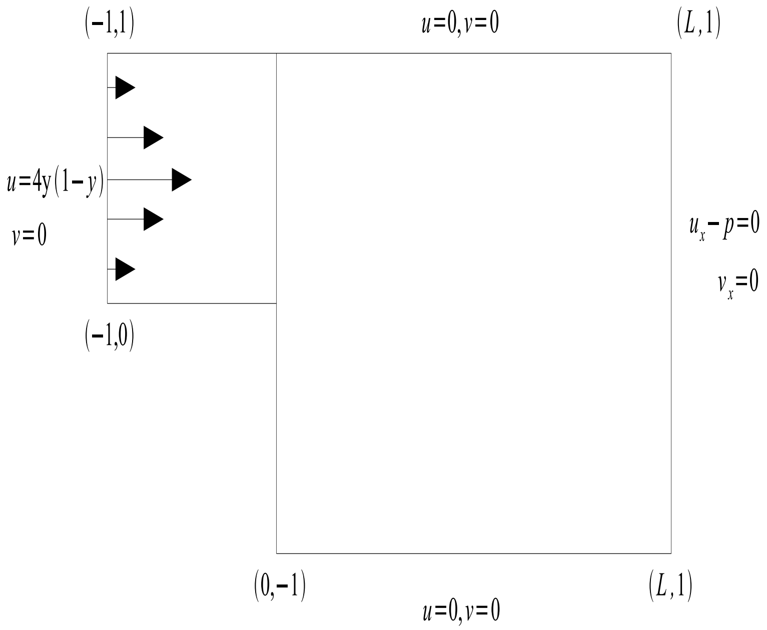
- **Proposition 2:** For an arbitrary ordering we suppose that ILUD decomposition exists for all $j < i$, where the i^{th} row is related to the continuity equation. If the i^{th} (pressure) unknown is preceded by at least one velocity unknown with a non-zero connection to this pressure unknown (so there is one $k < i$ such that $a_{i,k} \neq 0$), then the ILUD decomposition exists and $d_i > 0$.

Numerical experiments (SILU preconditioner)

Driven cavity flow problem



Backward facing step problem

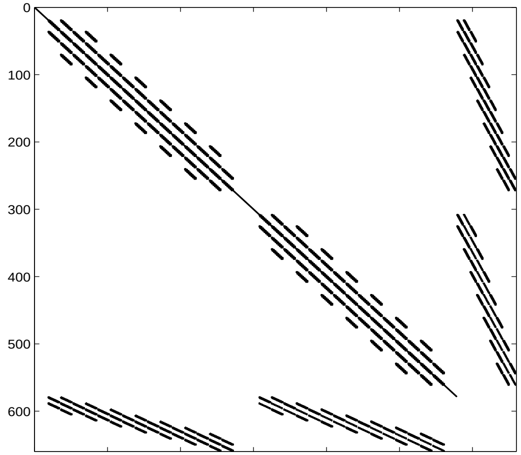


The iteration is stopped if the linear systems satisfy $\frac{\|r^k\|_2}{\|b\|_2} \leq tol$.

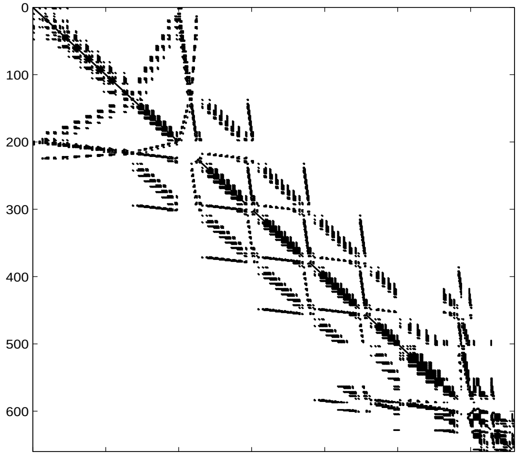
Numerical experiments (SILU preconditioner)

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

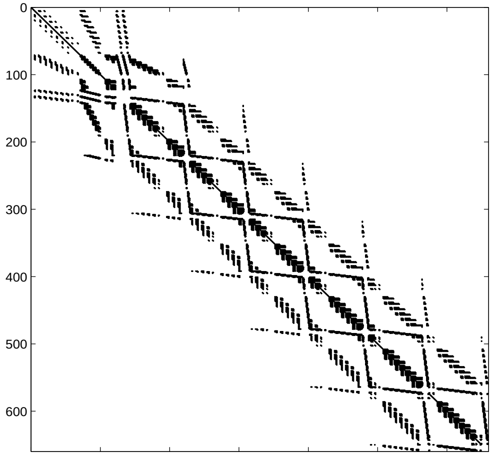
16×16 driven cavity flow with Q2-Q1 discretization



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

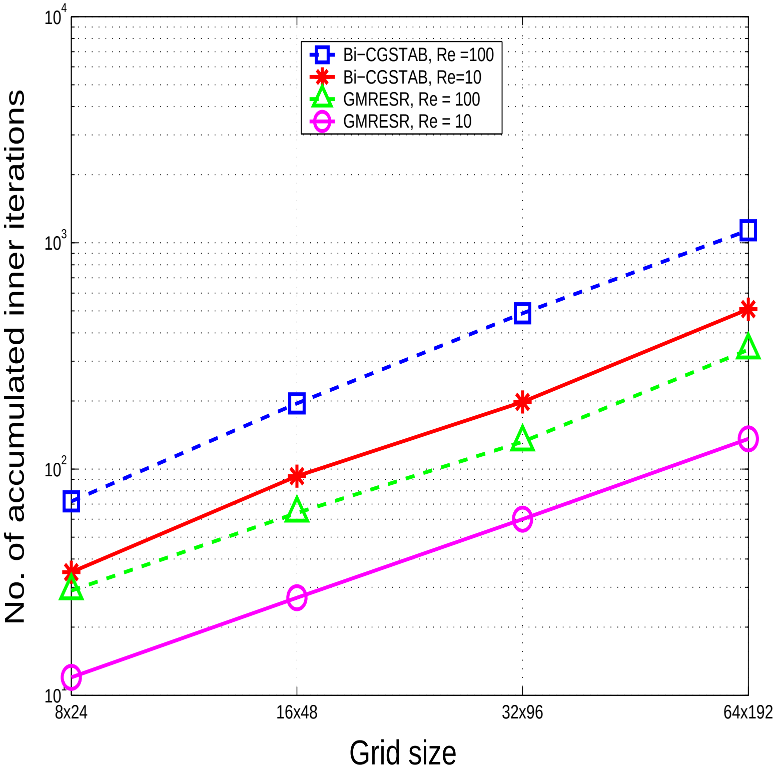
Numerical experiments (SILU preconditioners)

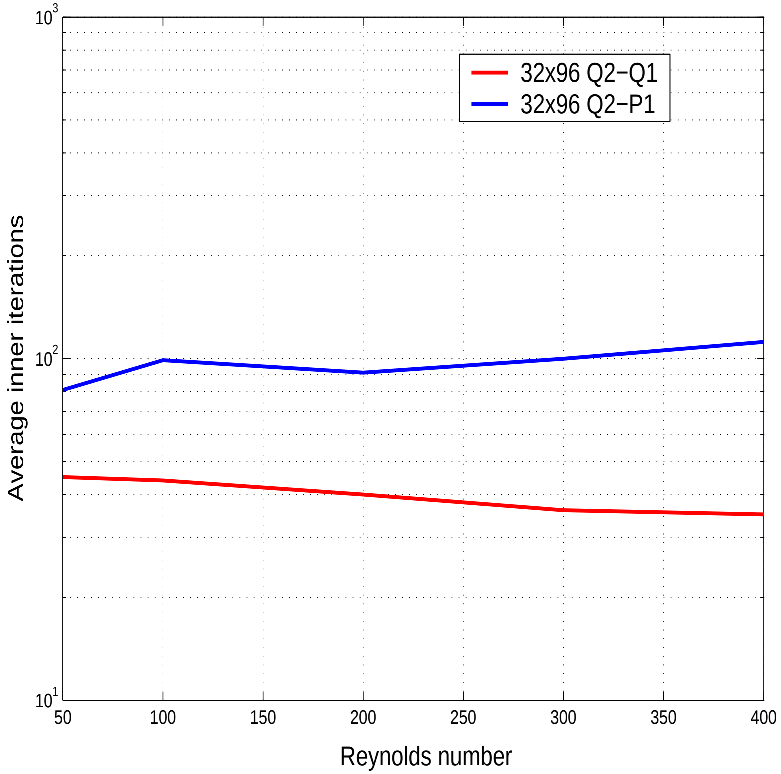
Stokes Problem in a square domain with Bi-CGSTAB,
$accuracy = 10^{-6}$, Sloan renumbering

	$Q2 - Q1$		$Q2 - P1$	
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 (SILU preconditioners)

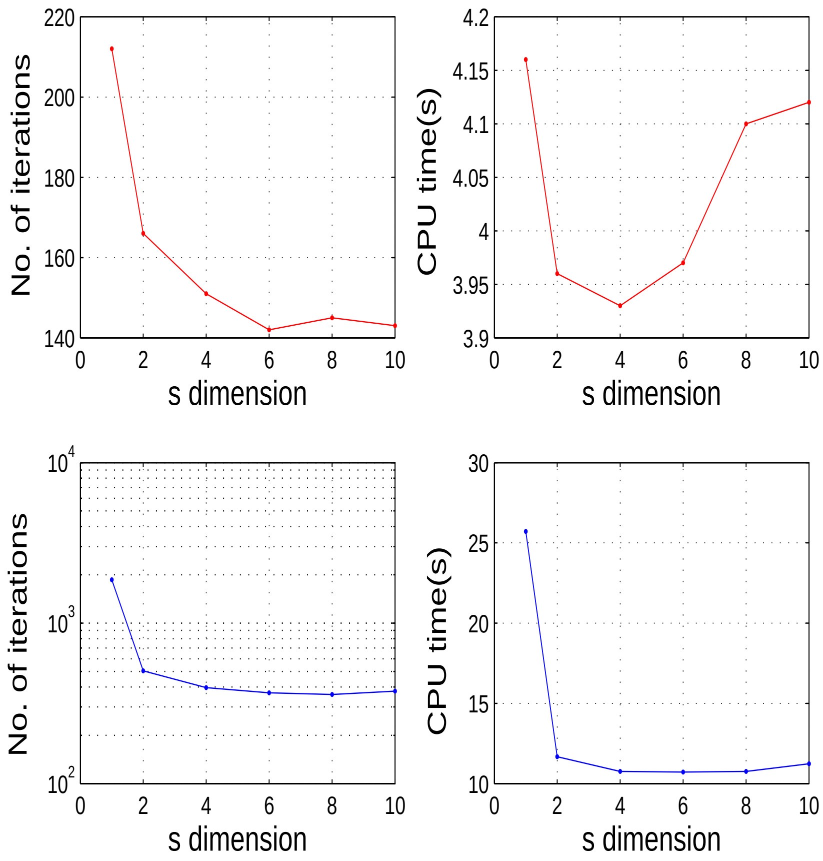
Effect of grid increase(Left) and Reynolds number(Right) on inner iterations for the Navier Stokes backward facing step problem with $accuracy = 10^{-2}$ using the p-last-level reordering





Numerical Experiments (IDR(s))

IDR(s): Top: 32×32, Bottom: 64×64 driven cavity Stokes flow problem



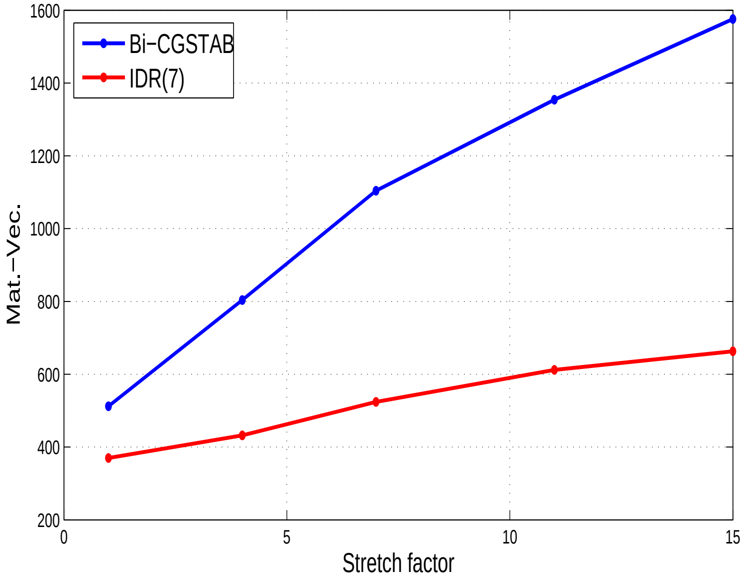
Numerical Experiments (IDR(s) vs Bi-CGSTAB)

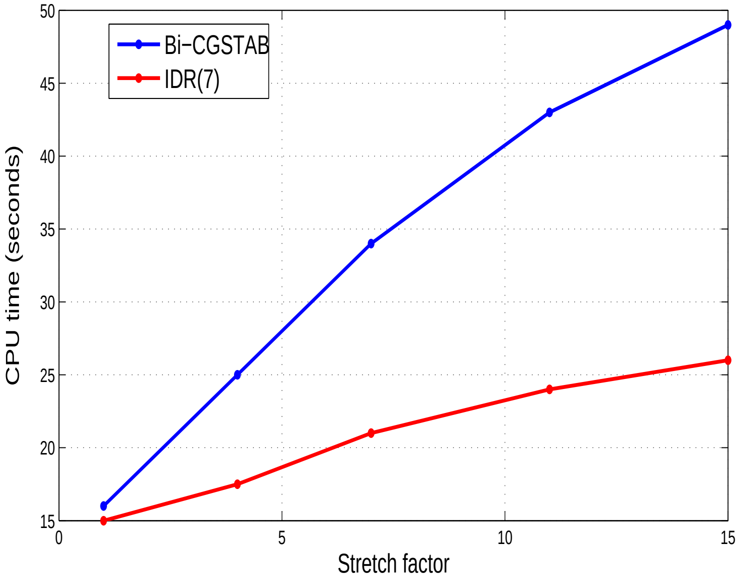
SILU preconditioner: Comparison of iterative methods for increasing grid size for the driven cavity Stokes flow problem.

Grid	Bi-CGSTAB	IDR(4)
	Mat.-Vec. (ts)	Mat.-Vec. (ts)
16×16	38(0.01)	33(0.01)
32×32	90(0.14)	75(0.14)
64×64	214(1.6)	159(1.4)
128×128	512(16)	404(15)
256×256	1386(183)	1032(156)

Numerical Experiments (IDR(s) vs Bi-CGSTAB)

SILU preconditioned: Comparison of iterative methods for increasing stretch factor for the driven cavity Stokes problem.





Numerical Experiments (IDR(s) vs Bi-CGSTAB)

SILU preconditioned: Comparison of iterative methods for the backward facing step Stokes problem.

Grid	Bi-CGSTAB	IDR(s)	
	Mat.-Vec.(ts)	Mat.-Vec.(ts)	s
32×96	214(1.3)	168(1.26)	4
64×96	NC	597(7.7)	4
96×96	NC	933(18)	4
128×96	NC	1105(31)	8

Numerical Experiments (IDR(s) vs Bi-CGSTAB(l))

SILU preconditioned: Comparison of iterative methods

Driven Cavity Stokes problem, stretch factor 10

Grid	Bi-CGSTAB(l)		IDR(s)	
	Mat.-Vec.(ts)	l	Mat.-Vec.(ts)	s
128×128	1104(36.5)	4	638(24.7)	6
256×256	5904(810)	6	1749(307)	8

Channel flow Stokes problem, length 100

Grid	Bi-CGSTAB(l)		IDR(s)	
	Mat.-Vec.(ts)	l	Mat.-Vec.(ts)	s
64×64	1520(12)	4	938(8.7)	8
128×128	NC	6	8224(335)	8

5. Block preconditioners

$$\mathcal{A} = \mathcal{L}_b \mathcal{D}_b \mathcal{U}_b = \begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ BM_l^{-1} & I \end{bmatrix} \begin{bmatrix} F & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & M_u^{-1}B^T \\ 0 & I \end{bmatrix}$$

$M_l = M_u = F$ and $S = -BF^{-1}B^T$ is the Schur-complement matrix.

$$\mathcal{U}_{bt} = \mathcal{L}_b \mathcal{D}_b = \begin{bmatrix} F & B^T \\ 0 & \hat{S} \end{bmatrix}, \quad \mathcal{L}_{bt} = \mathcal{D}_b \mathcal{U}_b = \begin{bmatrix} F & 0 \\ B & \hat{S} \end{bmatrix}.$$

Preconditioners are based on combination of these blocks involve:

$Fz_1 = r_1$ The velocity subsystem

$$S \longrightarrow \hat{S}$$

$\hat{S}z_2 = r_2$ The pressure subsystem

Block preconditioners

Block triangular preconditioners

$$P_t = \mathcal{U}_{bt} = \begin{bmatrix} F & B^T \\ 0 & \hat{S} \end{bmatrix}$$

- Pressure convection diffusion (PCD) [Kay et al, 2002]
  $\hat{S} = -A_p F_p^{-1} Q_p$, Q_p is the pressure mass matrix
- Least squares commutator (LSC) [Elman et al, 2002]
  $\hat{S} = -(BQ_u^{-1}B^T)(BQ_u^{-1}FQ_u^{-1}B^T)^{-1}(BQ_u^{-1}B^T)$, Q_u is the velocity mass matrix
- Augmented Lagrangian approach (AL) [Benzi and Olshanskii, 2006]
  F is replaced by $F_\gamma = F + \gamma B W^{-1} B^T$
  $\hat{S}^{-1} = -(\nu \hat{Q}_p^{-1} + \gamma W^{-1}), \; W = \hat{Q}_p$

Block preconditioners

SIMPLE-type preconditioners[Vuik et al-2000]

SIMPLE	SIMPLER
$z = \mathcal{U}_b^{-1}\mathcal{L}_{bt}^{-1} r$	$z = \mathcal{U}_{bt}^{-1}\mathcal{L}_b^{-1} r$
	$z = z + \mathcal{U}_b^{-1}\mathcal{L}_{bt}^{-1}(r - \mathcal{A}z)$
$M_u = D$	$M_l = M_u = D$, $D = diag(F)$
$\hat{S} = BD^{-1}B^T$	$\hat{S} = BD^{-1}B^T$
One Poisson Solve	Two Poisson solves
One velocity solve	Two velocity solves

Lemma: In the SIMPLER preconditioner/algorithm, both variants (one or two velocity solves) are identical .

Improvements in SIMPLE-type preconditioners

We use approximate solvers for subsystems, so flexible Krylov solvers are required (GCR, FGMRES, GMRESR)

- Relaxation parameter
- hSIMPLER
- MSIMPLER

Improvements in SIMPLE(R) preconditioners

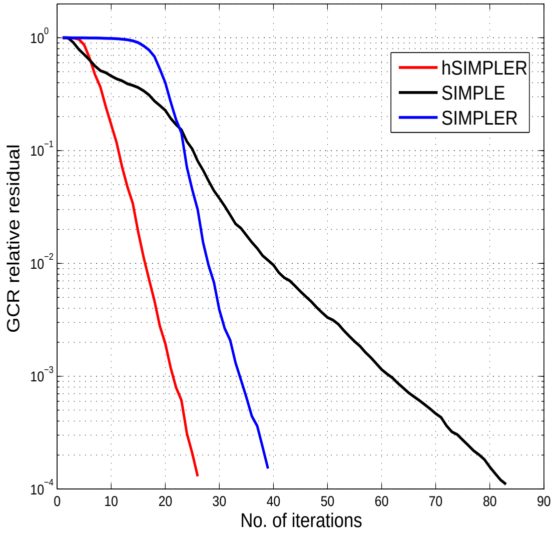
Relaxation parameter:

- Under-relaxation is well-known in SIMPLE-type methods.
- In SIMPLE preconditioner, velocity relaxation has no effect on the convergence, therefore only pressure is under-relaxed by a factor ω.
  $p = p^* + \omega \delta p$, where ω is chosen between 0 and 1.
- ω has no effect on convergence with SIMPLER due to extra pressure correction step.
- Faster convergence is achieved in some cases.
- Choice of ω is currently based on trial an error.

Improvements in SIMPLE(R) preconditioners

hSIMPLER preconditioner:

In hSIMPLER (hybrid SIMPLER), first iteration of Krylov method preconditioned with SIMPLER is done with SIMPLE and SIMPLER is employed afterwards.



- Faster convergence than SIMPLER
- Effective in the Stokes problem

Improvements in SIMPLE(R) preconditioners

MSIMPLER preconditioner:

Making the following changes in SIMPLER leads to the MSIMPLER preconditioner.

LSC: $\hat{S} \approx -(B\hat{Q_u}^{-1}B^T)(B\hat{Q_u}^{-1}\underbrace{F\hat{Q_u}^{-1}}B^T)^{-1}(B\hat{Q_u}^{-1}B^T)$

assuming $F\hat{Q_u}^{-1} \approx I$ (time dependent problems with a small time step)

$$\hat{S} = -B\hat{Q_u}^{-1}B^T$$

MSIMPLER uses this approximation for the Schur complement and updates scaled with $\hat{Q_u}^{-1}$.

-Convergence better than other variants of SIMPLE

-Cheaper than SIMPLER (in construction) and LSC (per iteration)

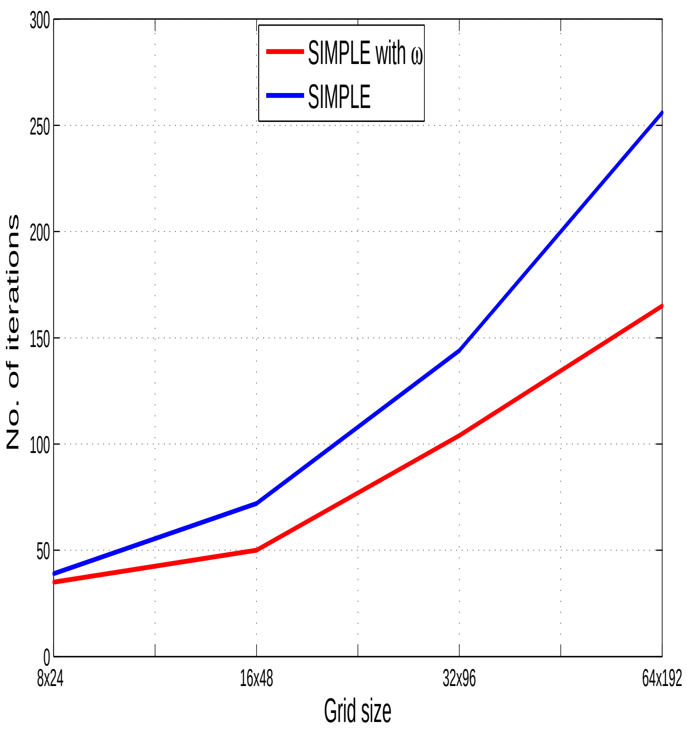
Numerical Experiments (SIMPLE-type preconditioners)

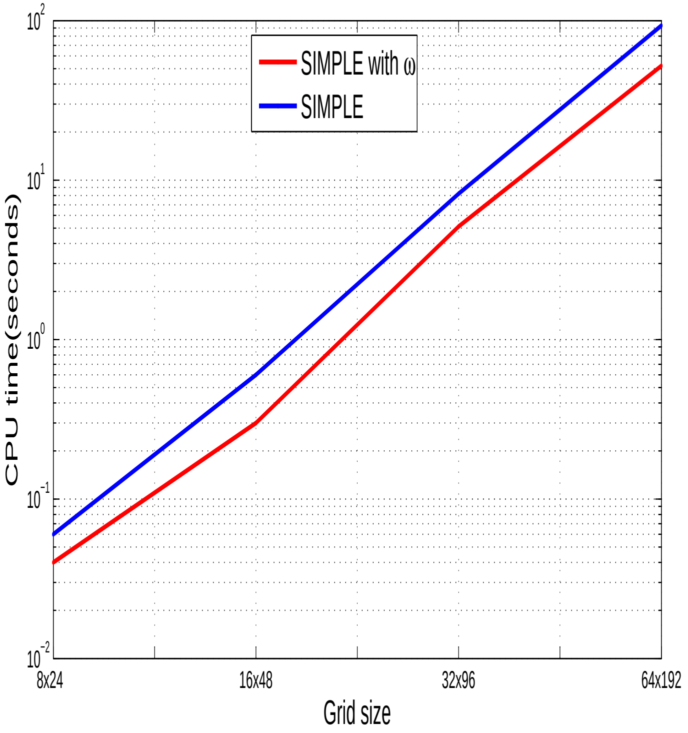
Stokes backward facing step solved with preconditioned GCR(20) with $accuracy$ of 10^{-6}, PCG used as an inner solver (SEPRAN), Blue: Low inner accuracy , Red: High inner accuracy

Grid	SIMPLE	SIMPLER	hSIMPLER	MSIMPLER
	iter. (ts)	iter. (ts)	iter. (ts)	iter. (ts)
8×24	39(0.06)	26(0.05)	19(0.03)	11(0.02)
	37(0.14)	19(0.07)	17(0.06)	12(0.05)
16×46	72(0.6)	42(0.5)	31(0.34)	12(0.1)
	68(1.94)	30(0.86)	24(0.68)	15(0.44)
32×96	144(8.2)	NC	44(5.97)	16(0.9)
	117(34)	114(32)	37(10.6)	20(5.75)
64×192	256(93)	NC	89(141)	23(8.5)
	230(547)	NC	68(161)	25(60)

Numerical Experiments (SIMPLE type preconditioners)

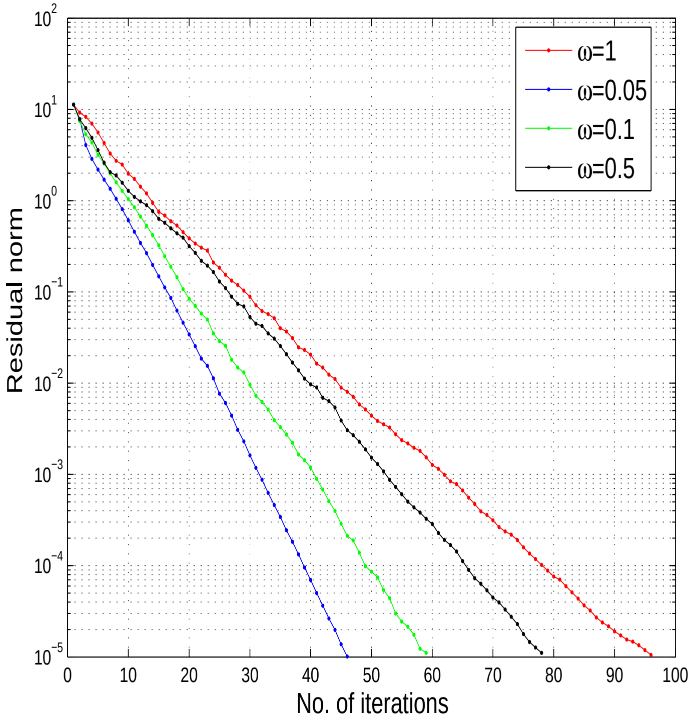
SIMPLE with relaxation parameter

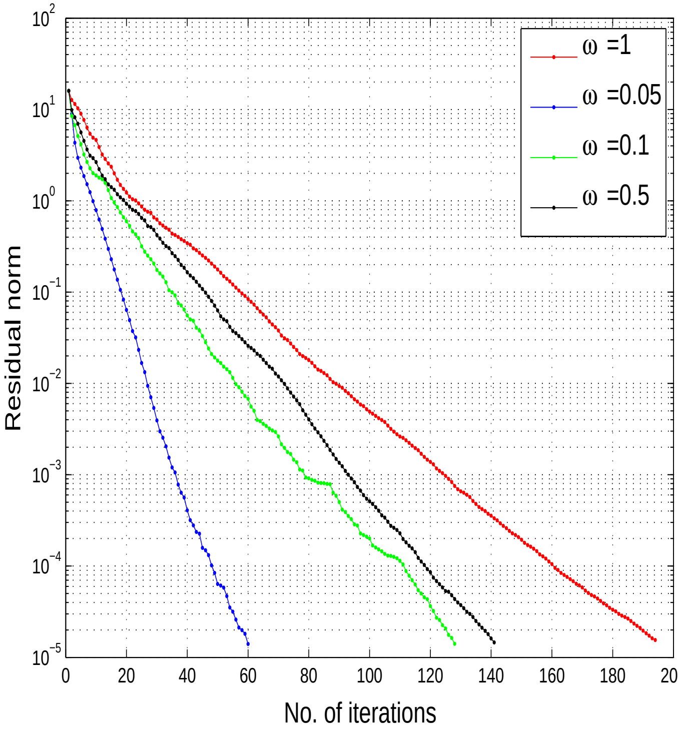




Numerical Experiments (SIMPLE type preconditioners)

Effect of relaxation parameter: The Stokes problem solved in Q2-Q1 discretized driven cavity problem with varying ω: 32×32 grid (Left), 64×64 grid (Right).





Numerical Experiments (comparison)

3D Backward facing step: Preconditioners used in the Stokes problem with preconditioned GCR(20) with $accuracy$ of 10^{-6} (SEPRAN) using Q2-Q1 hexahedrons

Grid	SIMPLE	LSC	MSIMPLER
iter. $(t_s)\frac{\text{in-it-}u}{\text{in-it-}p}$			
$8 \times 8 \times 16$	44(4) $\frac{97}{342}$	16(1.9) $\frac{41}{216}$	14(1.4) $\frac{28}{168}$
$16 \times 16 \times 32$	84(107) $\frac{315}{1982}$	29(51) $\frac{161}{1263}$	17(21) $\frac{52}{766}$
$24 \times 24 \times 48$	99(447) $\frac{339}{3392}$	26(233) $\frac{193}{2297}$	17(77) $\frac{46}{1116}$
$32 \times 32 \times 40$	132(972) $\frac{574}{5559}$	37(379) $\frac{233}{2887}$	20(143) $\frac{66}{1604}$

Numerical Experiments (comparison)

3D Backward facing step: Preconditioners used in solving the Navier-Stokes problem with preconditioned GCR(20) with $accuracy$ of 10^{-2} (SEPRAN) using Q2-Q1 hexahedrons

Re	LSC	MSIMPLER	SILU
	GCR iter. (t_s)	GCR iter. (t_s)	Bi-CGSTAB iter. (t_s)
		$16 \times 16 \times 32$	
100	173(462)	96(162)	321(114)
200	256(565)	145(223)	461(173)
400	399(745)	235(312)	768(267)
		$32 \times 32 \times 40$	
100	240(5490)	130(1637)	1039(1516)
200	397(7040)	193(2251)	1378(2000)
400	675(11000)	295(2800)	1680(2450)

Numerical Experiments (comparison)

3D Lid driven cavity problem (tetrahedrons):The Navier-Stokes problem is solved with accuracy 10^{-4}, a linear system at each Picard step is solved with accuracy 10^{-2} using preconditioned Krylov subspace methods. Bi-CGSTAB is used as inner solver in block preconditioners(SEPRAN)

Re	LSC	MSIMPLER	SILU
	GCR iter. (t_s)	GCR iter. (t_s)	Bi-CGSTAB iter. (t_s)
$16 \times 16 \times 16$			
20	30(20)	20(16)	144(22)
50	57(37)	37(24)	234(35)
100	120(81)	68(44)	427(62)
$32 \times 32 \times 32$			
20	38(234)	29(144)	463(353)
50	87(544)	53(300)	764(585)
100	210(1440)	104(654)	1449(1116)

Numerical Experiments (comparison)

2D Lid driven cavity problem on 64×64 stretched grid: The Stokes problem is solved with accuracy 10^{-6}. PCG is used as inner solver in block preconditioners (SEPRAN) .

Stretch factor	LSC	MSIMPLER	SILU
	GCR iter.	GCR iter.	Bi-CGSTAB iter.
1	20	17	96
8	49	28	189
16	71	34	317
32	97	45	414
64	145	56	NC
128	NC	81	NC

6. Preconditioners for the incompressible Stokes problem

The main motivation behind this work is to develop and test preconditioners for the variable viscosity Stokes problem.

- The preconditioners that are developed for the incompressible Navier-Stokes problem or Saddle point problems are also applied to the Stokes problem.
- Block diagonal preconditioner [Rusten and Winther -1992, Silvester and Wathen-1994]
- Block triangular preconditioner [Bramble and Pasaick-1988, Elman et. al 2005]
- Recent preconditioners [May and Moresi - 2008, Burstedde et. al - 2008, Grinevich and Olshanskii -2008]

Preconditioners for the incompressible Stokes problem

Block-triangular preconditioner

$$P_{PMM} = \mathcal{U}_{bt} = \begin{bmatrix} F & B^T \\ 0 & -1/\nu Q_p \end{bmatrix}$$

Q_p is the pressure-mass matrix.

- h-independent convergence
- Gives approximately two times faster convergence than the block diagonal approach [Elman et. al, 2005]
- One Poisson solve and one velocity solve

Preconditioners for the incompressible Stokes problem

The Schur method

- In this method, instead of solving $\mathcal{A}x = b$, factorized system is solved by solving $\mathcal{A}x = \mathcal{L}_b\mathcal{D}_b\mathcal{U}_b x = b$.
- $M_l = M_u = F$
- Instead of approximating S in $\mathcal{D}_b$, $BF^{-1}B^T$ is treated implicitly by computing $BF^{-1}B^T p$ at each step of Krylov method.
- The pressure mass matrix is used as preconditioner for the Schur-complement system.
- Efficient solver for the velocity subsystem is required.

Preconditioners for the incompressible Stokes problem

The Schur method

```
Initialize $u^{(0)}$, $p^{(0)}$ and $maxiter$ (maximum iterations)
Compute: $r_u = f - Fu^{(0)} - B^T p^{(0)}$
  $r_p = g - Bu^{(0)}$
For $k = 0$ to $maxiter$
    1.  Solve $Fu_f = r_u$
    2.  Solve $-BF^{-1}B^T p_\delta = r_p - Bu_f$
    3.  Update $u_\delta = u_f - u_l$, where $u_l$ is obtained by solving $Fu_l = B^T p_\delta$
    4.  Update $u^{(k+1)} = u^{(k)} + u_\delta$
    5.  Update $p^{(k+1)} = p^{(k)} + p_\delta$
    6.  Update $r_u = f - Fu^{(k+1)} - B^T p^{(k+1)}$
    7.  Update $r_p = g - Bu^{(k+1)}$
    8.  If converged Exit
End For
```

Scaling

The pressure-mass matrix scaling

The standard pressure-mass matrix is defined independently of the viscosity

$$(Q_p)_{i,j} = \int_\Omega \phi_i \phi_j d\Omega, \tag{1}$$

In case of variable viscosity, we consider two alternatives:

1. Explicit scaling:

   $$Q_{pe} = S_v^{-1} Q_p S_v^{-1}, \text{where} S_v = diag(\sqrt{\nu})$$

2. Implicitly scaling: This is done at the time of formation of the pressure-mass matrix. In this case, the smaller value of ν will dominate the definition of Q_{pi} (due to its inversion) at the nodes that are shared by more elements.

Scaling

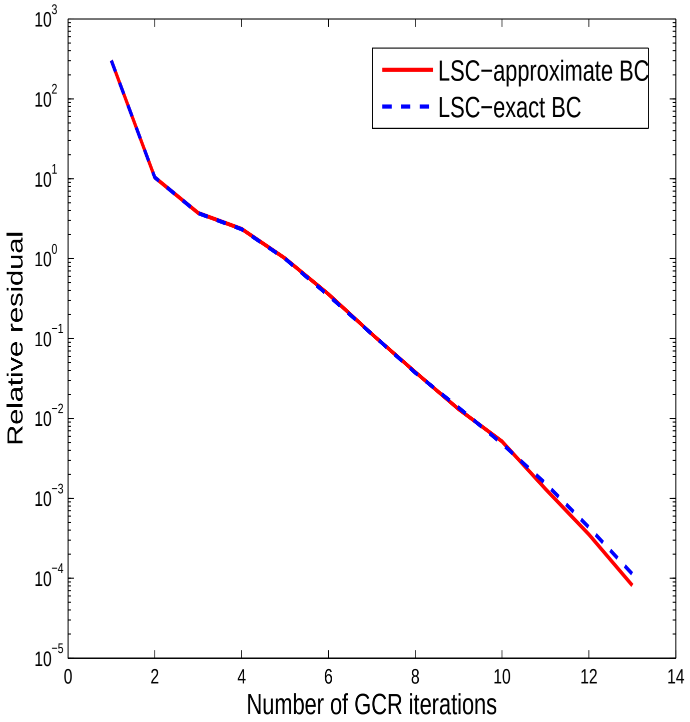
The velocity-mass matrix scaling

- *Use of AMG requires a constant number of unknowns per grid point.*
- *If one velocity component is prescribed, this will lead to only 1(2D) or 2(3D) degrees of freedom on those points.*

- Approximate BC: $c_n u_n + \sigma^{nt} = c_n \bar{u}$, where c_n a large number, $\bar{u}$ is the prescribed value of the normal velocity and σ^{nt} is the tangential component of the stress tensor.
- To get convergence with LSC and MSIMPLER, we must update the velocity mass matrix by multiplying the entries corresponding to the boundary elements with the approximate boundary conditions by a factor of c_n.

Scaling

The velocity-mass matrix scaling



Scaling

System matrix scaling

- *In high viscosity contrast problem, if we use convergence criteria based on the L_2 norm, some preconditioners e.g. PMM, lead to fewer iterations. However, an inaccurate solution is obtained with this convergence criteria.*
- *If we use a preconditioner for the Schur-complement that involves the diagonal of the velocity matrix D^{-1}, the error in the iterative method using a direct method for the subsystems becomes small. This has been verified for LSC_D, $BD^{-1}B^T$ and SIMPLE.*
- We use S_m as scaling matrix given:

$$S_m = \begin{bmatrix} \sqrt{diag(F)} & 0 \\ 0 & \sqrt{diag(BD^{-1}B^T)} \end{bmatrix}.$$

- We solve $S_m^{-1}\mathcal{A}S_m^{-1}S_m x = S_m^{-1}b$
- Convergence criteria is now based on scaled L_2 norm

Numerical Experiments (The Stokes problem, constant ν)

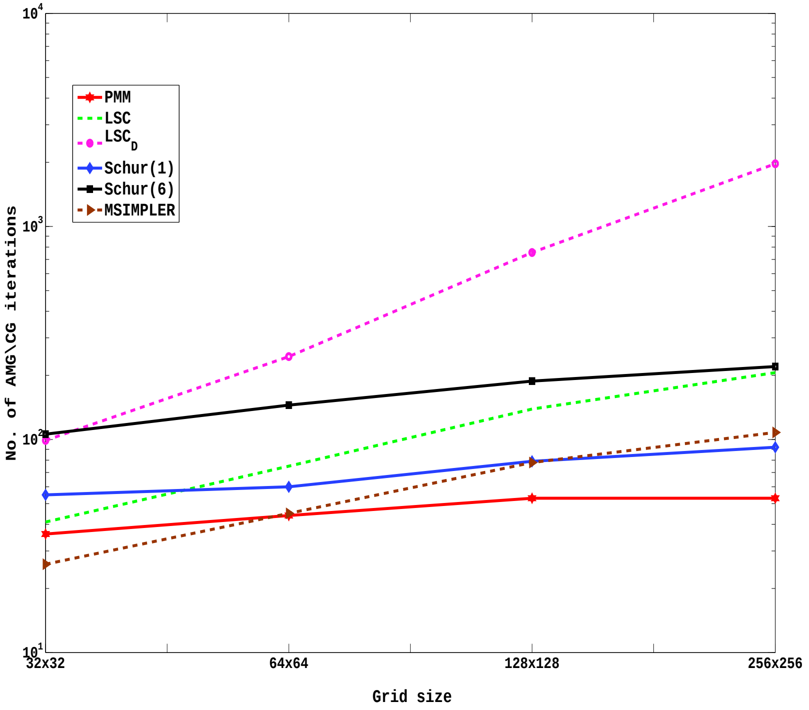
The Stokes driven cavity flow problem with Q2-Q1 discretization with AMG/CG for the velocity subsystem solves and ICCG(0) for the Schur subsystem solves. Solution accuracy is 10^{-6}.

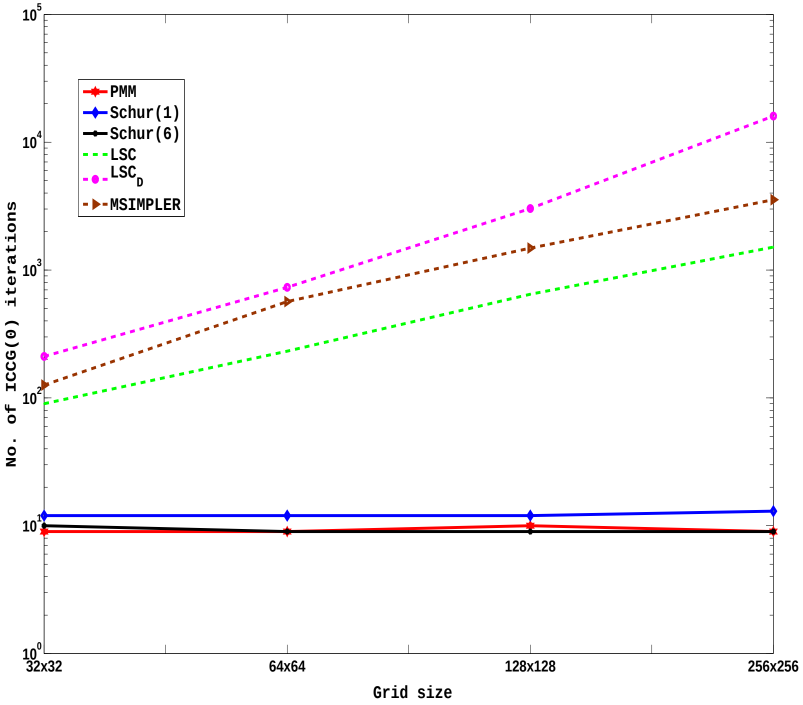
Preconditioner	Grids			
	32×32	64×64	128×128	256×256
	iter.(time in seconds)			
PMM	11(1.4)	10(5.6)	9(23.6)	9(97)
LSC	10(1.38)	13(8.3)	17(54)	22(319)
LSC_D^*	22(3.2)	37(25)	80(275)	180(2880)
MSIMPLER	13(1.5)	16(8)	22(50)	29(300)
Schur(1)	6(3)	5(10.2)	5(46)	6(221)
Schur(6)	1(2)	1(10.6)	1(53)	1(251)

$*$: LSC_D uses diagonal of the velocity system in the Schur complement approximation.

Numerical Experiments (The Stokes problem, constant ν)

Solution of **the constant-viscosity Stokes problem** with accuracy 10^{-6}: Left, the total number of iterations required for the velocity subsystem. Right, the total number of iterations required for the pressure subsystem.



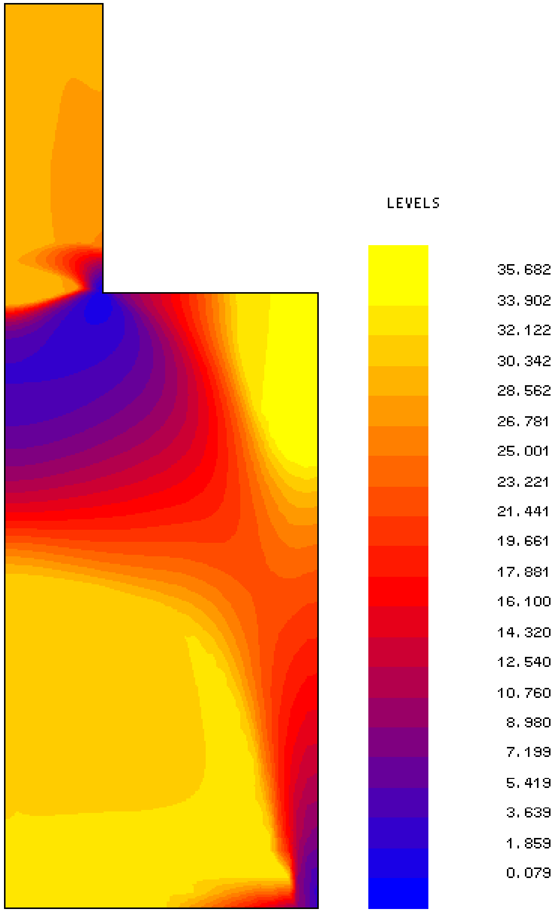


Numerical Experiments (The Stokes problem, variable ν)

Extrusion problem

A round aluminum rod is heated and pressed through a die.

The viscosity model used describes the viscosity as function of shear stress and temperature, which are highest at the die where the aluminum is forced to flow into a much smaller region.



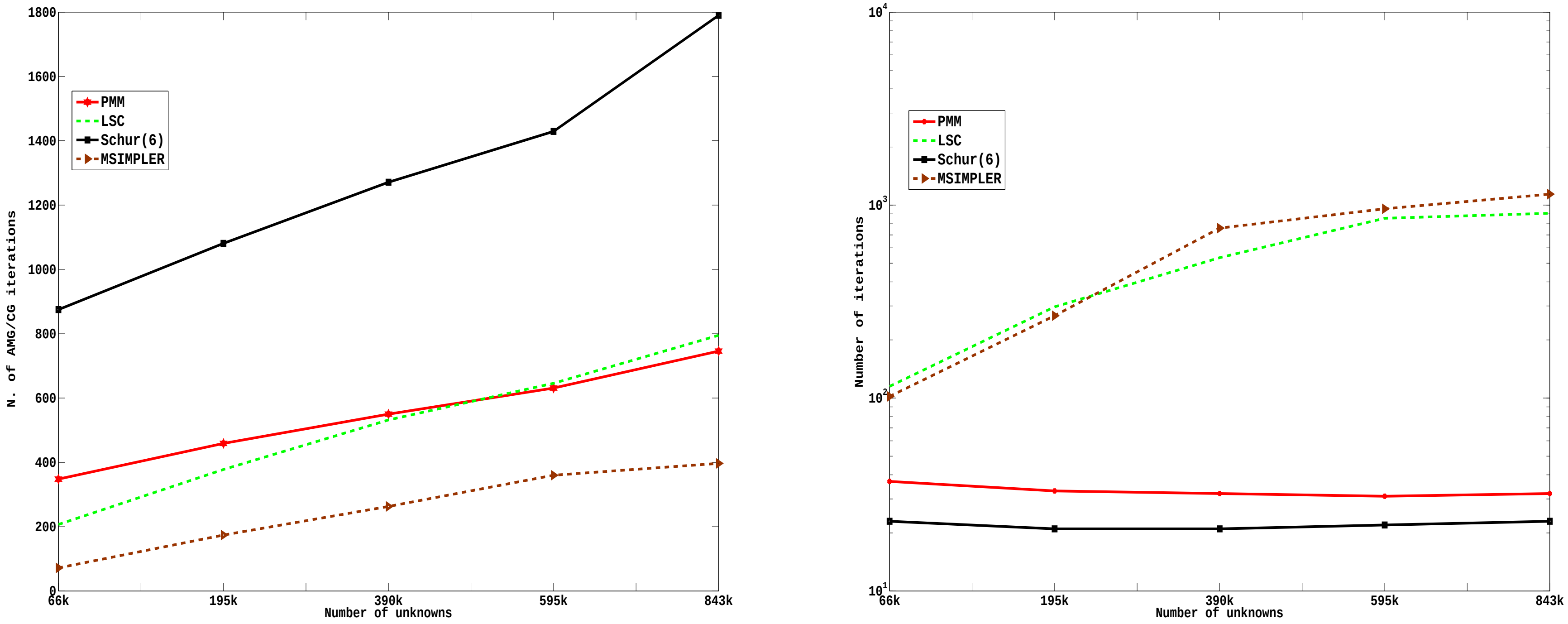
Numerical Experiments (The Stokes problem, variable ν)

The variable-viscosity Stokes problem with Q2-Q1 discretization with AMG/CG for the velocity subsystem and ICCG(0) (PMM, Schur method) or AMG/CG (LSC, LSC_D, MSIMPLER) for the Schur subsystem. Solution accuracy is 10^{-6}.

Grid ↓	Levels/N	PMM	LSC	LSC_D	MSIMPLER	Schur
tol $\longrightarrow$		10^{-3}	10^{-3}	10^{-6}	10^{-1}, 10^{-3}	10^{-6}
	iter.(time in seconds)					
$66k$	3/394	19(51)	11(35)	74(357)	15(35)	1(104)
$195k$	4/152	18(183)	13(188)	129(2650)	19(138)	1(370)
$390k$	5/300	18(429)	14(480)	> 1000	19(360)	1(869)
$595k$	5/408	19(743)	15(871)	> 1000	19(693)	1(1478)
$843k$	6/112	19(1229)	15(1406)	> 1000	21(989)	1(2686)

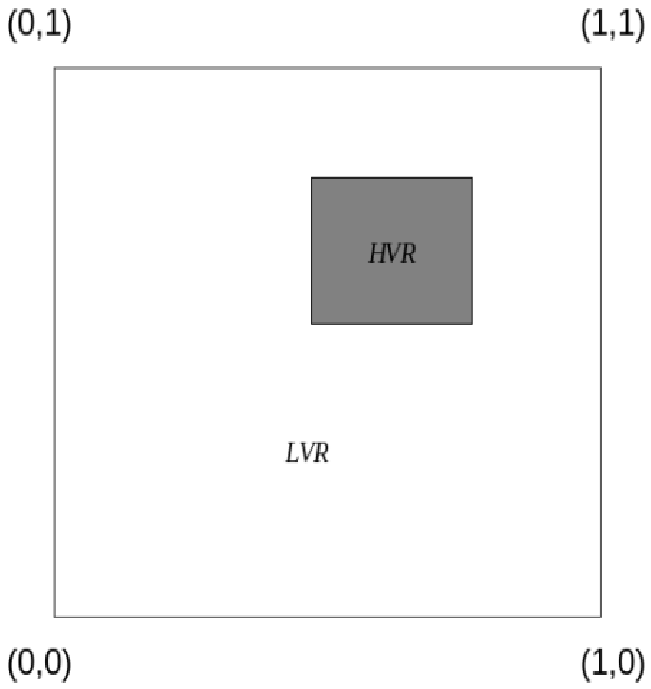
Numerical Experiments (The Stokes problem, variable ν)

Solution of the variable viscosity Stokes problem with accuracy 10^{-6}: Left, the total number of iterations required for the velocity subsystem. Right, the total number of iterations required for the pressure subsystem.



Numerical Experiments (The Stokes problem, jumping ν)

Two dimensional domain for the variable-viscosity Stokes problem (Left). At right, a 2D geodynamics test model: LVR represents the low viscosity region with density $\rho_1 = 1$ and viscosity $\nu_1 = 1$, and HVR denotes the high viscosity region with density $\rho_2 = 2$, and constant viscosity ν_2 ($1,\ 10^3$ and 10^6).



Numerical Experiments (The Stokes problem, jumping ν)

Iterative solution of the Stokes problem with SINKER configuration , accuracy = 10^{-6}.
$Error = \|p_{exact} - p_{PMM,\ LSC_D,\ Schur}\|_2$

ν	PMM		LSC_D		Schur	
	iter.	$Error$	iter.	$Error$	iter. (inner)	$Error$
30×30						
$\nu_2 = 10^6$	12	9×10^{-4}	26	7×10^{-6}	2(18)	2×10^{-8}
$\nu_2 = 10^3$	12	2×10^{-5}	26	3×10^{-6}	2(20)	2×10^{-10}
$\nu_2 = 10^1$	11	5×10^{-6}	24	1×10^{-6}	2(16)	2×10^{-10}
60×60						
$\nu_2 = 10^6$	13	8×10^{-3}	40	6×10^{-5}	2(19)	5×10^{-8}
$\nu_2 = 10^3$	13	3×10^{-5}	40	5×10^{-6}	2(20)	3×10^{-9}
$\nu_2 = 10^1$	13	1×10^{-6}	41	3×10^{-6}	2(18)	4×10^{-10}

Numerical Experiments (The Stokes problem, jumping ν)

Remark: The Schur method gives more accurate results might be the fact that it uses subfactors of the system iteratively in a classical way. If that is the case, we may also expect better results when using Richardson type iterative improvement of the form:

$$x_{k+1} = x_k + PMM^{-1}(b - \mathcal{A}x_k).$$

We observe that Richardson also gives better accuracy than PMM and LSC.

Conclusions

- *In ILU, A new scheme for the renumbering of grid points and reordering of unknowns is introduced that prevents the break down of the resulting SILU preconditioner and leads to faster convergence of Krylov subspace methods.*
- *MSIMPLER is at present the fastest of all SIMPLE-type preconditioners.*
- *In our experiments, MSIMPLER proved to be cheaper than SILU, especially when the problem is solved with high accuracy.*
- PMM shows h-independent convergence for all types of viscosity configurations. For the problems that require subsystem to be accurately solved, PMM and the Schur method give similar convergence.
- MSIMPLER shows better performance than LSC. Both have similar convergence characteristics.
- In a high viscosity contrast problem, the Schur complement matrix must be approximated by an operator that contains viscosity information for the problem.
- In large viscosity contrast problem,to get results efficiently, the Schur method is better option to use than the other preconditioners.

Conclusions

Further research

- To reduce the effect of stretching on convergence of the preconditioned Krylov subspsce methods.
- Use of deflation-type schemes in the variable discontinuous viscosity Stokes problem.

References

- Website: http://ta.twi.tudelft.nl/users/vuik/
- C. Vuik and A. Saghir and G.P. Boerstoel, "The Krylov accelerated SIMPLE(R) method for flow problems in industrial furnaces," *International Journal for Numerical methods in fluids*, 33 pp. 1027-1040, 2000.
- M. ur Rehman and C. Vuik and G. Segal, "A comparison of preconditioners for incompressible Navier-Stokes solvers," *International Journal for Numerical methods in fluids*, 57, pp. 1731-1751, 2008.
- M. ur Rehman and C. Vuik and G. Segal, "SIMPLE-type preconditioners for the Oseen problem," *International Journal for Numerical methods in fluids*, to appear.
- M. ur Rehman and T. Geenen and C. Vuik and G. Segal and S. P. MacLachlan "On iterative methods for the incompressible Stokes problem," *International Journal for Numerical methods in fluids*, Submitted.
- T. Geenen, M. ur Rehman, S.P. MacLachlan, G. Segal, C. Vuik, A. P. van den Berg, and W. Spakman. "Scalabale robust solvers for unstructured FE modeling applications; solving the Stokes equation for models with large, localized, viscosity contrasts,". *Geochemistry, Geophysics, Geosystems*, accepted.