A comparison of preconditioners for industrial incompressible flows

ECCOMAS 2008, Venice, Italy

30 June - 4 July 2008 .

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

Numerical analysis group, DIAM

Delft University of Technology

Outline

- Introduction
- Preconditioning
- Numerical experiments
- Conclusions

The incompressible Navier Stokes equation

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

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

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

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

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 $m \leq n$

- $F = A$ 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(Stokes problem), nonsymmetric otherwise.
Saddle point problem having large number of zeros on the main diagonal

Preconditioners

ILU preconditioner

$\mathcal{A} = LD^{-1}U + R$,

$(LD^{-1}U)_{i,j} = a_{i,j}$ 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

Efficient ILU preconditioner

- Pivoting
- A priori reordering/renumbering

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]

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

Matthias Bollhöfer, Yousef Saad. Multilevel Preconditioners Constructed From

Inverse-Based ILUs, *SIAM Journal on Scientific Computing, 27* , 5(2005), 1627-1650

New reordering scheme

- Renumbering of grid points: Grid points are renumbered with Sloan or Cuthill McKee algorithms
- Reordering of unknowns
  - **p-last per node reordering**, The velocity unknowns are ordered followed by pressure unknowns per node (Optimal profile but breakdown of ILU may occur)
  - **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.
  - **p-last per level reordering**: The grid is divided into levels. Each level consists of a connected set of nodes. Thereafter, the unknowns are ordered per level. At each level, first velocity unknowns are placed and then followed by the pressure unknowns.

Block preconditioners

Block triangular preconditioner

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

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

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

Block preconditioners

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 (sometime) and mildly dependent on Reynolds number
  - Require iterative solvers (Multigrid) for the (1,1) and (2,2) blocks
  - Require extra operators

Block preconditioners

SIMPLE preconditioner [Vuik 2000]

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

with approximation $F^{-1} = D^{-1} = diag(F)^{-1}$ in (2,2) and (1,2) in L and U block matrices.

Algorithm form:

1. Solve $Fu^* = r_u - B^T p^*$.
2. Solve $\hat{S}\delta p = r_p - Bu^*$.
3. update $u = u^* - D^{-1}B^T\delta p$.
4. update $p = p^* + \delta p$.

SIMPLE type preconditioner

$$\begin{pmatrix} u^* \\ p^* \end{pmatrix} = \begin{pmatrix} u^k \\ p^k \end{pmatrix} + M_L^{-1} B_L \left(\begin{pmatrix} r_u \\ r_p \end{pmatrix} - A \begin{pmatrix} u^k \\ p^k \end{pmatrix} \right),$$

$$\begin{pmatrix} u^{k+1} \\ p^{k+1} \end{pmatrix} = \begin{pmatrix} u^* \\ p^* \end{pmatrix} + B_R M_R^{-1} \left(\begin{pmatrix} r_u \\ r_p \end{pmatrix} - A \begin{pmatrix} u^* \\ p^* \end{pmatrix} \right).$$

Where

$$B_R = \begin{pmatrix} I & -D^{-1}B^T \\ 0 & I \end{pmatrix}, \; M_R = \begin{pmatrix} F & 0 \\ B & \hat{S} \end{pmatrix} \text{ and}$$

$$B_L = \begin{pmatrix} I & 0 \\ -BD^{-1} & I \end{pmatrix}, \; M_L = \begin{pmatrix} F & B^T \\ 0 & \hat{S} \end{pmatrix}.$$

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

SIMPLE type preconditioner

With Lemma and assuming u^k and p^k equal zero, the steps in SIMPLER reduce to

SIMPLER preconditioner:

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

- Two Poisson solve
- One velocity solve
- Gives faster convergence than SIMPLE

SIMPLE type preconditioner

MSIMPLER preconditioner

In SIMPLER, by making following changes leads to 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 $\hat{Q}_u^{-1}$.
Properties:

- Convergence better than other variants of SIMPLE
- Cheaper than SIMPLER (in construction) and LSC (per iteration)

Numerical Experiments

- Driven Cavity flow (2D)
- Backward facing flow (2D and 3D)
- Q2-Q1 finite element discretization [Taylor, Hood - 1973]
- Q2-P1 finite element discretization [Crouzeix, Raviart - 1973]
- GCR(20), Bi-CGSTAB, GMRES
- The iteration is stopped if the linear systems satisfy $\frac{\|r^k\|_2}{\|b\|_2} \leq tol$,

Numerical experiments(ILU preconditioners)

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

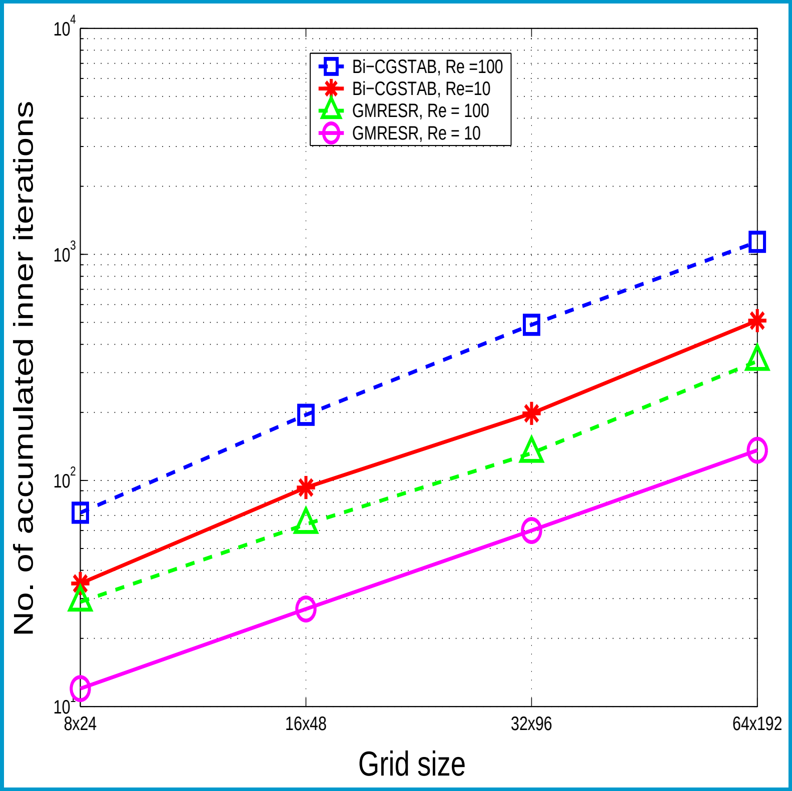
Numerical experiments (SILU preconditioner)

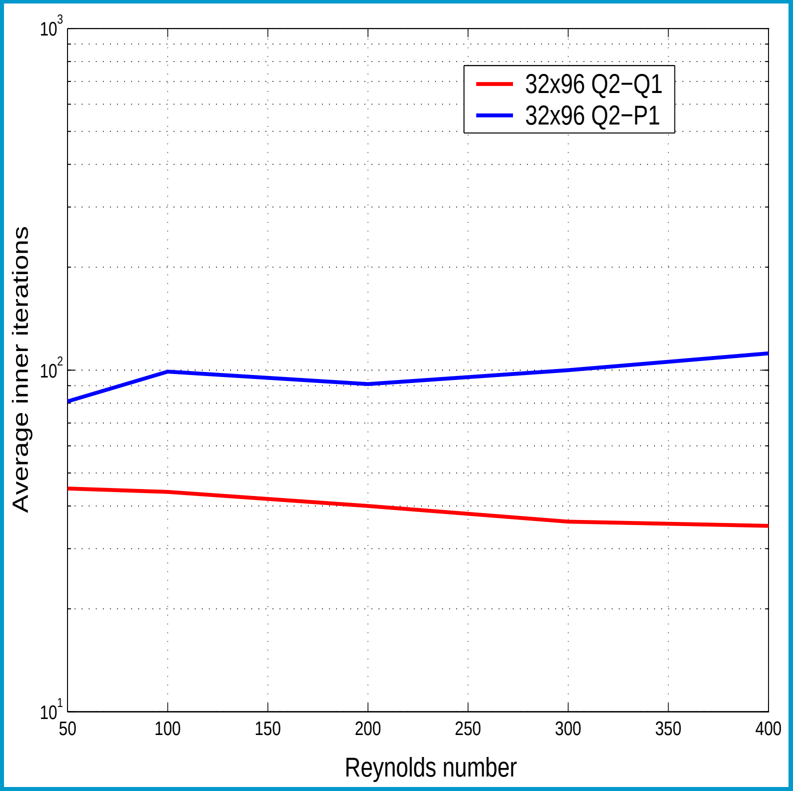
Stokes Problem in a square domain with Bi-CGSTAB ,
$accuracy = 10^{-6}$**, Sloan renumbering**

	$Q2-Q1$		$Q2-P1$	
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 preconditioner)

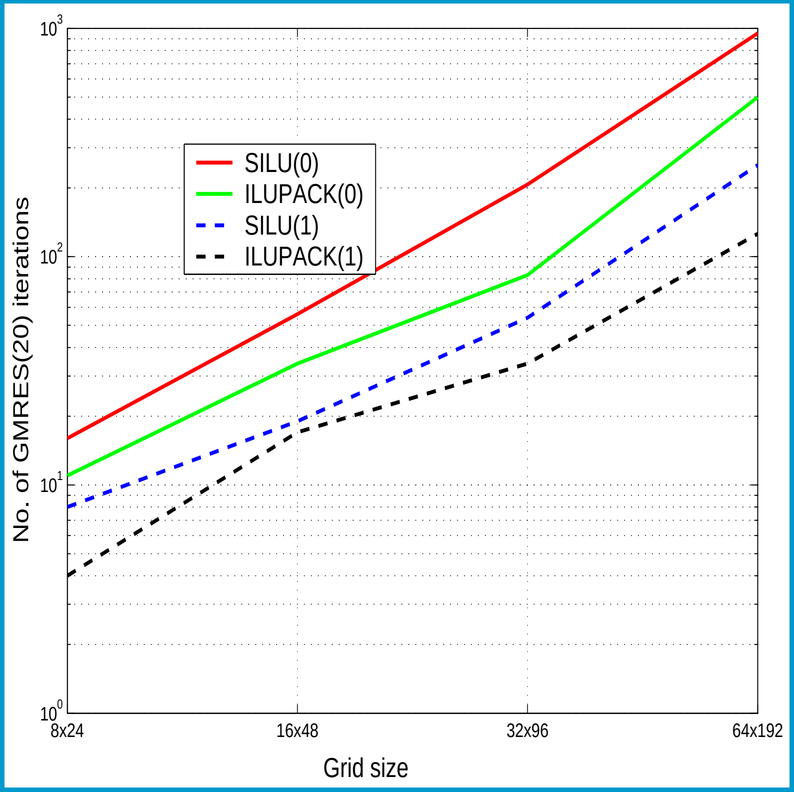
Effect of grid increase(Left) and Reynolds number(Right) on inner iterations for the Navier Stokes backward facing step problem with $accuracy = 10^{-2}$ using the p-last-level reordering

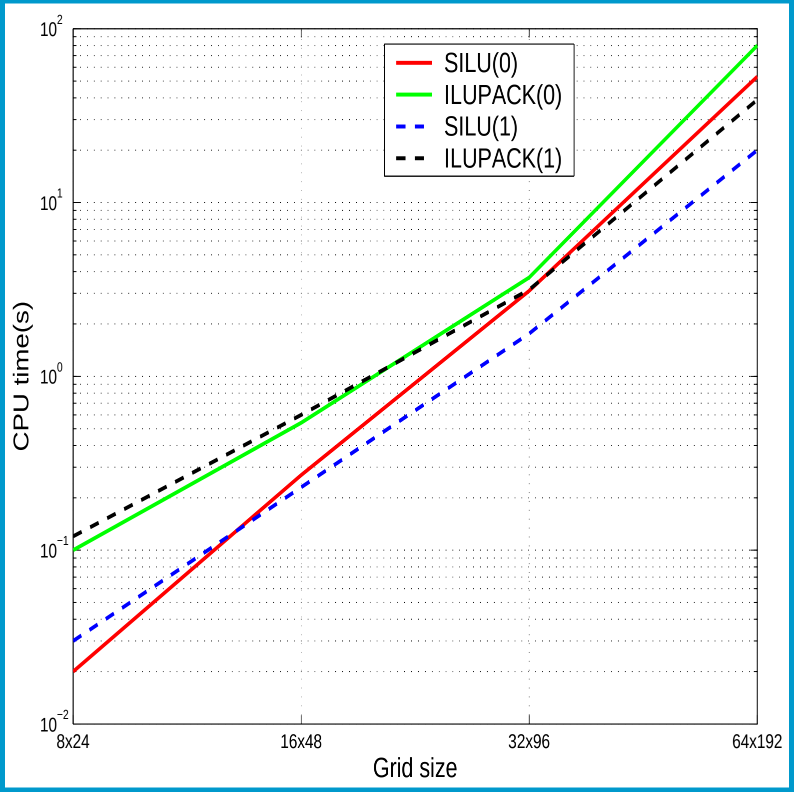




Numerical experiments (ILU preconditioners)

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





64x192 grid (4 iterations, const. time(s) = 155, solver time(s)=5, gain factor =9)

TU Delft

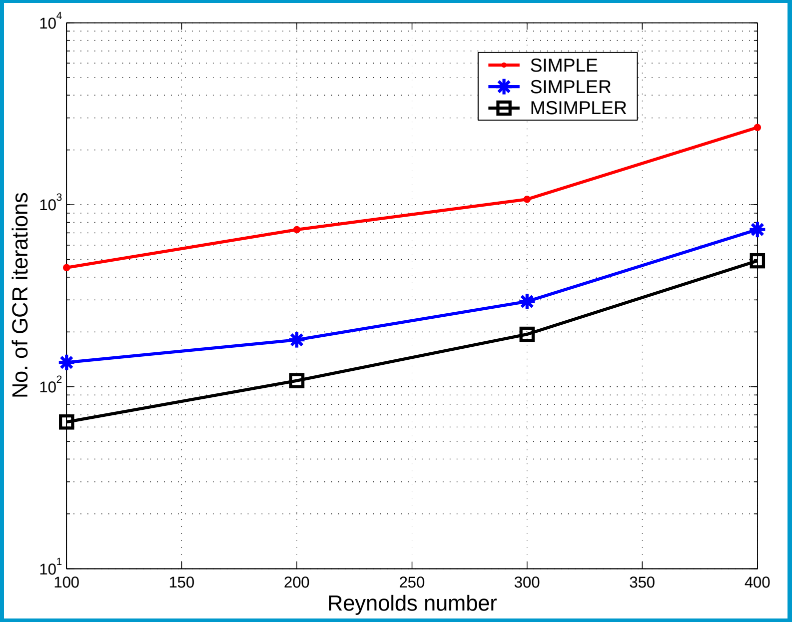
Numerical Experiments (SIMPLE type preconditioners)

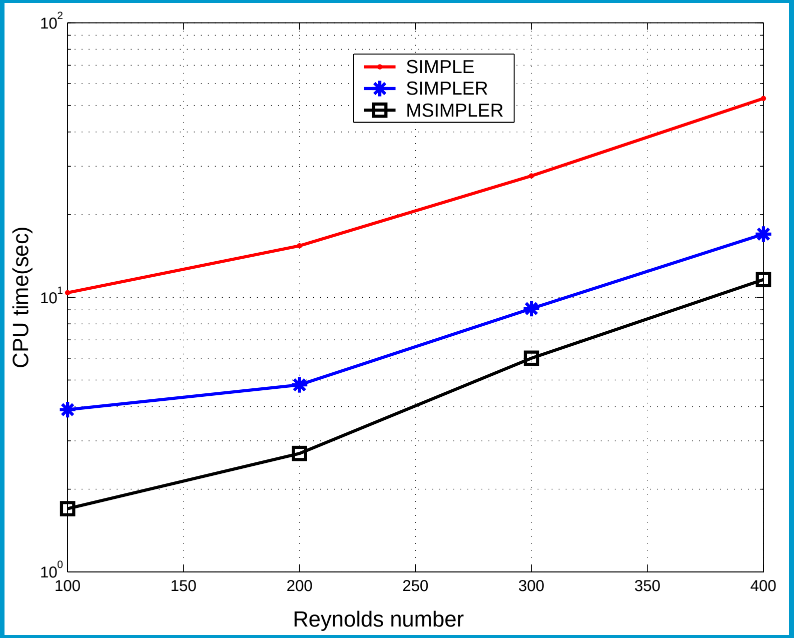
Stokes backward facing step, GCR(20) with $accuracy$ of 10^{-4}, PCG used as an inner solver (SEPRAN)

	SIMPLE		SIMPLER		MSIMPLER	
Grid size	out-it, t(s)	$\frac{\text{in-it-}u}{\text{in-it-}p}$	out-it, t(s)	$\frac{\text{in-it-}u}{\text{in-it-}p}$	out-it, t(s)	$\frac{\text{in-it-}u}{\text{in-it-}p}$
16×48	49, 0.8	$\frac{145}{765}$	28, 0.58	$\frac{224}{849}$	9, 0.15	$\frac{22}{260}$
32×96	89, 8.9	$\frac{418}{2585}$	66, 20	$\frac{2680}{2654}$	10, 0.97	$\frac{32}{568}$
64×192	193, 148	$\frac{1940}{10067}$	NC	-	14, 8.4	$\frac{90}{1433}$

Numerical Experiments (SIMPLE type preconditioners)

The Navier-Stokes problem solved in Q2-Q1 discretized 16×48 backward facing step with varying Reynolds number, Number of accumulated inner iterations(Left), CPU time in seconds (Right)-(SEPRAN)





Numerical Experiments (overall 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	SILU (Bi-CGSTAB)
iter. $(t_s)\frac{\text{in-it-}u}{\text{in-it-}p}$				iter. (t_s)
$8 \times 8 \times 16$	44(4) $\frac{97}{342}$	16(1.9) $\frac{41}{216}$	14(1.4) $\frac{28}{168}$	26(0.7)
$16 \times 16 \times 32$	84(107) $\frac{315}{1982}$	29(51) $\frac{161}{1263}$	17(21) $\frac{52}{766}$	65(16.7)
$24 \times 24 \times 48$	99(447) $\frac{339}{3392}$	26(233) $\frac{193}{2297}$	17(77) $\frac{46}{1116}$	117(118)
$32 \times 32 \times 40$	132(972) $\frac{574}{5559}$	37(379) $\frac{233}{2887}$	20(143) $\frac{66}{1604}$	189(235)

Numerical Experiments (overall 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	NC	193(2251)	1378(2000)
400	675(11000)	295(2800)	1680(2450)

Numerical Experiments (overall 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)

Conclusions

- Two new preconditioners from ILU family and block type presented that competes with the preconditioners published recently.
- In ILU, 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.
- MSIMPLER is at present the fastest of all SIMPLE-type preconditioners.
- In contrast with SIMPLER, MSIMPLER is not sensitive to the accuracies that are used for the inner solvers.
- In all our experiments MSIMPLER proved to be cheaper than LSC. This concerns both the number of outer iterations, inner iterations and CPU time.
- The number of outer iterations in MSIMPLER hardly increases if a direct solver for the subsystems is replaced by an iterative solver. This is in contrast with LSC where large differences are observed.
- In our experiments, MSIMPLER proved to be cheaper than SILU, especially when the problem is solved with high accuracy.

Thank you for your attention !