Algebraic Multilevel ILU-Based Preconditioning for Stationary Flow Problems

Jan Mayer

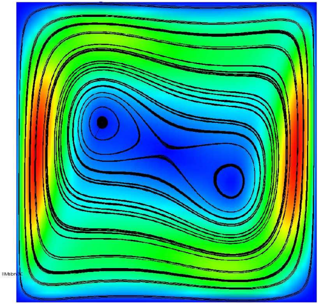
Hendryk Bockelmann, Vincent Heuveline

Universität Karlsruhe

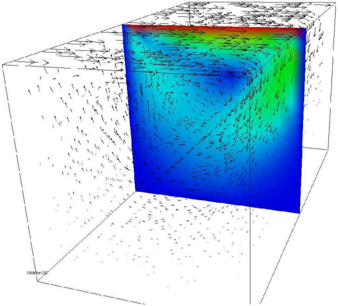
GAMM Linear Algebra Workshop, Sept. 2008

Outline

Two CFD Benchmark Problems

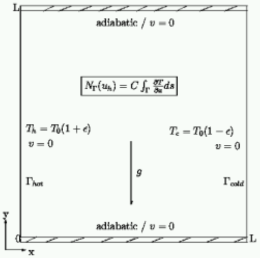


Streamlines
Heat Driven Cavity,
Rayleigh-Number 100 000



Flow Field
Lid Driven Cavity in 3D,
Reynolds-Number 100

Heat Driven Cavity: A Highly Challenging Problem



T_0	=	600	K
P_0	=	101325	Pa
$T^\star$	=	273	K
$\mu^\star$	=	$1.68 \cdot 10^{-5}$	$kg.m^{-1}.s^{-1}$
S	=	110.5	K
$\mathcal{P}r$	=	0.71	
c_p	=	1004.5	$J.K^{-1}$
R	=	287	$J.kg^{-1}.K^{-1}$
L	=	1	m

- Pseudo time-step approach: **highly time consuming** (several days/weeks computational time assuming HPC)
- Stationary approach: **stiff problem** (less than one hour computational time on a workstation)

[Paillère et al. 2005] more then 30 institutes involved.

Heat Driven Cavity: Low Mach Model

Asymptotic model (pressure splitting)

$$\begin{aligned} \partial_t \rho + \nabla \cdot (\rho v) &= 0\,, \\ \partial_t(\rho v) + [\nabla \cdot \rho v v] + \nabla p + \nabla \cdot \tau &= \rho g\,, \\ c_p \rho D_t T - D_t p - \nabla \cdot (\kappa \nabla T) + (\tau : \nabla v) &= 0\,. \end{aligned}$$

$$\begin{aligned} p(x,t) &= p_{th}(t) + p_{hyd}(x,t) \\ \rho &= \frac{p_{th}}{RT} \end{aligned}$$

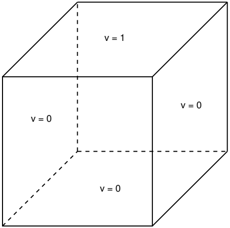
$$\begin{aligned} \tau &:= \mu(\nabla v + \nabla v^T) + \frac{2}{3}\mu(\nabla \cdot v)I \\ P_{th} &= m_0 \left(\int_\Omega \frac{1}{RT} dx \right)^{-1} \end{aligned}$$

where τ, c_p, κ, g, and μ represent the *shear-stresstensor*, the *heat capacity*, the *heat conductivity*, the *volume forces* and the *shear viscosity*, respectively.

Lid Driven Cavity: Navier-Stokes Equations

Model: incompressible Navier-Stokes equations

$$\begin{aligned} -\nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \frac{1}{\rho}\nabla p &= 0 \text{ in } \Omega \\ \nabla\cdot\mathbf{u} &= 0 \text{ in } \Omega \\ \mathbf{u} &= \mathbf{b} \text{ on } \partial\Omega \end{aligned}$$



Here, we chose $\Omega = [0,1]^d$,

$\mathbf{u}_{\text{top}} = 1$ and $\mathbf{u}|_{\partial\Omega} = 0$ otherwise.

$\nu = 0.01$, $\text{Re} = \frac{\rho \mathbf{u}_{\text{top}} L}{\nu} = 100$ guarantees a stationary solution.

Variational Formulation: Stokes Model

$$
\begin{aligned}
-\Delta u + \nabla p &= f \quad \text{in} \;\; \Omega, \\
\nabla \cdot u &= 0 \quad \text{in} \;\; \Omega, \qquad (1)\\
u &= 0 \quad \text{on} \;\; \partial\Omega.
\end{aligned}
$$

Find $(u, p) \in (H_0^1(\Omega))^d \times L_0^2(\Omega)$ such that

$$
\begin{array}{lclcll}
a(\mathbf{u}, \varphi) & - & b(p, \varphi) & = & (\mathbf{f}, \varphi) & \forall \varphi \in \mathbf{H}_0^1(\Omega), \\
b(q, \mathbf{u}) & & & = & 0 & \forall q \in L_0^2(\Omega),
\end{array}
$$

where

$$
a(\mathbf{u}, \varphi) = (\nabla \mathbf{u}, \nabla \varphi), \quad b(q, \varphi) = (q, \nabla \cdot \varphi).
$$

FEM Discretization: Conformal Stable Elements

FEM: Find $(\mathbf{u}_N, p_N) \in \mathbf{V}_N \times M_N$ such that

$$\begin{array}{rcrcll} a(\mathbf{u}_N, \varphi_N) & - & b(p_N, \varphi_N) & = & (\mathbf{f}, \varphi_N) & \forall \varphi_N \in \mathbf{V}_N, \\ b(q_N, \mathbf{u}_N) & & & = & 0 & \forall q_N \in M_N. \end{array}$$

(Discrete) Inf-Sup condition (LBB-condition):

$$\inf_{q_N \in M_N \cap L^2_0(\Omega)} \sup_{\mathbf{v}_N \in V_N \cap H^1_0(\Omega)} \frac{(\nabla \cdot \mathbf{v}_N, q_N)}{|\mathbf{u}_N|_{1,\Omega} ||q_N||_{0,\Omega}} \geq \beta_N > 0,$$

where $\beta_N = \beta_N(\mathbf{V}_N, M_N)$.

Taylor-Hood Element (Q_2, Q_1)
no pressure stabilization needed!

Discretization: Algebraic Formulation

1. Linearization of Navier-Stokes equations by a (damped) Newton iteration yields linear saddle point problem.
2. Discretization by Galerkin FE method yields a linear system with coefficient matrix

$$A = \begin{pmatrix} N & B \\ B^T & 0 \end{pmatrix}$$

Overview of the Solution Process

Algorithm

choose initial $x = (u, p)$ and tolerance ε, set $k = 0$
compute residual r of nonlinear system

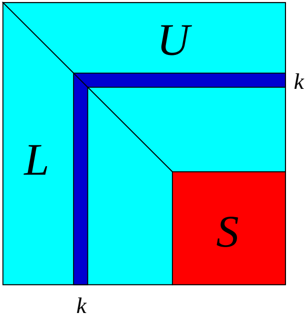
while ($\|r\| > \varepsilon$ **and** $k <$ MaxIter)
compute Jacobian (linearization) A
solve $A\Delta x = r$ with preconditioned GMRES
update solution $x = x + \omega_k \Delta x$
compute residual r of non-linear system
set $k = k + 1$
end while

Preconditioning the Linear System

The Algebraic Multilevel Incomplete Factorization: Practical View

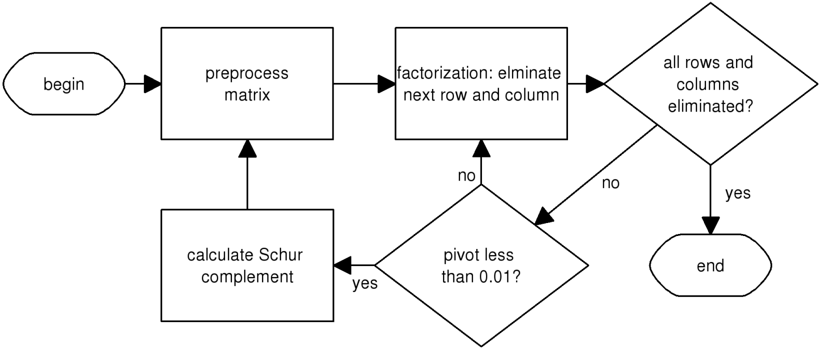
Algorithm

1. Preprocess matrix (permute/scale rows and columns to make more suitable for incomplete LU-factorization).
2. Factor until absolute value of pivot less than 0.01.
3. Compute approximate Schur complement.
4. Goto 1, proceed recursively.



Preconditioning the Linear System

The Algebraic Multilevel Incomplete Factorization: Summary



Preprocessing the Linear System

Options

Preprocessing: permute/scale row and columns to make A more suitable for incomplete factorization, e.g.

1. Normalize columns and then rows of A
   $\longrightarrow$ lid driven cavity
2. I-matrix, i.e. such that $|a_{ij}| \leq 1$ and $|a_{ii}| = 1$ for all i, j
3. I-matrix + symmetric permutation (preserves I-matrices)
   $\longrightarrow$ heat driven cavity

Symmetric Permutations for I-Matrices

Let A be an I-matrix of dimension n. One option is:

Algorithm

1. Assign a weight

$$w_k = \mathrm{nz}(A_{:,k}) \cdot ||A_{:,k}||_1 + \mathrm{nz}(A_{k,:}) \cdot ||A_{k,:}||_1$$

   for each index $k = 1, \ldots, n$.
2. Reorder rows and columns of A by increasing weights.

Note: low weights indicate sparse rows and columns having good diagonal dominance

HiFlow

HiFlow is an FE package featuring

- CFD (incompressible Navier-Stokes, Low-Mach flows, heat convection)
- reactive flows
- eigenvalue lab for stability analysis
- conformal hp-FEM
- a posteriori error estimation
- moving boundaries

More information under www.hiflow.de

ILU++

ILU++ is a software package for solving large sparse linear systems. It provides several:

- preprocessing techniques
- multilevel incomplete LU-factorisations
- dropping techniques
- iterative solvers

ILU++ is fully templated, object-oriented self-contained C++ code.

ILU++ in HiFlow

```
#include <vector>
#include "iluplusplus_interface.h"

void GMRES(
    const iluplusplus::multilevel_preconditioner& Pr,
    ...)
{
     std::vector<double> v;
                 ⋮
     Pr.apply_preconditioner(v);
                 ⋮
}
```

ILU++ in HiFlow

```
int main()
{
     std::vector<int> ia,ja;
     std::vector<double> val;

     /* setup CSR matrix using ia ja val */

     iluplusplus::iluplusplus_precond_parameter p;
     iluplusplus::multilevel_preconditioner Pr;

     Pr.setup(val, ja, ia, iluplusplus::ROW, p);
     GMRES(Pr,...);
}
```

ILU++ in HiFlow

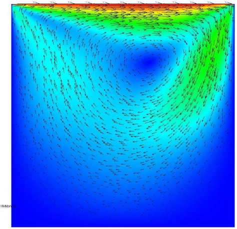
```
int main()
{
     std::vector<int> ia,ja;
     std::vector<double> val;

     /* setup CSR matrix using ia ja val */

     iluplusplus::iluplusplus_precond_parameter p;
     iluplusplus::multilevel_preconditioner Pr;

     Pr.setup(val, ja, ia, iluplusplus::ROW, p);
     GMRES(Pr,...);
}
```

Also implemented the ILU++ linear solver in HiFlow.

Numerical Results

Lid Driven Cavity

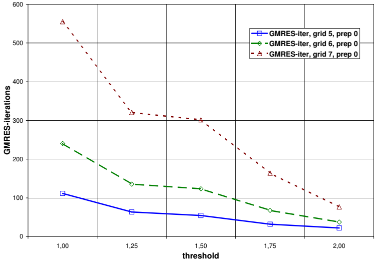
	grid	dofs	nnz Jac.	storage Jac.
2D	5	9 539	377 289	4.43 MB
2D	6	37 507	1 502 089	17.6 MB
2D	7	148 739	5 994 249	70.3 MB
3D	3	29 988	5 752 816	66.2 MB
3D	4	221 892	45 061 456	518 MB

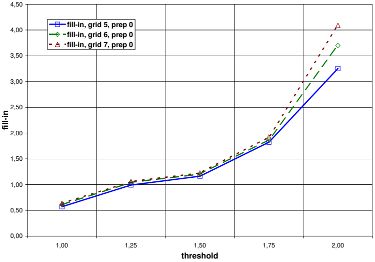


- each problem required 4 or 5 Newton steps
- best preprocessing is just normalization

Numerical Results

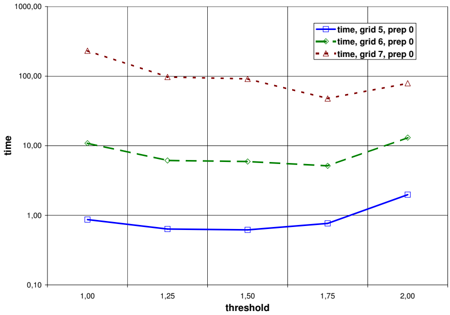
Results for 2D Lid Driven Cavity





Numerical Results

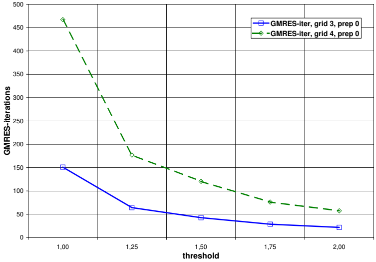
Results for 2D Lid Driven Cavity

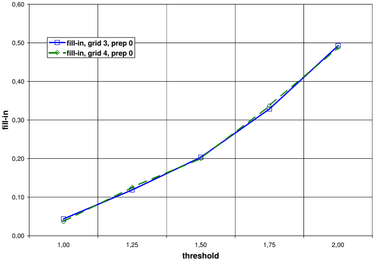


- fill-in can be quite low
- good choice for dropping threshold is easy
- PQ preprocessing also good, others too expensive

Numerical Results

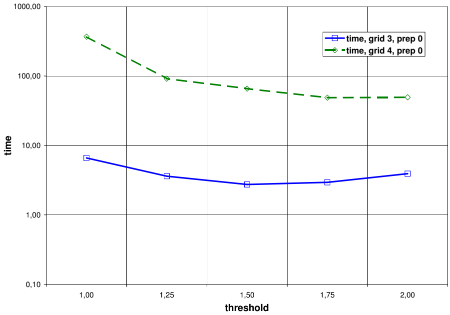
Results for 3D Lid Driven Cavity





Numerical Results

Results for 3D Lid Driven Cavity

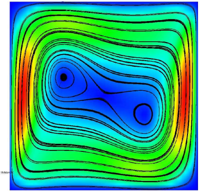


- fill-in is almost negligible
- good choice for dropping threshold is easy
- PQ preprocessing also good, others too expensive

Numerical Results

2D Low Mach Problems (Heat Driven Cavity)

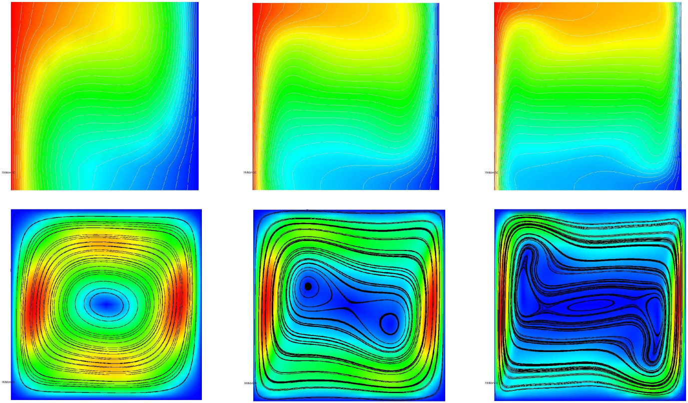
grid	dofs	nnz Jac.	storage Jac.
4	10 436	571 216	6.66 MB
5	40 836	2 271 376	26.5 MB
6	161 540	9 058 576	106 MB
7	642 564	36 180 496	421 MB



- Temperature difference: $T_\varepsilon = 0.01$
- Rayleigh numbers $\mathrm{Ra} = 10^4, 10^5, 10^6$
- Solution from lower level used as initial guess
  $\longrightarrow$ 1-2 Newton steps
- I-matrix with symmetric permutation worked well

Numerical Results

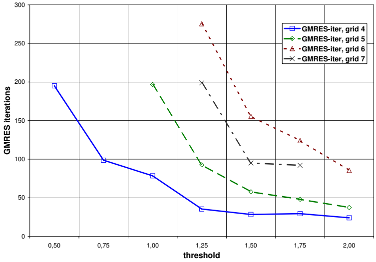
Isolines and Streamlines for 2D Low Mach Problems (Heat Driven Cavity)

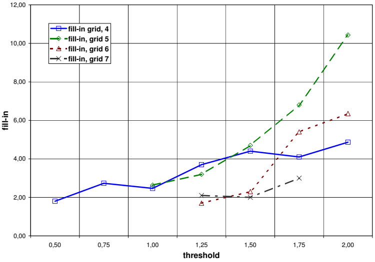


$Ra = 10^4$ $Ra = 10^5$ $Ra = 10^6$

Numerical Results

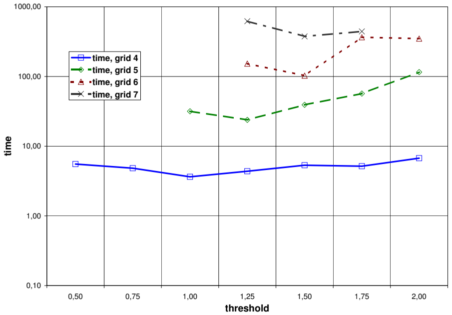
Results for 2D Low Mach Problems: Ra $= 10^6$





Numerical Results

Results for 2D Low Mach Problems: Ra $= 10^6$



- fill-in is reasonable
- good choice for dropping threshold is easy

Conclusion and Outlook

Conclusion

- Best preprocessing depends only on problem type.
- Good value for threshold is predictable.
- Even for difficult flow problems, the proposed preconditioner can be used as an efficient “black-box” approach.

Conclusion and Outlook

Conclusion

- Best preprocessing depends only on problem type.
- Good value for threshold is predictable.
- Even for difficult flow problems, the proposed preconditioner can be used as an efficient “black-box” approach.

Outlook

- Better selection of preprocessing and dropping threshold?
- Parallelization? Possibly with domain decomposition?
- 3D low mach problems?

References

Model, FE:
Heuveline: Int. J. Num. Meth. Fluids 41, 1339 – 1356, (2003)

I-Matrices:
Duff, Koster: SIMAX 22, 973 – 996, (2001)

Symm. Permutations:
M: SISC 30, 982 – 996, (2008)

Multilevel ILU Preconditioning for CFD
Bockelmann, Heuveline, M: Algebraic multilevel ILU based preconditioners for stationary flow problems, IWRMM Preprint.

Software:
HiFlow: `www.hiflow.de`
ILU++: `www.iluplusplus.de`

Thank you for your attention!

Questions?