Updating preconditioners for sequences of non-symmetric linear systems

Philipp Birken
Jurjen Duintjer Tebbens, Andreas Meister, Miroslav Tůma

University of Kassel (SFB/TRR 30), Academy of Sciences (Prague)

21. August 2007 / Harrachov 07

Outline

- The Finite Volume Scheme
- Preconditioner Updates
- Numerical Results

Motivation

- We consider problems of compressible flows from computational fluid dynamics (CFD).
- Problems are nonlinear.
- Both unsteady and steady problems are solved via implicit time stepping schemes involving Newtons method.
- Thus, solving a CFD problem amounts to solving sequences of linear, usually unsymmetric, sparse blocksystems.
- Solving these systems takes the majority of computing time.
- Thus, speeding up this part of the solver is worthwhile!

Outline

- The Finite Volume Scheme
- Preconditioner Updates
- Numerical Results

The Euler equations

Hyperbolic system of conservation laws (mass, momentum, energy) modeling inviscid flow:

$$
\begin{aligned}
\partial_t \rho + \nabla_x \cdot \mathbf{m} &= 0, && (1)\\
\partial_t m_i + \sum_{j=1}^{2} \partial_{x_j}(m_i v_j + p\delta_{ij}) &= 0, \qquad i = 1, 2, && (2)\\
\partial_t(\rho E) + \nabla_x \cdot (H\mathbf{m}) &= 0. && (3)
\end{aligned}
$$

With $\mathbf{u} = (\rho, m_1, m_2, \rho E)^T$ this gives

$$\mathbf{u}_t + \nabla_x \cdot \mathbf{f}(\mathbf{u}) = \mathbf{0}.$$

Closed by equation of state $p = (\gamma - 1)\rho e$.

Finite volume scheme

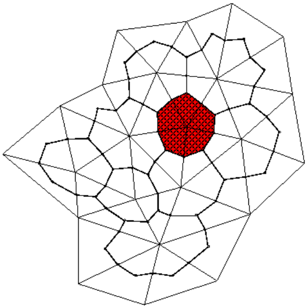
Integration over control volume σ and the divergence theorem give

$$\frac{d}{dt}\int_{\sigma} \mathbf{u}\,dx + \oint_{\partial\sigma} (\mathbf{f}_1(\mathbf{u})n_1 + \mathbf{f}_2(\mathbf{u})n_2)\,ds = 0.$$

Considering mean values in each cell

$$\mathbf{u}_i(t) := \frac{1}{|\sigma_i|}\int_{\sigma_i} \mathbf{u}\,dx$$

and a polygonal mesh, we obtain:



$$\frac{d}{dt}\mathbf{u}_i(t) = -\frac{1}{|\sigma_i|}\sum_{j\in N(i)}\sum_{k=1}^{2}\frac{|l_{ij}^k|}{2}\left(\sum_{\ell=1}^{2}\mathbf{f}_\ell\left(\mathbf{u}(x_{ij}^k,t)\right)\mathbf{n}_{ij,\ell}^k\right).$$

Discrete equations

- Numerical flux function: AUSMDV or low Mach preconditioned Lax-Friedrichs-scheme

$$\mathbf{H}^{LF}(\mathbf{u}_L, \mathbf{u}_R; \mathbf{n}) = \frac{1}{2}(\mathbf{f}(\mathbf{u}_L) + \mathbf{f}(\mathbf{u}_R))\mathbf{n} - \frac{1}{2}\mathbf{D}(\mathbf{u}_L, \mathbf{u}_R; \mathbf{n}) \cdot (\mathbf{u}_R - \mathbf{u}_L)$$

- Here steady state problems, thus implicit Euler in time:

$$\mathbf{\Omega}\,\underline{\mathbf{u}}^{n+1} = \mathbf{\Omega}\,\underline{\mathbf{u}}^{n} + \Delta t \mathbf{H}(\underline{\mathbf{u}}^{n+1})$$

- Solution of nonlinear equation system with one Newton step per timestep:

$$\mathbf{A}\Delta\underline{\mathbf{u}} = \mathbf{rhs}(\underline{\mathbf{u}}^{n})$$

The linear system

- System matrix: $\mathbf{A} = \left[\mathbf{\Omega} + \Delta t \frac{\partial \mathbf{H}(\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} \right]_{\underline{\mathbf{u}}^n}$.
- This system is sparse, nonnormal, not diagonally dominant and sometimes ill conditioned.
- Small Δt means easy systems, but low convergence rate of nonlinear iteration.
- Linear equation systems solved with right preconditioned BiCGSTAB.

The reference preconditioning technique

- Preconditioning via ILU(0).
- As the matrices are not independent, the ILU(0) is stored and recomputed every k timesteps (Freezing + Recomputing).
- The optimal choice of k depends on the precise problem and is between 5 and 50.
- For a good k, this leads to effective preconditioners, so that the systems take between 5 and 30 BiCGSTAB iterations.
- Over the last decades, lot of research on this. Today, improvement of preconditioner alone difficult.
- Idea: Try to share some of the computational effort among the whole sequence.

Outline

- The Finite Volume Scheme
- Preconditioner Updates
- Numerical Results

Notation and Setting

- Given a sequence of linear equation systems of same dimension.
- Reference system $\mathbf{Ax} = \mathbf{b}$ with reference preconditioner $\mathbf{M} = (\mathbf{LD})\mathbf{U}$ or $\mathbf{M} = \mathbf{L}(\mathbf{DU})$.
- $\mathbf{D}$ is blockdiagonal, $\mathbf{L}$, $\mathbf{U}$ block triangular with ones on the diagonal.
- Current system: $\mathbf{A}^+\mathbf{x} = \mathbf{b}^+$.
- Difference matrix $\mathbf{B} = \mathbf{A} - \mathbf{A}^+$.

Related Work

We look for preconditioner $\mathbf{M}^{+}$ that is obtained algebraically.
Related work:

- Approximate updates for the SPD case (Meurant 2001, Benzi, Bertaccini 2003, Bertaccini 2004)
- Updating for nonsymmetric systems for nonblock preconditioners (Duintjer Tebbens, Tůma 2007)

Trivially: Ideal preconditioner of same accuracy as $\mathbf{M}$ is given by

$$\|\mathbf{A} - \mathbf{M}\| = \|\mathbf{A}^{+} - (\mathbf{M} - \mathbf{B})\|.$$

Problem: Systems with $\mathbf{M} - \mathbf{B}$ in general not easy to solve!

Idea

Consider cheap approximations of $\mathbf{M} - \mathbf{B}$ that lead to useful preconditioners. If the upper triangle contains significant information:

$$\mathbf{M} - \mathbf{B} = \mathbf{L}(\mathbf{DU} - \mathbf{L}^{-1}\mathbf{B}) \approx \mathbf{L}(\mathbf{DU} - \mathbf{B})$$

Then approximate $\mathbf{DU} - \mathbf{B}$ via $\mathit{btriu}(\mathbf{DU} - \mathbf{B})$ to obtain:

$$\mathbf{M}^{+} = \mathbf{L}(\mathbf{DU} - \mathit{btriu}(\mathbf{B})).$$

Or otherwise by same reasoning from $\mathbf{M} - \mathbf{B} = (\mathbf{LD} - \mathbf{BU}^{-1})\mathbf{U}$:

$$\mathbf{M}^{+} = (\mathbf{LD} - \mathit{btril}(\mathbf{B}))\mathbf{U}.$$

Properties of the new method

Theorem (Accuracy)

Let $\|\cdot\|$ *be* $\|\cdot\|_F$. *If*

$$\rho = \frac{\|btril(\mathbf{B})(\mathbf{I}-\mathbf{U})\|\,(2\|\mathbf{E}-bstriu(\mathbf{B})\| + \|btril(\mathbf{B})(\mathbf{I}-\mathbf{U})\|)}{\|btril(\mathbf{B})\|^2} < 1,$$

where bstriu denotes the block strict upper triangular part, then the accuracy $\|\mathbf{A}^+ - (\mathbf{LD}-btril(\mathbf{B}))\mathbf{U}\|$ *of the updated preconditioner is higher than the accuracy of the frozen preconditioner* $\|\mathbf{A}^+ - \mathbf{LDU}\|$ *with*

$$\|\mathbf{A}^+ - (\mathbf{LD}-btril(\mathbf{B}))\mathbf{U}\| \le \sqrt{\|\mathbf{A}^+ - \mathbf{LDU}\|^2 - (1-\rho)\|btril(\mathbf{B})\|^2}.$$

Chosing the right Triangle

Three different options (using the Frobenius-Norm):

- Stable update criterion: $\|\mathbf{U}-\mathbf{I}\|$ and $\|\mathbf{L}-\mathbf{I}\|$
- Information flow criterion: $\|btril(\mathbf{B})\|$ and $\|btriu(\mathbf{B})\|$
- Unscaled stable update criterion: Compare $\|\mathbf{D}-\mathbf{DU}\|$ with $\|\mathbf{LD}-\mathbf{D}\|$.

Two problems:

1. A criterion query is not without cost and is not necessary every step. But how often is useful?
2. Every scaling with $\mathbf{D}^{-1}$ is costly due to blockstructure.

Here: Periodic recomputation of preconditioner, only then a query.

Steady and unsteady flows

- While the update is cheap, it is not without cost. Especially near steady state, updating becomes unnecessary.
- Therefore, store the number of iterations in the first step ($iter_0$).
- Do not update in the second step.
- In the following steps, switch to updating, once the number of iterations has crossed a threshhold ($> iter_0 + k$)
- Reset after recomputation of reference preconditioner.

Outline

- The Finite Volume Scheme
- Preconditioner Updates
- Numerical Results

Results Cylinder, Supersonic

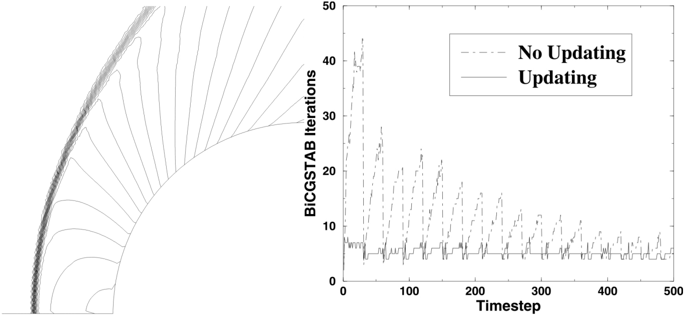


Figure: Pressure Isolines, steady state of a Quartercylinder at Mach 10 after 3000 timesteps and BiCGSTAB iterations for the first 500. 20994 cells, 83976 unknowns.

Results Cylinder, Supersonic

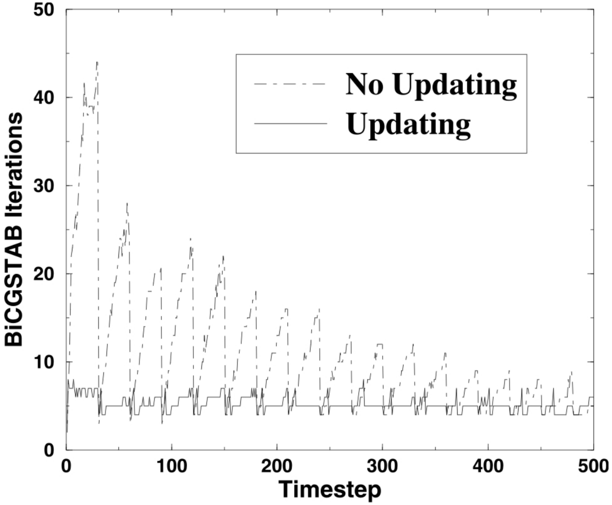


Figure: BiCGSTAB iterations for the first 500 timesteps. 20994 cells, 83976 unknowns.

Results Cylinder, Supersonic

	No updating		Stable		Unsc. stable		Inf. flow	
Per.	Iter.	Time	Iter.	Time	Iter.	Time	Iter.	Time
10	10683	7020	11782	7284	11782	7443	11782	7309
20	12294	6340	12147	6163	12147	6300	12147	6276
30	13787	7119	12503	5886	12503	5894	12503	5991
40	15165	6356	12916	5866	12916	5835	12916	5856
50	16569	6709	11962	5821	13139	5670	13139	5925

CPU-Times and Iterationnumbers after 3000 timesteps.

Speedup only during first 500 timesteps!

Results Cylinder, Supersonic

Here, unknowns were reordered to respect physical flow of information, leading to a mostly triangular matrix. Thus:

1. Algorithm always choses the lower left part.
2. Updates are close to refactorization in every step.

i	$\|A^{(i)} - LDU\|_F$	$\|A^{(i)} - M^{(i)}\|_F$	Bound from th.	ρ
2	37.454	34.277	36.172	0.571
3	37.815	34.475	36.411	0.551
4	42.096	34.959	36.938	0.245
5	50.965	35.517	37.557	0.104
6	55.902	36.118	38.308	0.083

Results NACA-Profile, Mach 0.8

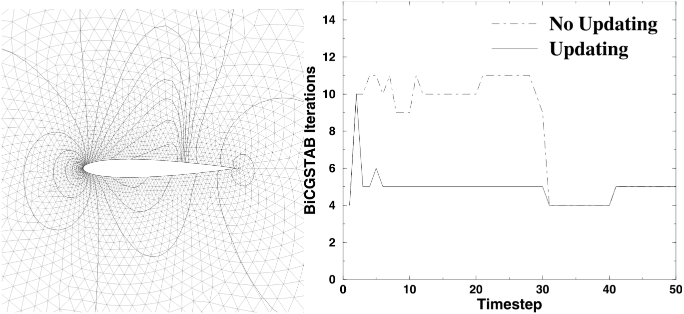


Figure: Pressure Isolines, steady state of a NACA0012-profile at Mach 0,8 after 750 time steps and BiCGSTAB iterations for the first 50. 4605 cells and 18420 unknowns.

Results NACA-Profile, Mach 0.8

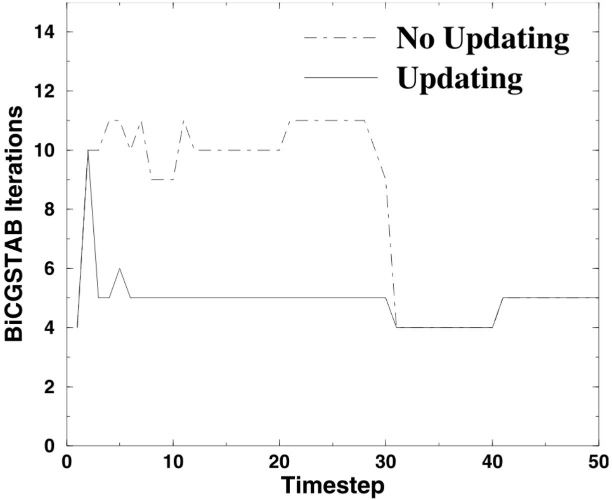


Figure: BiCGSTAB iterations for the first 50 timesteps. 4605 cells and 18420 unknowns.

Results NACA-Profile, Mach 0.8

	No updating		Stable		Unsc. stable		Inf. flow	
Per.	Iter.	Time	Iter.	Time	Iter.	Time	Iter.	Time
10	5375	543	5336	498	5336	494	5336	483
20	5454	497	5364	469	5364	468	5364	459
30	5526	491	5379	464	5379	467	5379	453
40	5558	491	5411	456	5411	462	5411	452
50	5643	525	5413	466	5413	470	5413	448

CPU-Times and Iterationnumbers after 750 timesteps.

Summary and Outlook

- Implicit Finite Volume method for the Euler equations.
- Black Box Updating technique for the sequence of linear equation systems for the Block-ILU-preconditioner.
- New technique is better than the reference scheme for unsteady situations.
- New technique is not worse than freezing with periodic recomputing for steady situations.
- New technique is robust for a large class of problems.
- Outlook: Unsteady test cases. Matrix-free? Spatial adaptation?