Performance Imrovement of Algebraic Multigrid Solver by Vector Sequence Extrapolation

A. Jemcov[1] J.P. Maruszewski[1] H. Jasak[2]

[1]ANSYS-Fluent Inc. [2]Wikki Ltd.

May 30, 2007

Outline

Outline

Outline

1. Introduction
   - Goals of Presentation
   - Motivation
2. Extrapolation Methods
   - Fixed-Point Methods for Linear Systems of Equations
   - Projective Forward Extrapolation
   - Minimal Polynomial Extrapolation
   - Reduced Rank Extrapolation
3. Computational Examples
   - LES of Forward Facing Step
   - Free-Surface Simulation of Droplet Impact
4. Conclusions
   - Summary

Outline

Outline

Goals

- Performance improvement of Agglomerative Algebraic Multigrid Method (AAMG) for large matrices is examined
- Performance improvement must be achieved through code non-intrusive techniques that can be easily parallelized
- Hybrid method that uses Vector Sequence Extrapolation and AAMG solver is proposed
- In particular, three hybrid methods are considered:
  - Projective Forward Extrapolation (PFE)
  - Minimal Polynomial Extrapolation (MPE)
  - Reduced Rank Extrapolation (RRE)
- Compare performance of three methods and make recommendations

Outline

Motivation

- Implicit segregated pressure based solvers used in this study (PISO, SIMPLE, etc.)
- Implicit nature of the solvers leads to linear systems of equations for pressure and momentum that need to be solved
- Momentum equations are mostly dominated by local structures, traditional stationary iterative methods are (mostly) effective
- Pressure equation has a global nature dominated by many scales, convergence of traditional stationary iterative methods can be slow
- Geometric complexity and low quality mesh are another source of difficulties

AMG Method

- Implicit discretization of pressure equation results in sparse linear problem

$$\boldsymbol{Ax} = \boldsymbol{b}$$

- Matrix $\boldsymbol{A}$ stored in arrow format with sparsity pattern stored in compressed row format
- Stationary iterative methods obtained by decomposing matrix $\boldsymbol{A}$

$$\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$$

- Depending on choice of $\boldsymbol{M}$ and $\boldsymbol{N}$, Gauss-Seidel, Jacobi, SSOR, etc. are obtained

AMG Method

- In multigrid methods, stationary iterative methods are used as smoothers rather than solvers
- Acceleration is obtained through application of smoothers on hierarchy of matrix levels obtained by coarsening procedure
- Restriction $\boldsymbol{R}_n^{n+1}$ and prolongation operators $\boldsymbol{P}_{n+1}^n$ are obtained through process of agglomeration
- Coarse matrices are obtained through projection

$$\boldsymbol{A}^{n+1} = \boldsymbol{R}_n^{n+1} \boldsymbol{A}^n \boldsymbol{P}_{n+1}^n$$

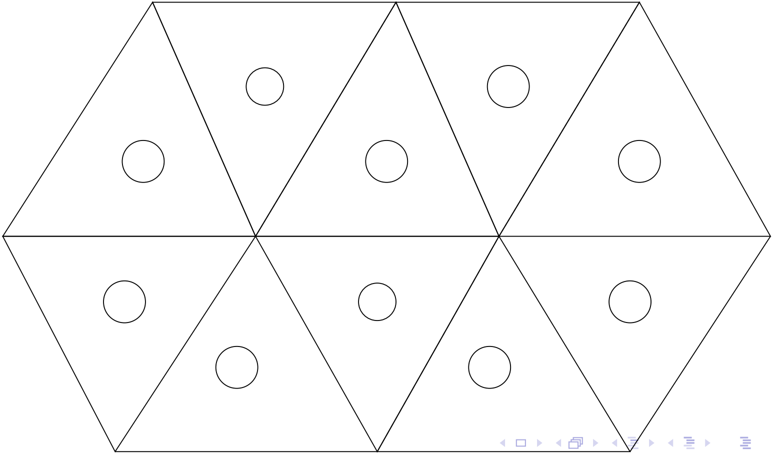
- Restriction and prolongation operators are also used to transfer solution and residual to and coarse levels

$$\boldsymbol{x}^{n+1} = R_n^{n+1} \boldsymbol{x}^n$$

$$\boldsymbol{x}^n = P_{n+1}^n \boldsymbol{x}^{n+1}$$

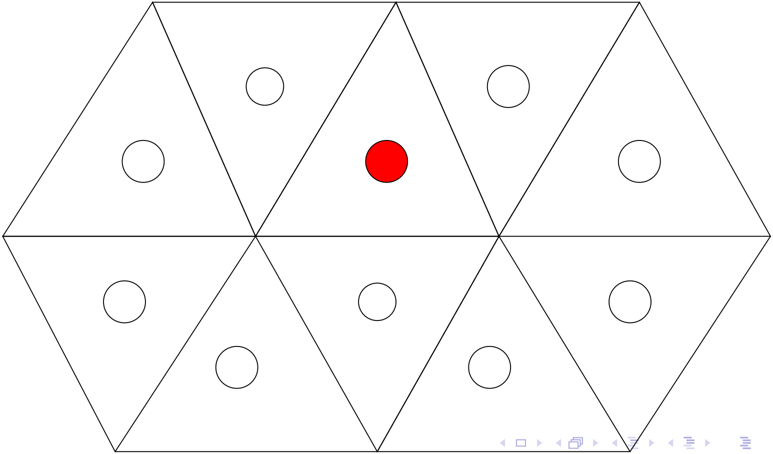
AMG Coarsening

- Agglomeration process



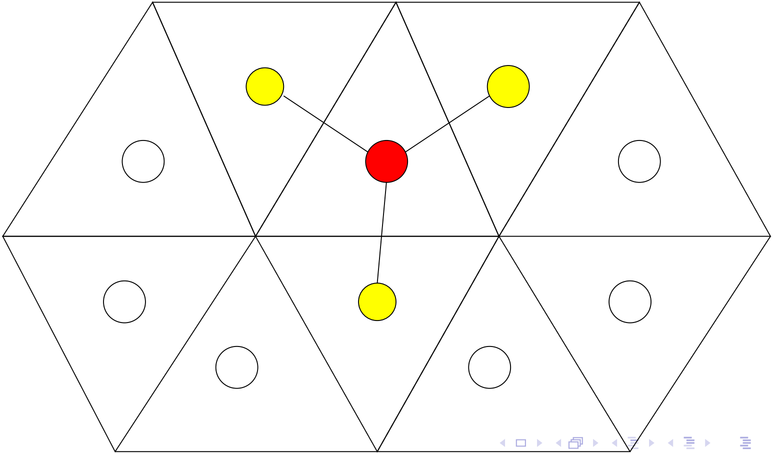
AMG Coarsening

- Agglomeration process



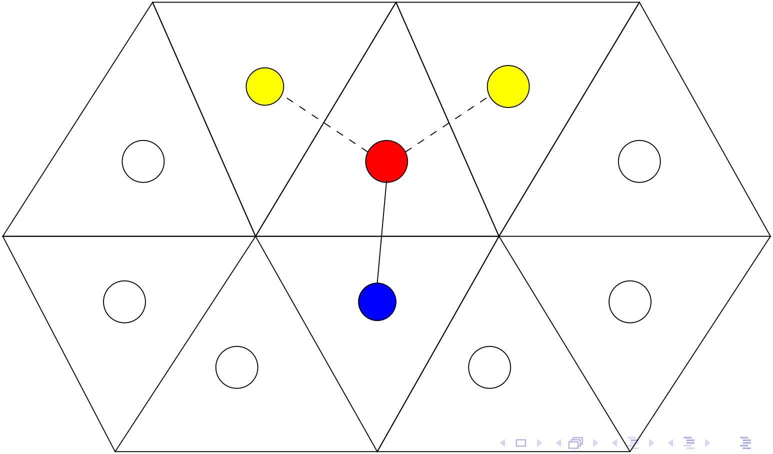
AMG Coarsening

- Agglomeration process



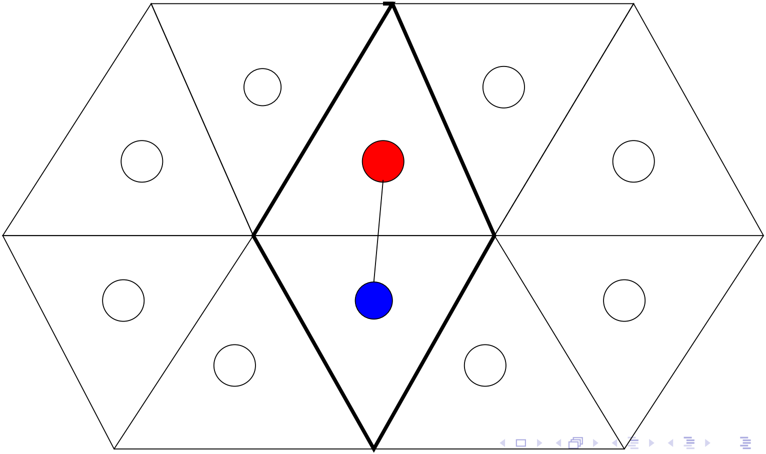
AMG Coarsening

- Agglomeration process



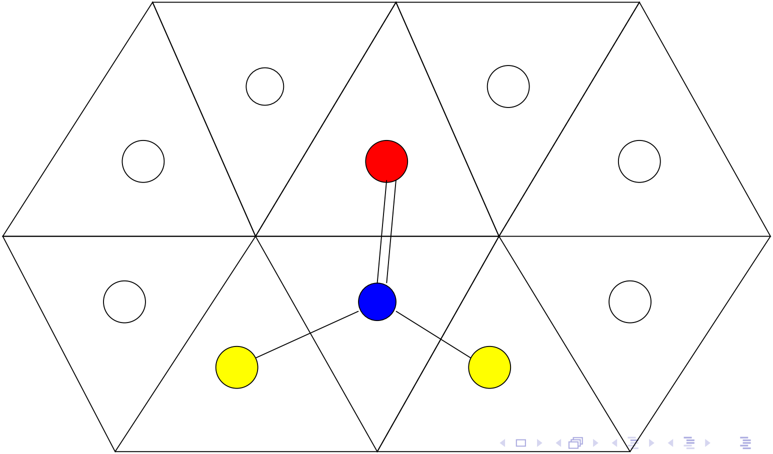
AMG Coarsening

- Agglomeration process



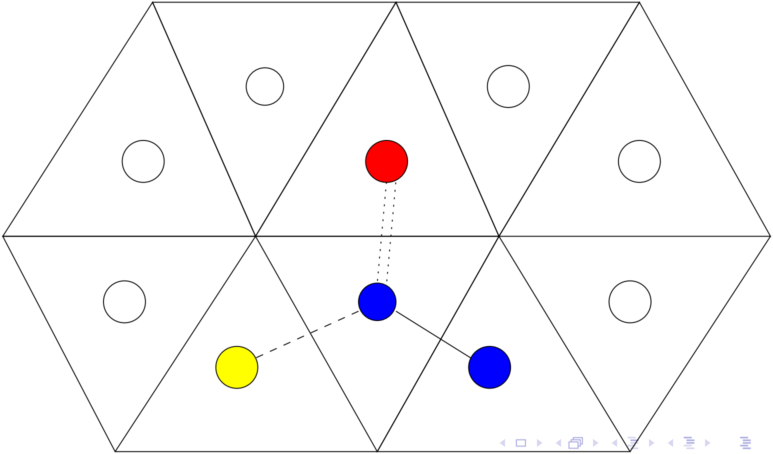
AMG Coarsening

- Agglomeration process



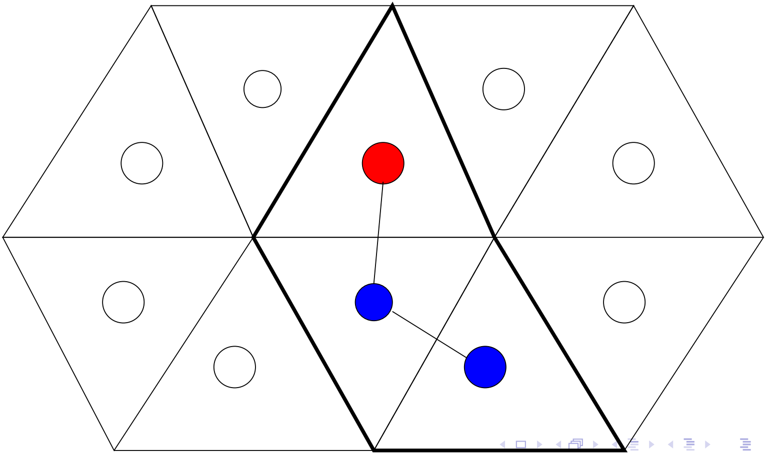
AMG Coarsening

- Agglomeration process



AMG Coarsening

- Agglomeration process



μ-Cycle($\mathbf{x}^n, \mathbf{r}^n$)

Create multigrid levels:

$\mathbf{A}^n$, $\mathbf{R}_n^{n+1}$, $\mathbf{P}_{n+1}^n$, $n = 0, 1, 2, ..., N-1$

for $n = 0$ *to* $N-1$

ν_1 *pre-smoothing sweeps:*

solve $\mathbf{A}^n\mathbf{x}^n = \mathbf{b}^n$, $\mathbf{r}^n = \mathbf{b}^n - \mathbf{A}^n\mathbf{x}^n$

if $n \,! = N-1$

$\mathbf{b}^{n+1} = \mathbf{R}_n^{n+1}\mathbf{r}^n$

$\mathbf{x}^{n+1} = 0$

$\mathbf{x}^{n+1} = \mu$-Cycle($\mathbf{x}^{n+1}, \mathbf{r}^{n+1}$) μ times

Correct $\mathbf{x}_{new}^n = \mathbf{x}^n + \mathbf{P}_{n+1}^n\mathbf{x}^{n+1}$

end

ν_2 *post-smoothing sweeps:*

solve $\mathbf{A}^n\mathbf{x}_{new}^n = \mathbf{b}^n$

end

AMG Method

- Agglomerative Algebraic Multigrid Method is favored in computationally intensive calculations that involve matrices with M-matrix properties
- AAMG Algorithm is very cheap to implement - doesn't require storage of prolongation and restriction operators (matrices)
- Every equation in fine level has a coarse equation representation
- Assumption is that the error is homogeneous - prolongation coefficients are all equal
- Convergence of AAMG can be improved by using scaling of corrections (variational formulation)

Outline

Fixed-Point Methods

- Consider again linear problem

$$\mathbf{Ax} = \mathbf{b}$$

- With splitting $\mathbf{A} = \mathbf{M} - \mathbf{N}$ stationary iterative method is obtained

$$\mathbf{x}^{(\nu+1)} = \mathbf{Rx}^{(\nu)} + \mathbf{M}^{-1}\mathbf{b}$$

- Here $\mathbf{R}$ is the iteration matrix

$$\mathbf{R} = \mathbf{M}^{-1}\mathbf{N}$$

Fixed-Point Methods

- Define error as the difference between current iterate $\mathbf{x}^{(\nu)}$ and the fixed-point $\mathbf{x}^{(*)}$

$$\mathbf{e}^{(\nu)} = \mathbf{x}^{(\nu)} - \mathbf{x}^{*}$$

- Error propagation equation

$$\mathbf{e}^{(\nu+1)} = \mathbf{R}\mathbf{e}^{(\nu)}$$

- The iteration matrix possesses the recursive property:

$$\mathbf{e}^{(\nu+1)} = \mathbf{R}^{\nu}\mathbf{e}^{(0)}$$

Fixed-Point Methods

- Define corrections

$$\triangle \mathbf{x}^{(\nu)} = \mathbf{x}^{(\nu)} - \mathbf{x}^{(\nu-1)}$$

- The recursive property is also applicable to corrections

$$\triangle \mathbf{x}^{(\nu+1)} = \mathbf{R}^{\nu} \triangle \mathbf{x}^{(0)}$$

- The importance of the recursive property is that generates Krylov subspace

$$\mathcal{K} = span(\triangle \mathbf{x}^{(0)}, \triangle \mathbf{x}^{(1)}, \cdots, \triangle \mathbf{x}^{(n)})$$

- Therefore, solution can be sought in that subspace in the following form:

$$\mathbf{x}^{*} = \mathbf{x}^{(\nu)} + \sum_{\nu} \alpha_{\nu} \triangle \mathbf{x}^{(\nu)}$$

Fixed-Point Methods

- The fundamental question is: ”How α_ν can be determined to accelerate iterative process
- α_ν are obtained by introducing additional conditions (constraints) in the Krylov subspace
- Three methods of selecting α_ν presented here
- In practice, finding the solution is implemented as a restarted procedure

$$\mathbf{x}^n = \mathbf{x}^{(n-k)} + \sum_{\nu=0}^{k} \alpha_\nu \triangle \mathbf{x}^{(\nu)}$$

- It is important to determine the proper dimension k of the restart space to get the best performance

Outline

Projective Forward Extrapolation

- The simplest approach to determining *alpha*$_{(\nu)}$ is to use a constant value and restrict Krylov subspace to two dimensions

$$\mathbf{x}^* = \mathbf{x}^{(\nu)} + \alpha \triangle \mathbf{x}^{(\nu)}$$

- If α is set to 1, linear extrapolation based on first order difference is obtained
- This is the PFE-AAMG algorithm
- To make this algorithm effective, number of AAMG smoothing steps must be performed before extrapolation
- In applications we use 10 AAMG smoothing steps
- Method is attractive from the efficiency point of view due to low storage requirements

Projective Forward Extrapolation

- PFE-AAMG method was inspired by work of Gear at al. (2005)
- Fixed-Point iteration is treated as discrete dynamical system with prescribed mapping

$$\mathbf{x}^{(\nu+1)} = \mathbf{R}\mathbf{x}^{(\nu)} + \mathbf{M}^{-1}\mathbf{b}$$

- The idea is to remove fast transients and then to project onto a slow manifold
- This method works well for ODEs and it assumes differentiability of the smooth manifold
- PFE method can also be used for nonlinear problems with fast and slow time-scales

Outline

Minimal Polynomial Extrapolation

- If higher Krylov subspace of higher dimensionality is used, PFE is not effective due to the assumptions of differentiability
- Instead, natural condition of the restarted method at convergence is

$$\mathbf{x}^* = \mathbf{x}^* + \sum_{\nu=0}^{k} \alpha_\nu \mathbf{e}^\nu$$

- In other words, $\alpha_{(\nu)}$ must satisfy the following equation

$$\sum_{\nu=0}^{k} \alpha_{\nu=0}^{k} \mathbf{e}^\nu = 0$$

- Ideally, this condition should be connected to the iterative process so that α_ν are determined from that process

Minimal Polynomial Extrapolation

- One way to connect extrapolating coefficients and iterative process is to use property of Krylov subspace methods that are known to implicitly define minimal polynomial P_k related to iteration matrix $\boldsymbol{R}$

$$P_k(\mathbf{R})\triangle\mathbf{x}^0 = \sum_{\nu=0}^{k} c_\nu \mathbf{R}^\nu \triangle\mathbf{x}^{(0)} = 0$$

- Further manipulation leads to

$$\sum_{\nu=0}^{k} c_\nu \mathbf{R}^\nu \triangle\mathbf{x}^{(0)} = (\mathbf{I}-\mathbf{R})\sum_{\nu=0}^{k} c_\nu \mathbf{e}^{(\nu)} = 0$$

- Since $\mathbf{I}-\mathbf{R}$ has a full rank, we obtain

$$\sum_{\nu=0}^{k} c_\nu \mathbf{e}^{(\nu)} = 0.$$

Minimal Polynomial Extrapolation

- Coefficients c_ν are obtained by requiring that the following L_2 norm is minimized is

$$argmin_{\alpha_\nu} \left\| \mathbf{x}^{(\nu)} + \sum_{\nu=0}^{k} \alpha_\nu \triangle \mathbf{x}^{(\nu)} \right\|_2$$

- In practical terms we collect k correction vectors $\triangle \mathbf{x}^{(\nu)}$ and form rectangular matrix

$$\mathbf{U}_{k-1} = \left[\triangle \mathbf{x}^{(0)}, \triangle \mathbf{x}^{(1)}, \cdots, \triangle \mathbf{x}^{(k-1)} \right]$$

Minimal Polynomial Extrapolation

- The following over-determined problems is solved

$$\mathbf{U}_{k-1}\tilde{c} = \triangle \mathbf{x}^{(k)}$$

- MPE method is obtained when the following constraint is used

$$c_k = 1$$

- α_ν coefficients are recovered by a simple scaling

$$\alpha_\nu = \frac{c_\nu}{\sum_{v=0}^{k-1} c_\nu}$$

- MPE method is equivalent to Full Orthogonal Method in Linear Algebra

Outline

Reduced Rank Extrapolation

- Reduced Rank Extrapolation is recovered by selecting the constraints to be

$$\sum_{\nu}^{k} \alpha_\nu = 1$$

- This leads to the following over-determined problem

$$\mathbf{U}_k \tilde{\alpha} = 0$$

- α_ν coefficients are obtained from

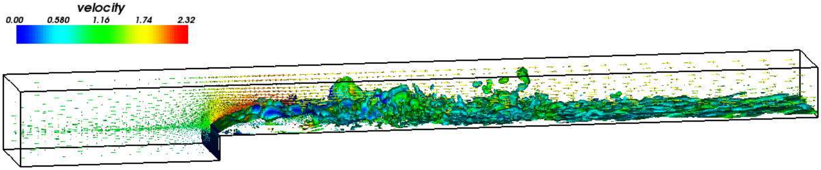
$$\alpha_\nu = c_\nu$$

- RRE method is equivalent to GMRES method in Linear Algebra

Outline

LES

- Consider turbulent flow over forward facing step at $Re = 10000$

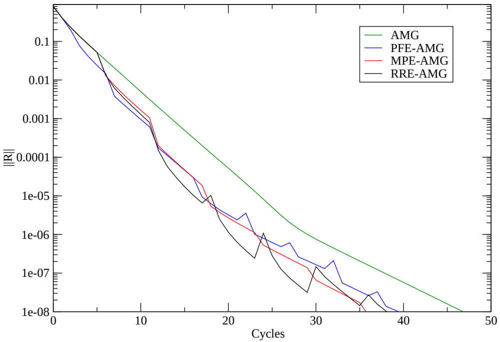


Computational Examples

- Second order accurate in space and time scheme used
- CFL number held at unity
- 2 PISO correctors
- Computational mesh size 660000 cells
- Mesh aggressively graded towards the wall
- Computational hardware: 2.16GHz Intel CoreDuo CPU with 2Gb
- Convergence tolerance for pressure equation set to $10E - 08$

LES

- W-Cycle, Group Size 4, ILU(0) smoother, 0 pre-sweeps and 2 post-sweeps



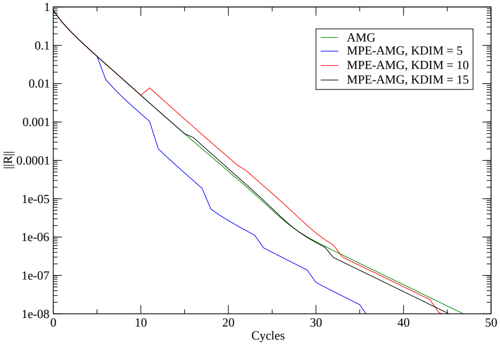
LES

- W-Cycle, Group Size 4, ILU(0) smoother, 0 pre-sweeps and 2 post-sweeps

Method	AMG Cycles	Time [s]
AMG	47	27.95
PFE-AMG	40	24.20
MPE-AMG	36	22.90
RRE-AMG	39	24.58

LES: MPE-AMG

- W-Cycle, Group Size 4, ILU(0) smoother, 0 pre-sweeps and 2 post-sweeps



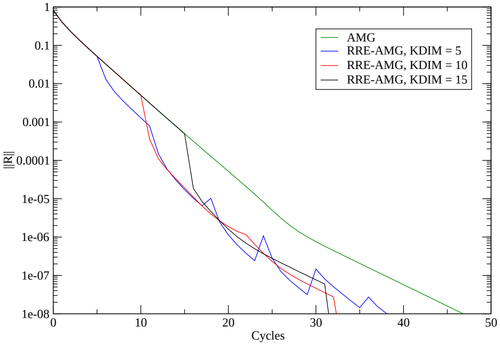
LES: MPE-AMG

- W-Cycle, Group Size 4, ILU(0) smoother, 0 pre-sweeps and 2 post-sweeps

Dimension	AMG Cycles	Time [s]
0	47	27.95
5	36	22.9
10	45	29.24
15	46	29.56

LES-RRE-AMG

- W-Cycle, Group Size 4, ILU(0) smoother, 0 pre-sweeps and 2 post-sweeps



LES: RRE-AMG

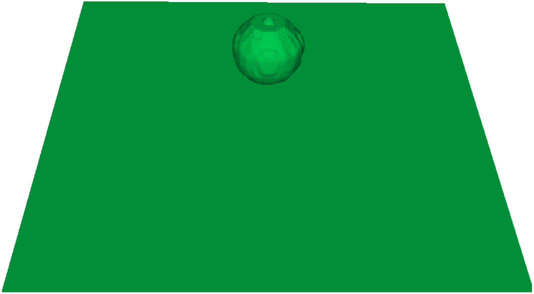
- W-Cycle, Group Size 4, ILU(0) smoother, 0 pre-sweeps and 2 post-sweeps

Dimension	AMG Cycles	Time [s]
0	47	27.95
5	39	24.58
10	33	21.86
15	32	22.27

Outline

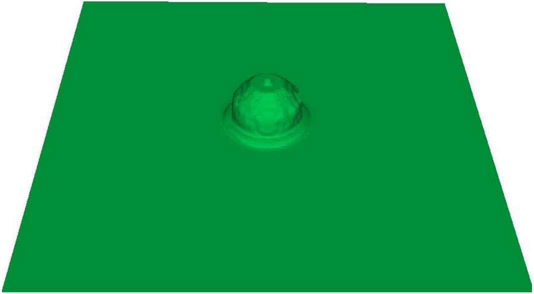
Free Surface

- Droplet free surface at time $t = 2.77325E - 05\,s$



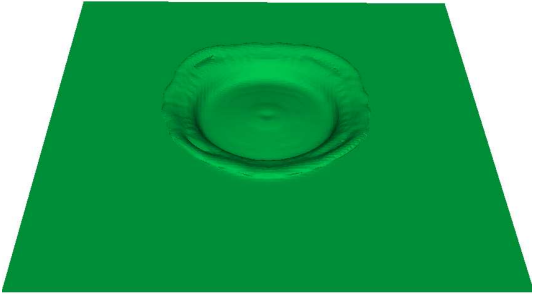
Free Surface

- Droplet free surface at time $t = 4.01785E - 05\,s$



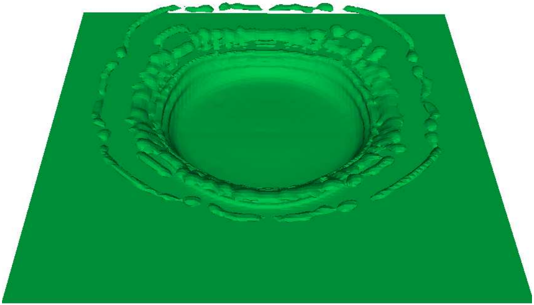
Free Surface

- Droplet free surface at time $t = 6.80427E - 05\,s$



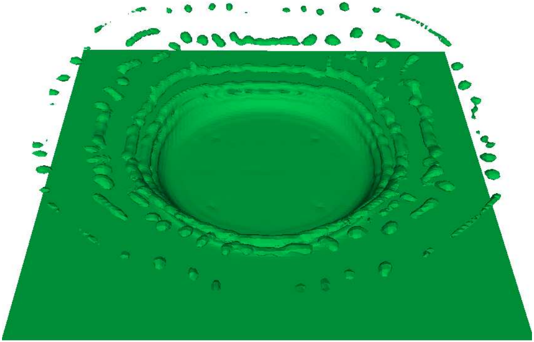
Free Surface

- Droplet free surface at time $t = 6.80427E - 05\,s$



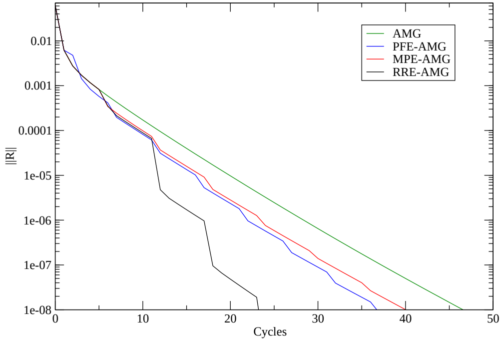
Free Surface

- Droplet free surface at time $t = 8.78372E - 05\,s$



Free Surface

- W-Cycle, Group Size 4, ILU(0) smoother, 0 pre-sweeps and 2 post-sweeps



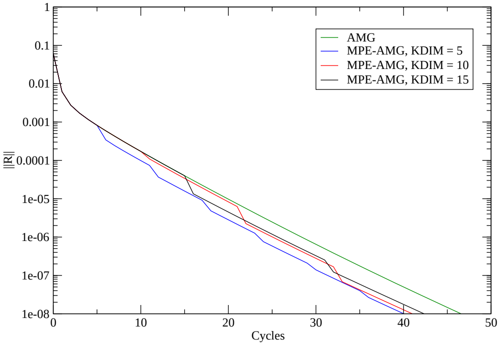
Free Surface: MPE-AMG

- W-Cycle, Group Size 4, ILU(0) smoother, 0 pre-sweeps and 2 post-sweeps

Method	AMG Cycles	Time [s]
AMG	47	63.39
PFE-AMG	37	48.21
MPE-AMG	41	58.90
RRE-AMG	24	36.13

Free Surface: MPE-AMG

- W-Cycle, Group Size 4, ILU(0) smoother, 0 pre-sweeps and 2 post-sweeps



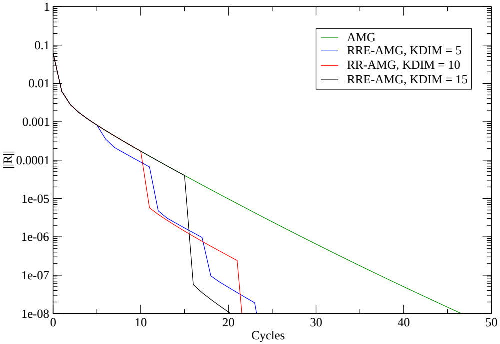
Free Surface: MPE-AMG

- W-Cycle, Group Size 4, ILU(0) smoother, 0 pre-sweeps and 2 post-sweeps

Dimension	AMG Cycles	Time [s]
0	47	63.39
5	41	58.90
10	41	59.95
15	43	63.60

Free Surface: RRE-AMG

- W-Cycle, Group Size 4, ILU(0) smoother, 0 pre-sweeps and 2 post-sweeps



Free Surface: RRE-AMG

- W-Cycle, Group Size 4, ILU(0) smoother, 0 pre-sweeps and 2 post-sweeps

Dimension	AMG Cycles	Time [s]
0	47	63.39
5	41	58.90
10	22	33.18
15	21	32.56

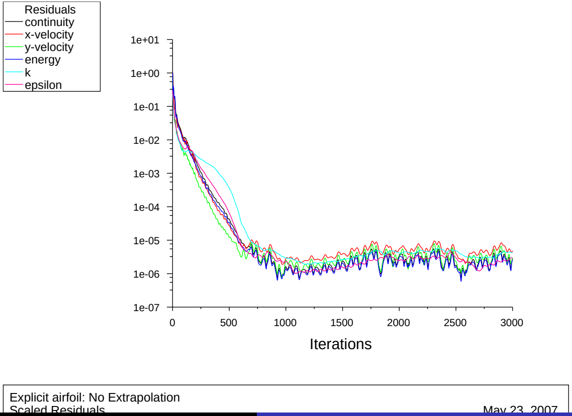
Outline

Conclusions

- All three algorithms show significant improvements of AMG algorithm
- PFE-AMG cheap and simple, not as effective as MPE and RRE
- RRE superior to MPE with the same implementation complexity
- Dimension of restart space is important
- All three algorithms are implemented as wrappers to AMG and easily extend to any linear and nonlinear solver

RRE Acceleration of Explicit Density Density Based Solver

- Baseline run NACA 0012 - no acceleration



RRE Acceleration of Explicit Density Density Based Solver

- Accelerated run NACA 0012 - RRE acceleration