Deformation of CFD Meshes with Anisotropic Cells in a Viscous Boundary Layer Using Line-Implicit Methods

Arne Rempke

German Aerospace Center (DLR), Institute of Software Methods for Product Virtualization, High Performance Computing, Dresden

STAB 9.11.2022, Berlin

DLR

Outline

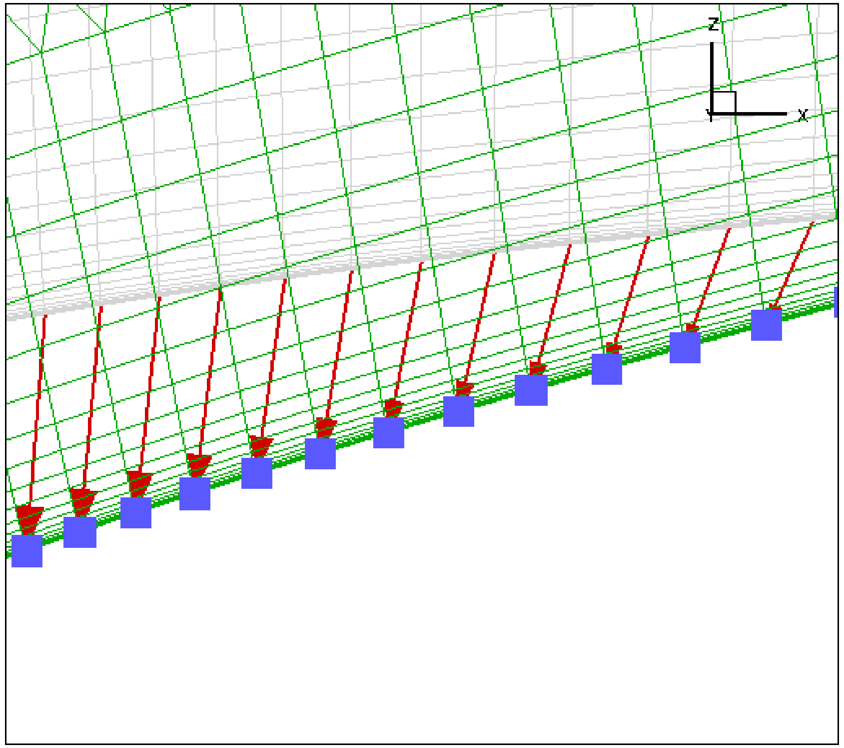
- Elasticity-Analogy Volume CFD Mesh Deformation
- Line-Implicit and Multigrid Methods
- Numerical Results
- Summary

Elasticity-Analogy Mesh Deformation

Elasticity Analogy for Volume Mesh Deformation

Basic idea

- Consider spacial CFD domain as a solid body
- Apply elasticity model with stiffened small elements
  - Virtual material parameters model the behaviour of the body
- Receive deformations for given boundary displacements by solving a partial differential equation



Linear Elasticity Equations

Theory

Strain tensor:

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

Stress tensor:

$$\sigma_{ij} = \frac{E}{1+\nu}\varepsilon_{ij} + \frac{\nu E}{(1+\nu)(1-2\nu)}\varepsilon_{kk}\delta_{ij}$$

Elasticity equation:

$$\frac{\partial \sigma_{ij}}{\partial x_j} = 0, \qquad \text{on } \Omega$$

$$\boldsymbol{u} = \boldsymbol{u_D}, \qquad \text{on } \Gamma_D$$

Implementation

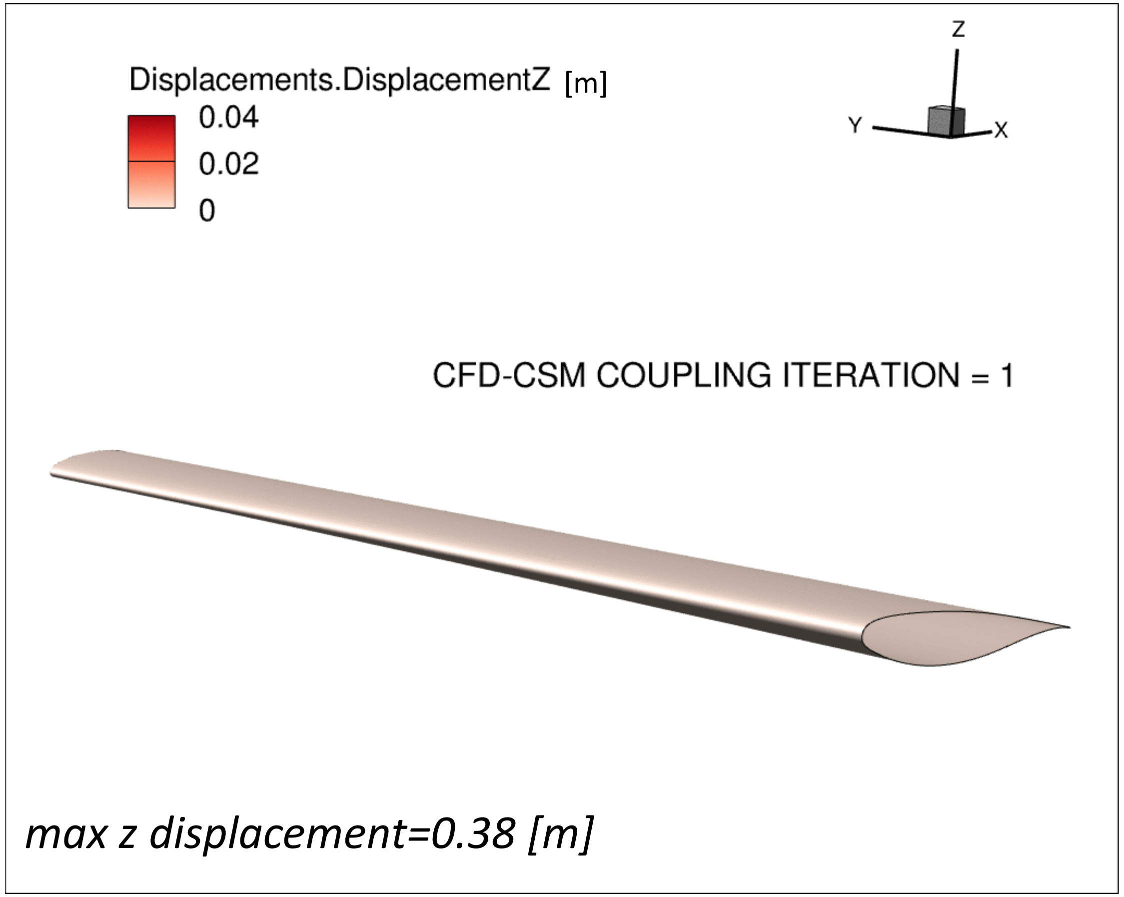
1. Assemble stiffness matrix $\boldsymbol{K}$
2. Apply dirichlet boundary conditions $\boldsymbol{b} = \boldsymbol{I_N K u_D}$ and $\boldsymbol{K_{sym}} = \boldsymbol{I_N K I_N} + \boldsymbol{I_D}$
3. Solve symmetric system

$$\boldsymbol{K_{sym} u} = \boldsymbol{b}$$

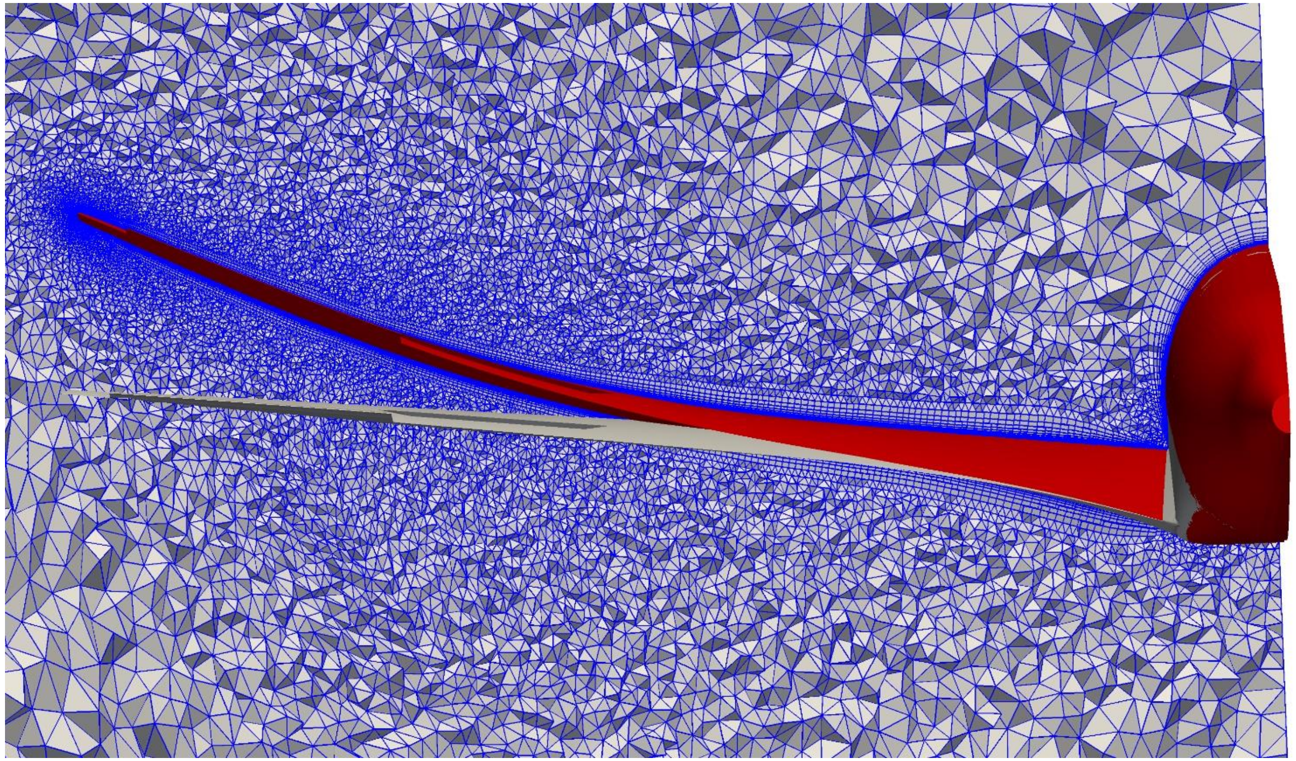
Default material parameters:

- Artificial stiffening $E = \left(\det \frac{\partial \boldsymbol{x}}{\partial \xi}\right)^{-1}$
- Poisson number $\nu = 0.3$

Volume Mesh Deformation for Deflected Wings

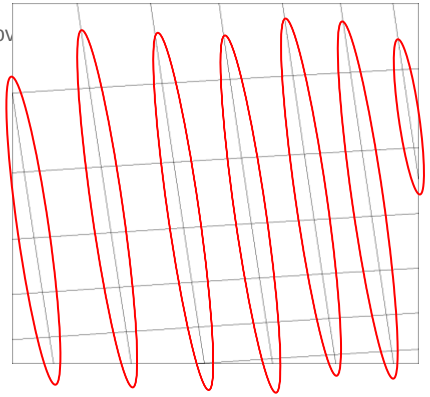


© Marco Cristofaro



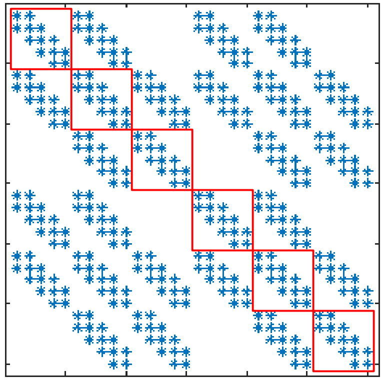
© Lars Reimer

Line-Implicit Methods



What is lines-implicit?

Instructions

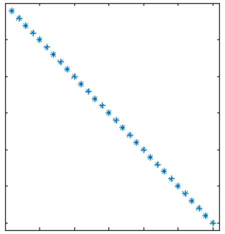


1. Identify one-dimensional lines of connected DoFs within the mesh

2. Each such line corresponds to a tridiagonal submatrix in the Jacobian matrix A (possibly after permuation)

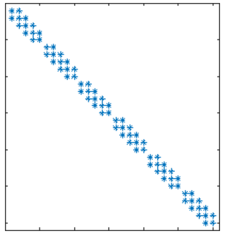
3. A tridiagonal submatrix can be inverted exactly in-place (no infill/approximation) and fast ($\mathcal{O}(n)$) (*Thomas-Algorithm*)

4. This inversion is used instead of the diagonal inversion in a Jacobi-method:



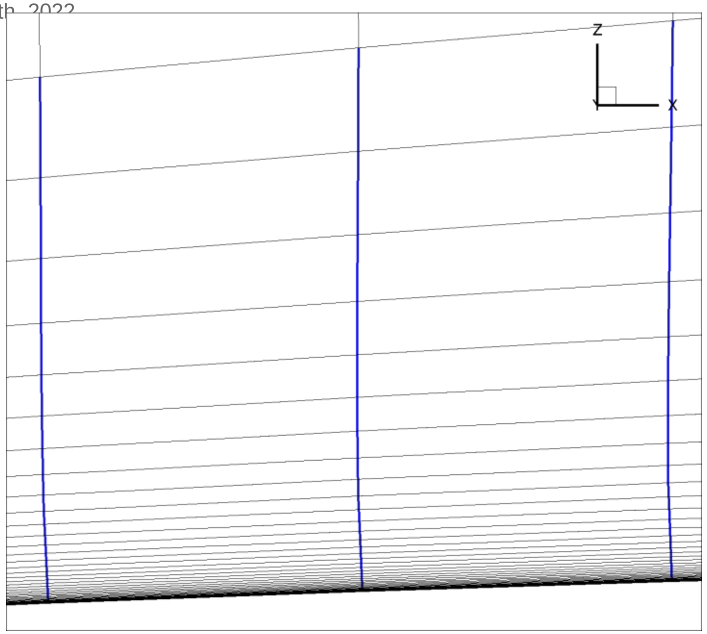
$$x^{(i+1)} := x^{(i)} + D^{-1}\left(b - Ax^{(i)}\right)$$

where



$D := \mathrm{diag}(A)$ (point-implicit) or $D := \mathrm{tridiag}(A)$ (lines-implicit)

Where to use lines-implicit?



Help to handle

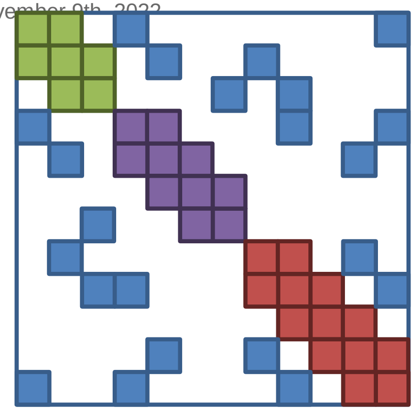
- Anisotropic meshes with cells of high aspect-ratio (e.g. >1000:1)
  ➔ Select lines involving the neighbors with shortest distance

- PDEs with strong advective transport
  ➔ Select lines along transport direction

However:

- Lines need to be chosen carefully!
  If „wrong“ lines are chosen, lines-implicit can become worse than point-implicit: Slower convergence or even divergence

Lines-implicit solving in Spliss



Sparse **block** matrix system solver

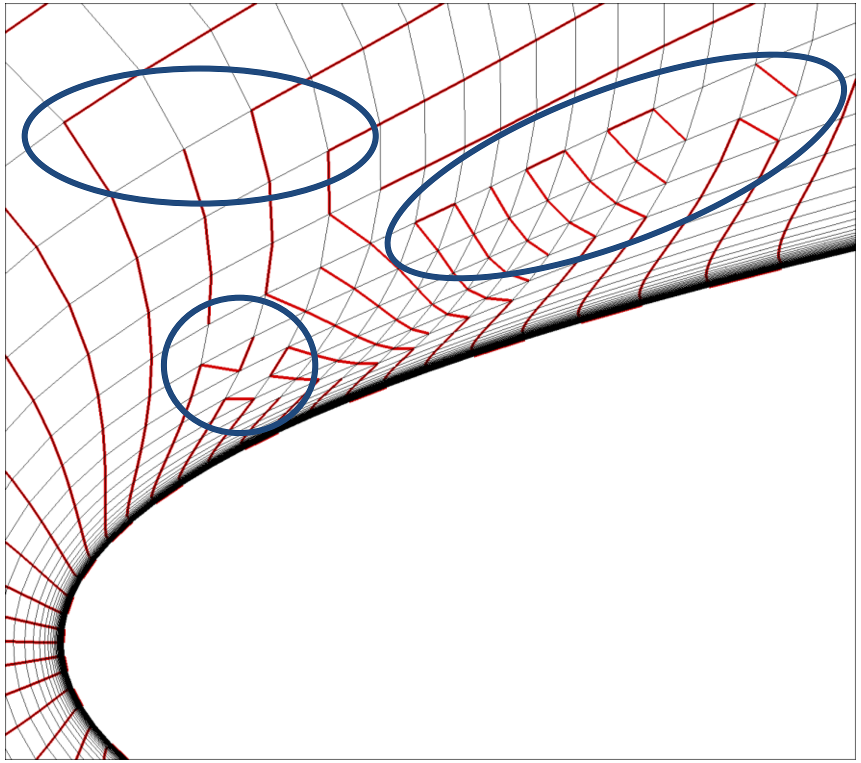
LinesInversion solver component

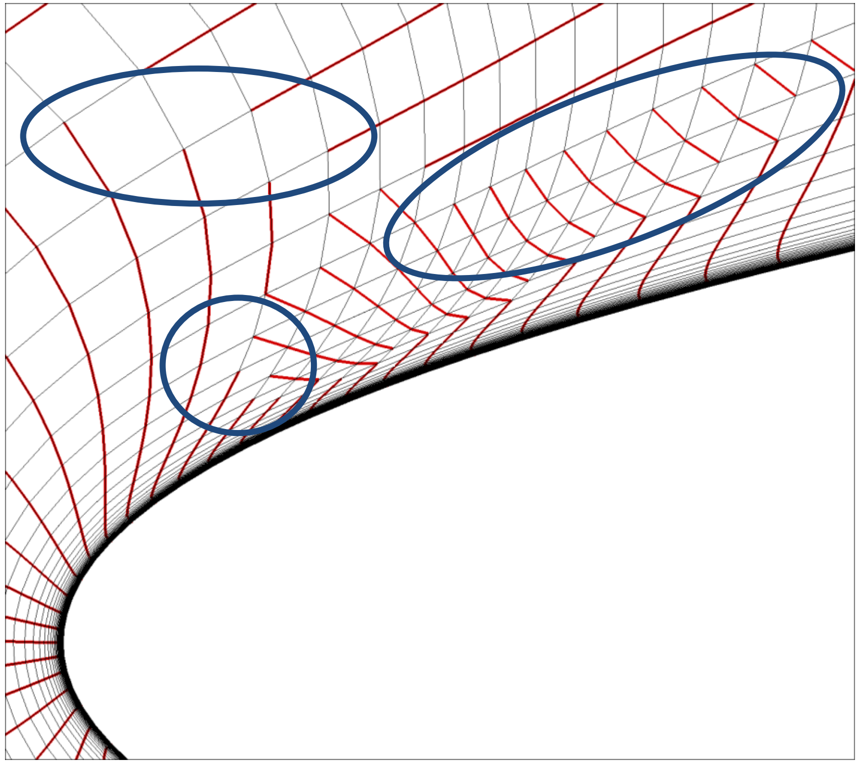
- Solves local block-tridiagonal linear equation systems
  - Each scalar entry from slide 8 is a dense block matrix
- Can be used in combination with other solver components (i.e. not only Jacobi but also e.g. (multi-color) GaussSeidel method)

Helpers to detect lines (e.g. in unstructured meshes)

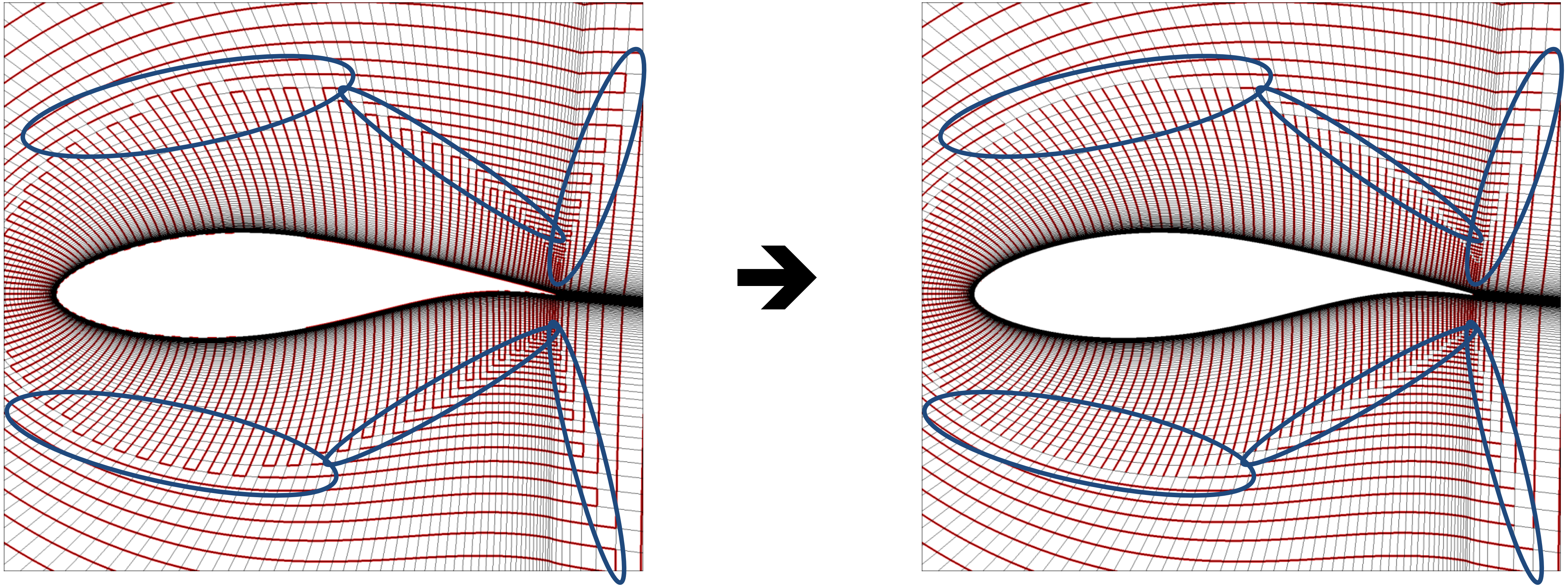
- From matrix connectivity and a user-given coupling strength
- Ensures no loops and crossings are involved when selecting the strongest couplings to form the lines
- New algorithm to detect lines ensures almost all nodes can be connected to lines

Improvements in line detection algorithm

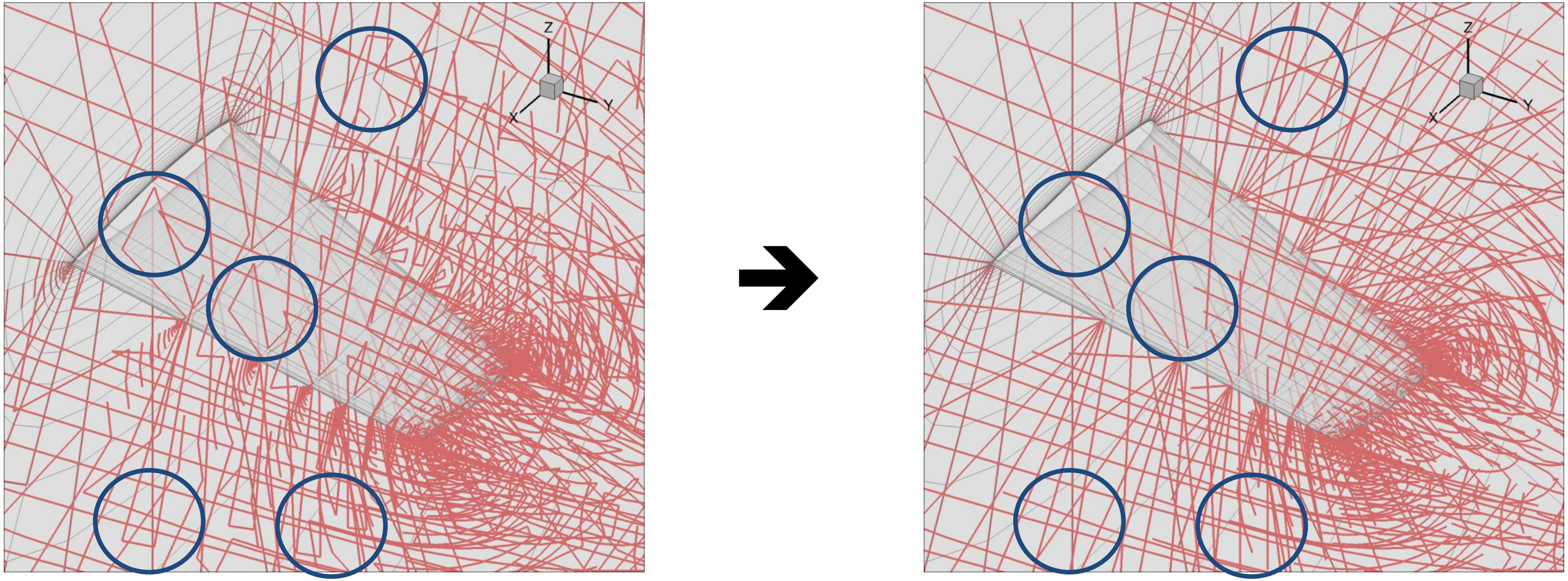




Improvements in line detection algorithm



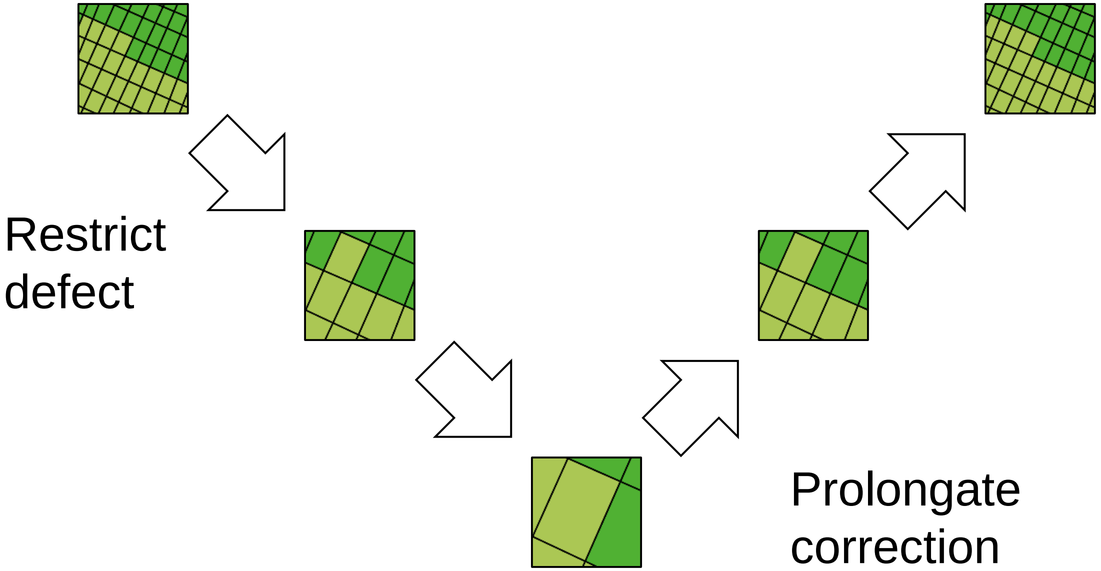
Improvements in line detection algorithm



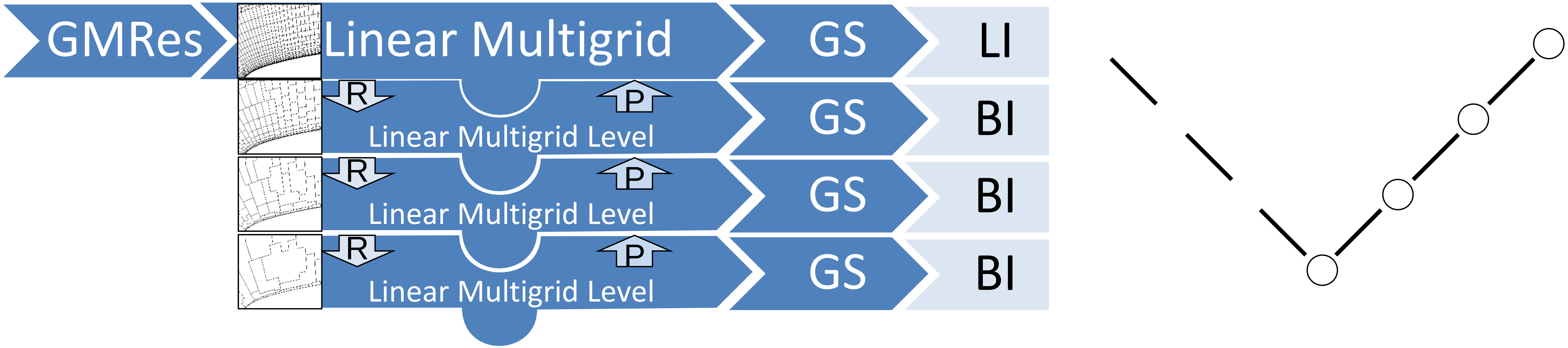
Multigrid Methods

What is multigrid?

- Use multiple levels to solve a problem based on a (structured) grid
- Helps to reduce low-frequency errors if the smoother concentrates on high-frequency errors



Applying GMRes preconditioned by a 4 level V(0,1) MG with GaussSeidel-smoothers on each level, lines-implicit on finest level, point-implicit on others



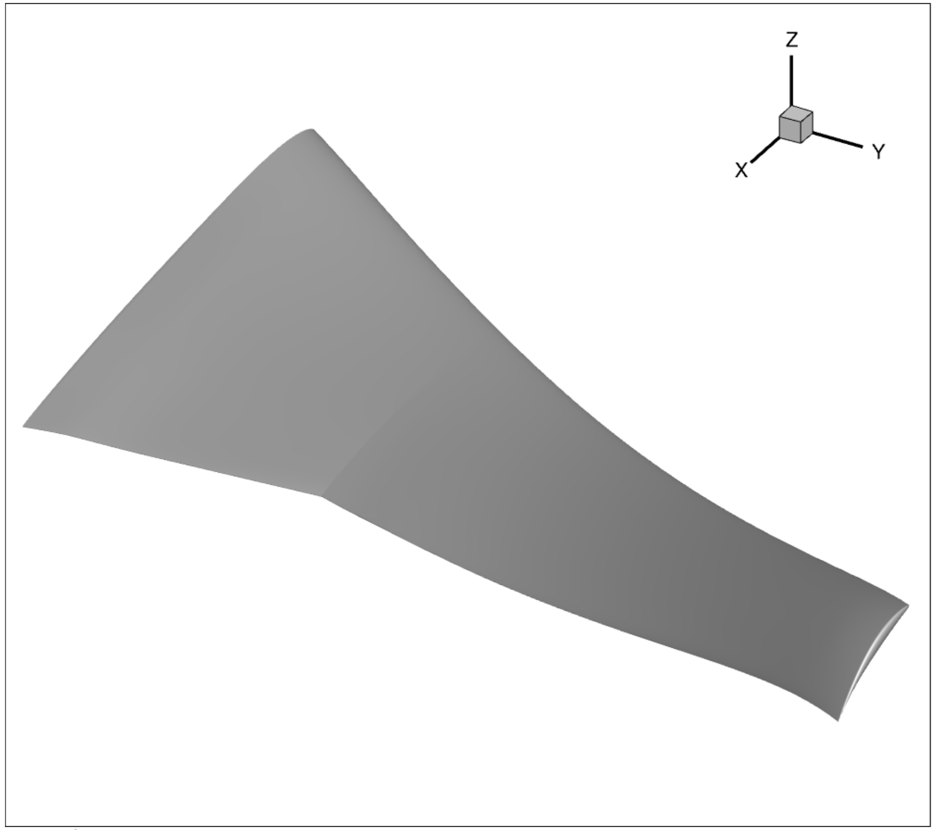
MG used for MeshDeformation

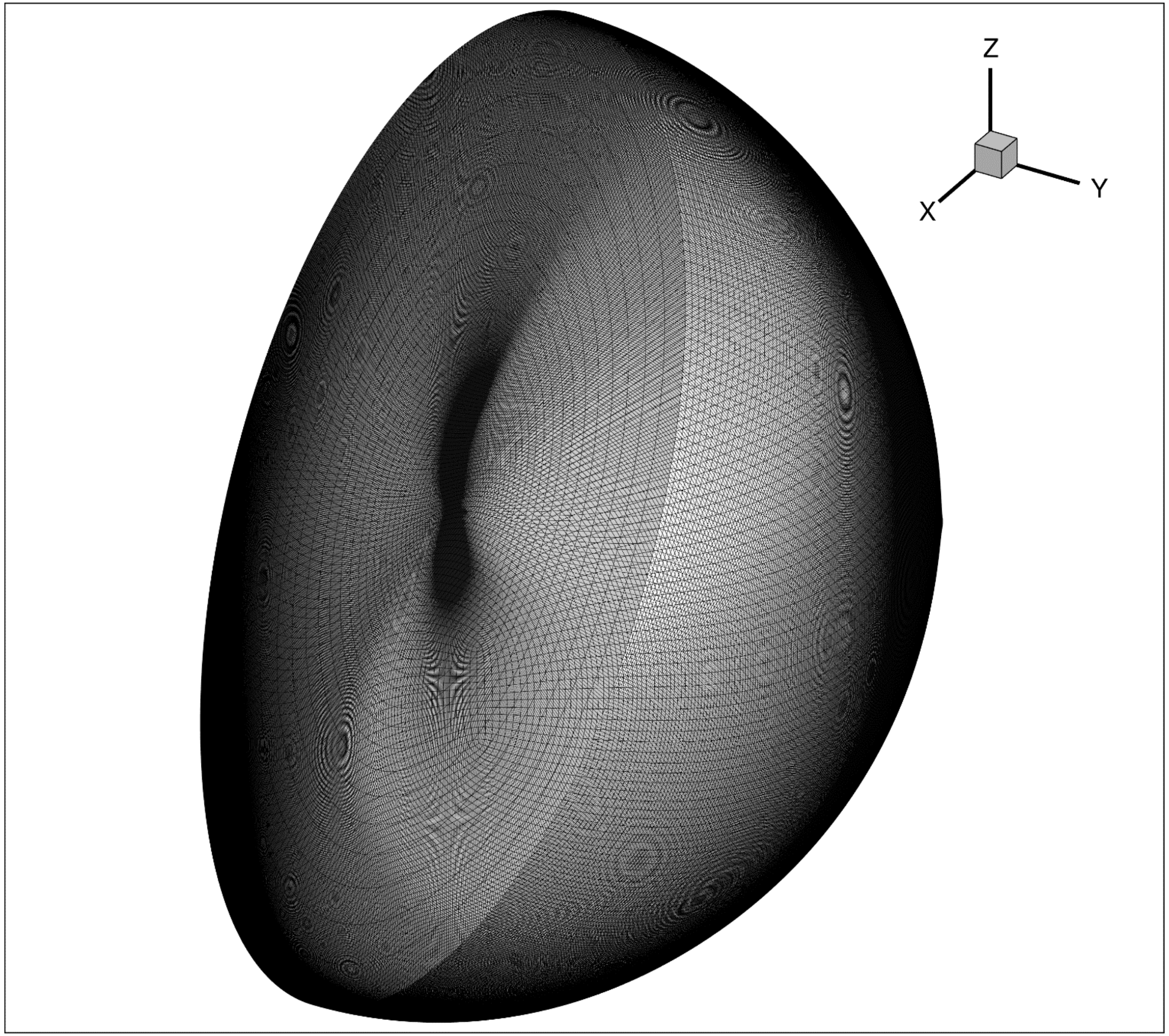
- h-coarsening with isometric uniform agglomeration (no semi-coarsening, no redistribution in level-transfers)
- Multi-color GaussSeidel with two iterations as smoother on all levels
- When LinesInversion is used, only on the finest level, all other levels point-implicit BlockInversion
- MG is preconditioner for GMRes and does just one single V(0,1)-“cycle“ each time it is called

Numerical Results (1)

3D CRM Wing

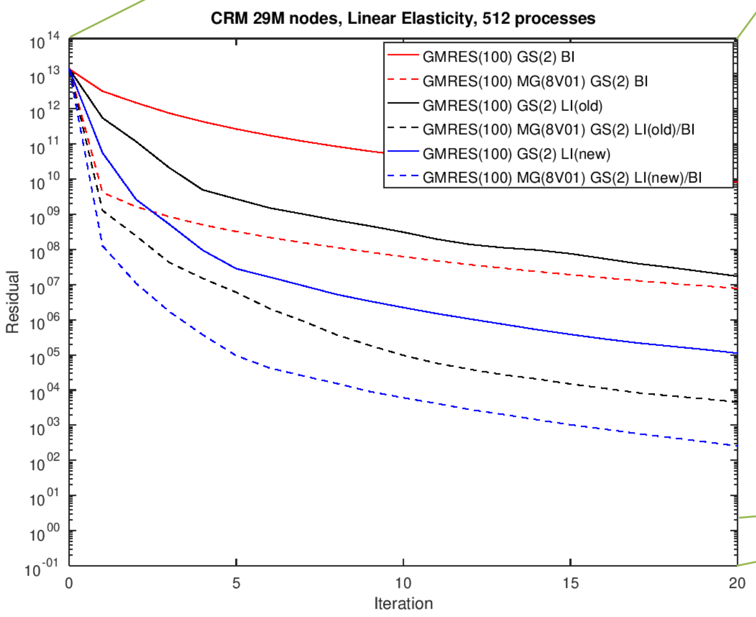
- Mesh of Hexa cells only
    - 28.8M cells, 29M nodes
- Aspect ratio 4000:1 in boundary layer

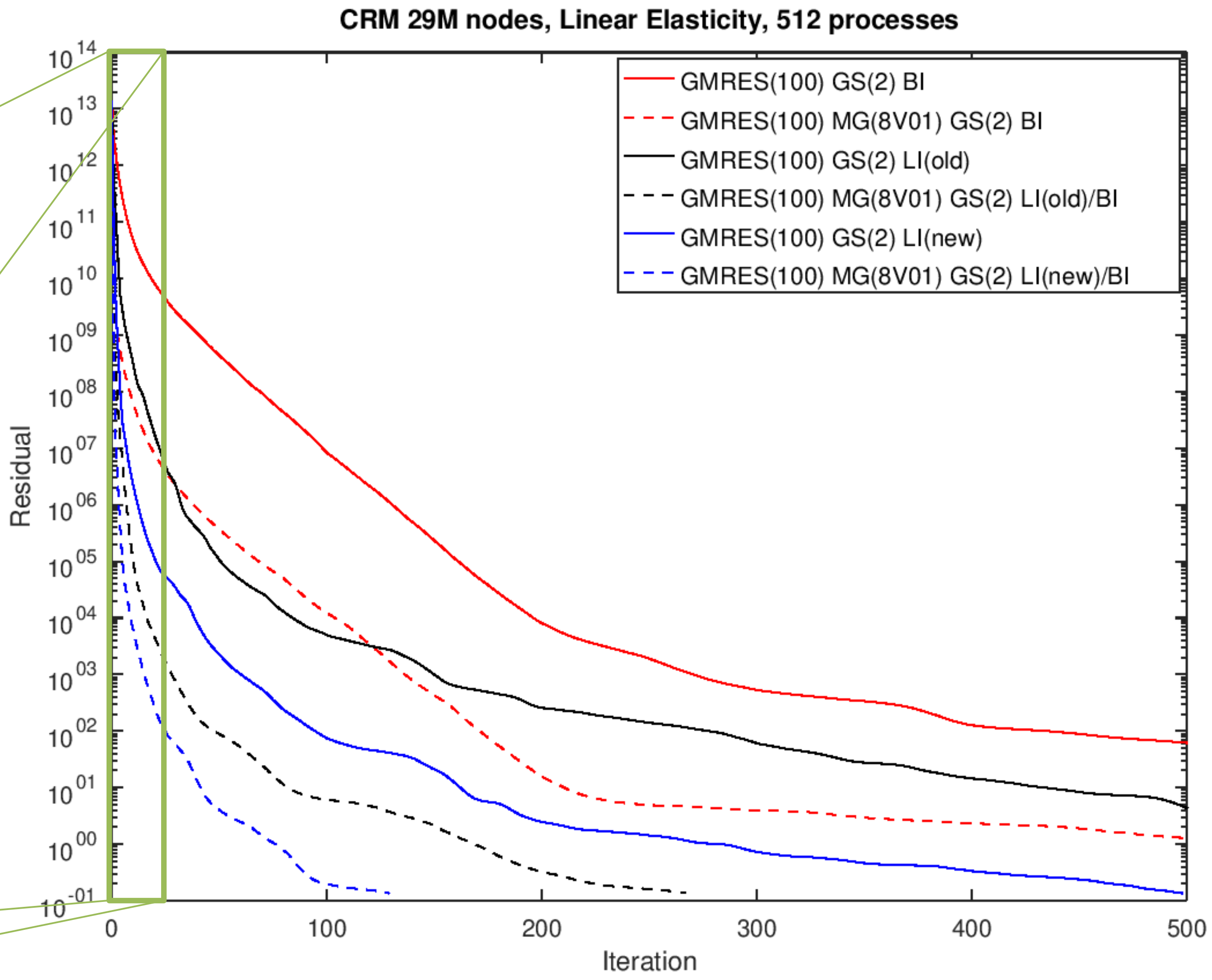




Lines-implicit and Multigrid needs fewer iterations

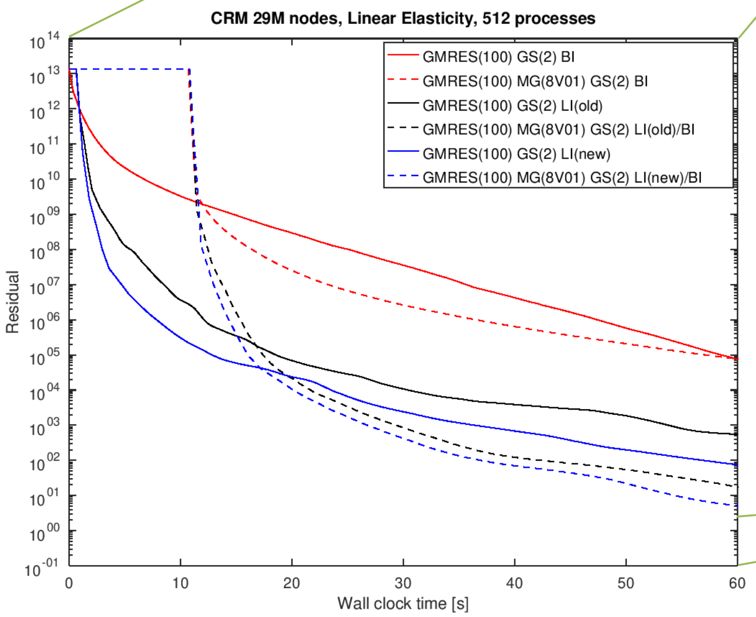
- All methods converge
- LI(old) compared to BI improves, LI(new) even more
- Compared to the same algorithm on a single (finest) grid, multigrid needs fewer

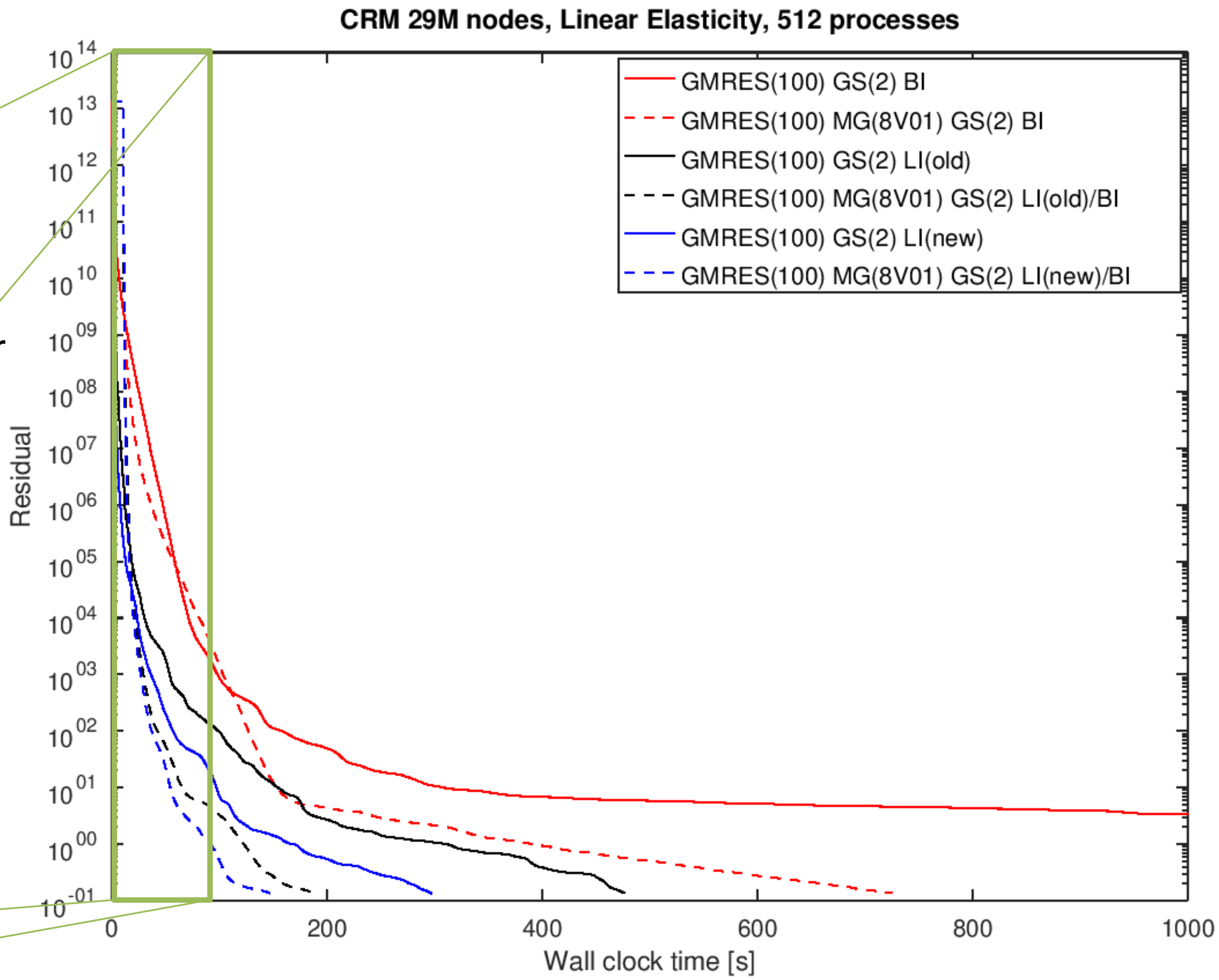




Multigrid is faster

- Effort for LI is similar to BI, Lines detection not expensive
- Even though the additional setup for the coarse levels takes some time before the actual start of the first iteration, the stronger MG algorithm often pays off quite soon

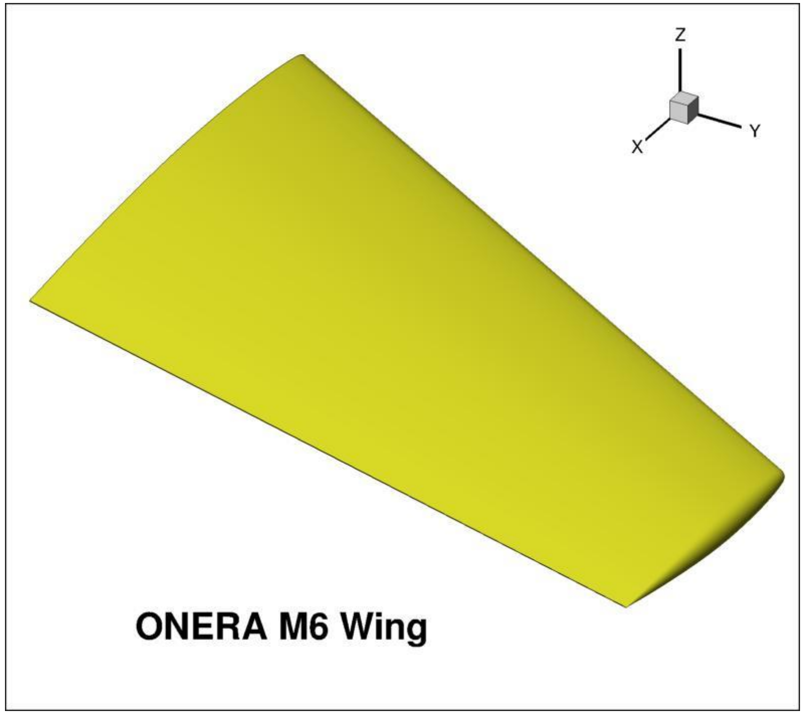


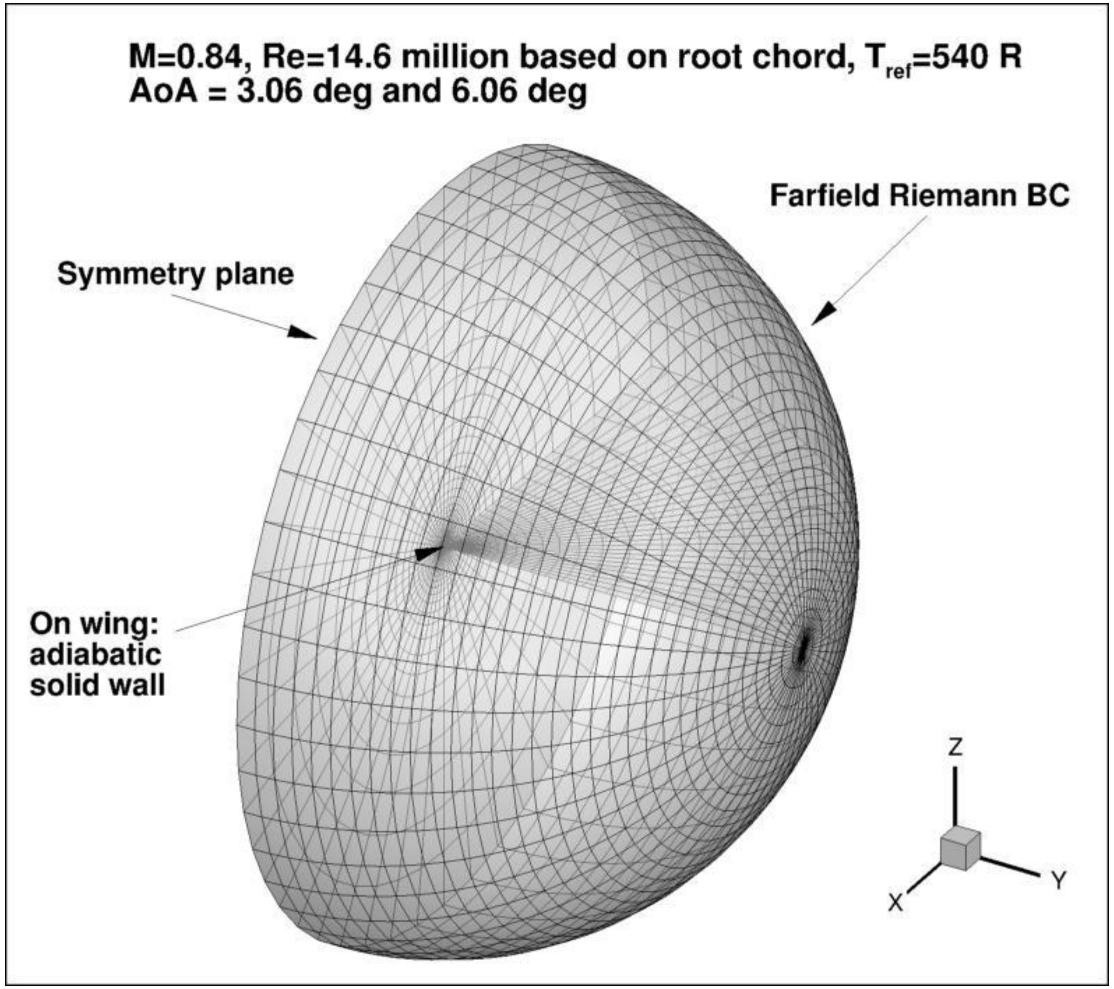


Numerical Results (2)

3D ONERA M6 Wing

- Mesh of Hexa and Prism cells:
  - 69.2M cells, 60.8M nodes
- From NASA's Turbulence Modelling Resource

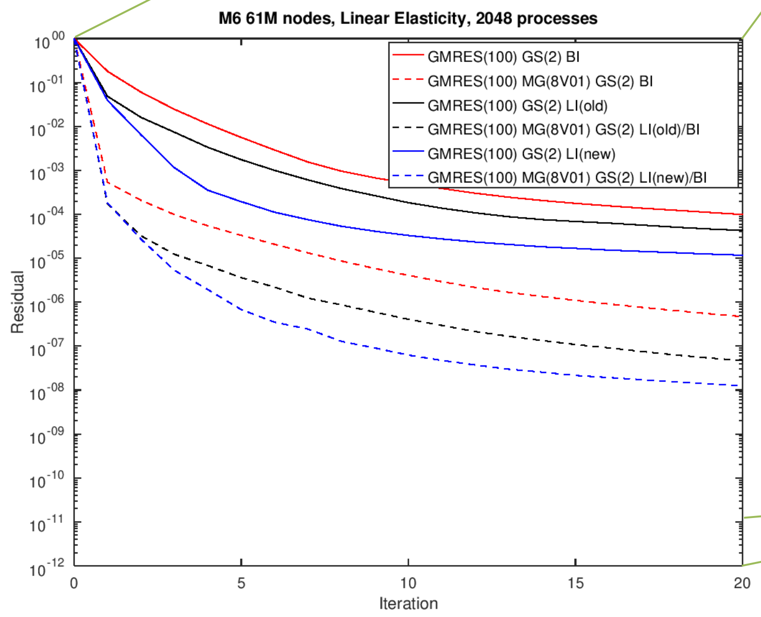


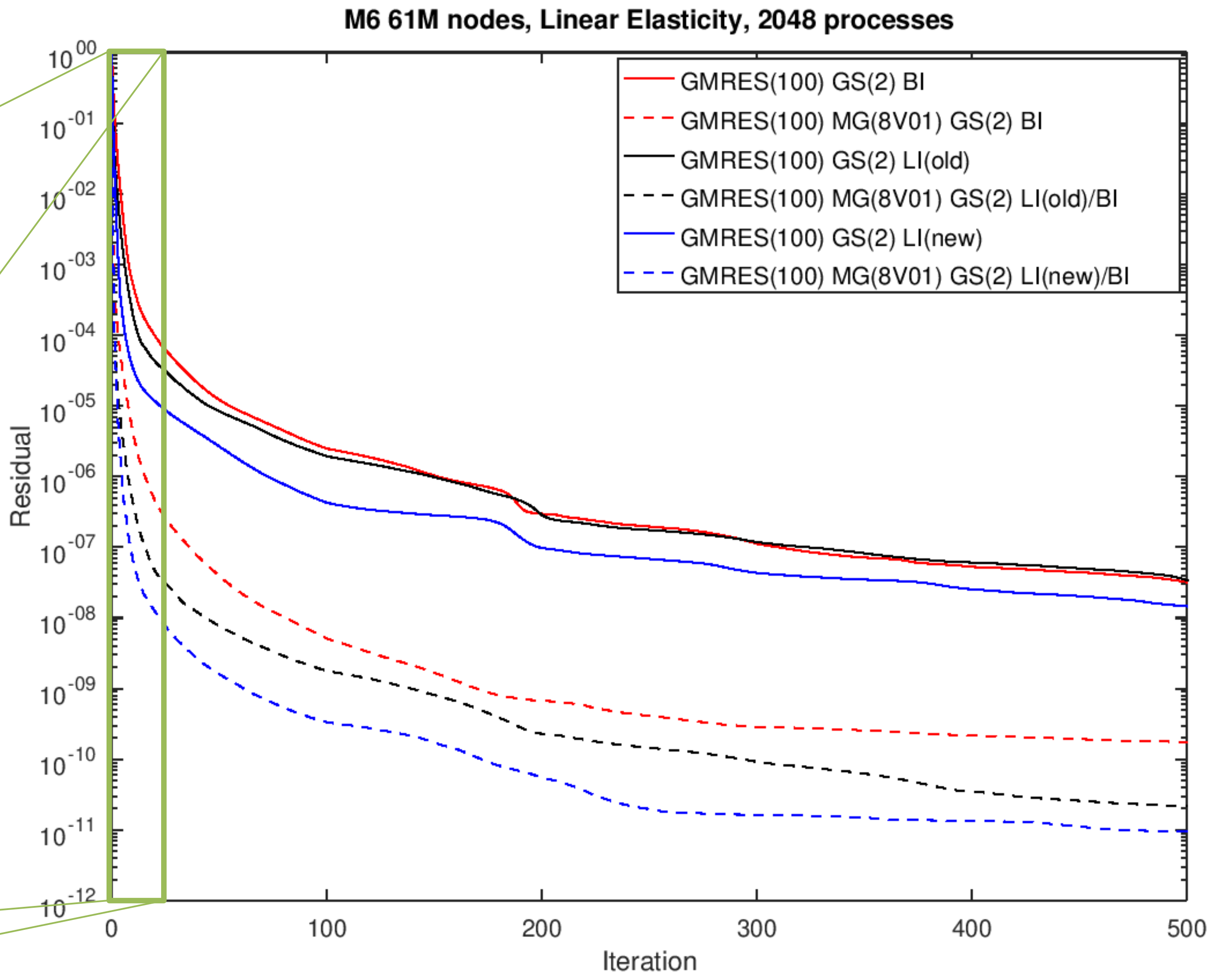


- Viscous boundary layer to compute pressure gradients at $Re = 14.6 \times 10^6$
- Aspect ratio 18000:1 in boundary layer

Lines-implicit and Multigrid needs fewer iterations

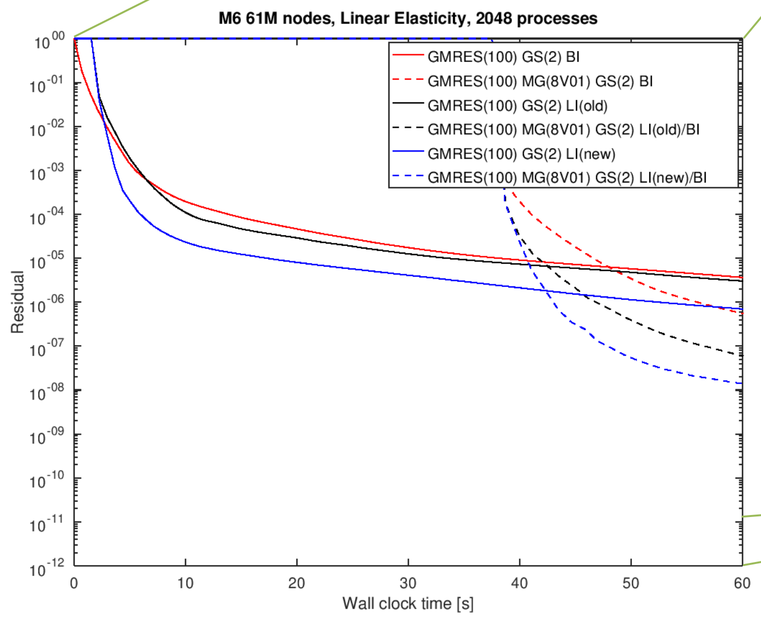
- LI(old) similar to BI, LI(new) improves
- Compared to the same algorithm on a single (finest) grid, multigrid needs significantly fewer iterations

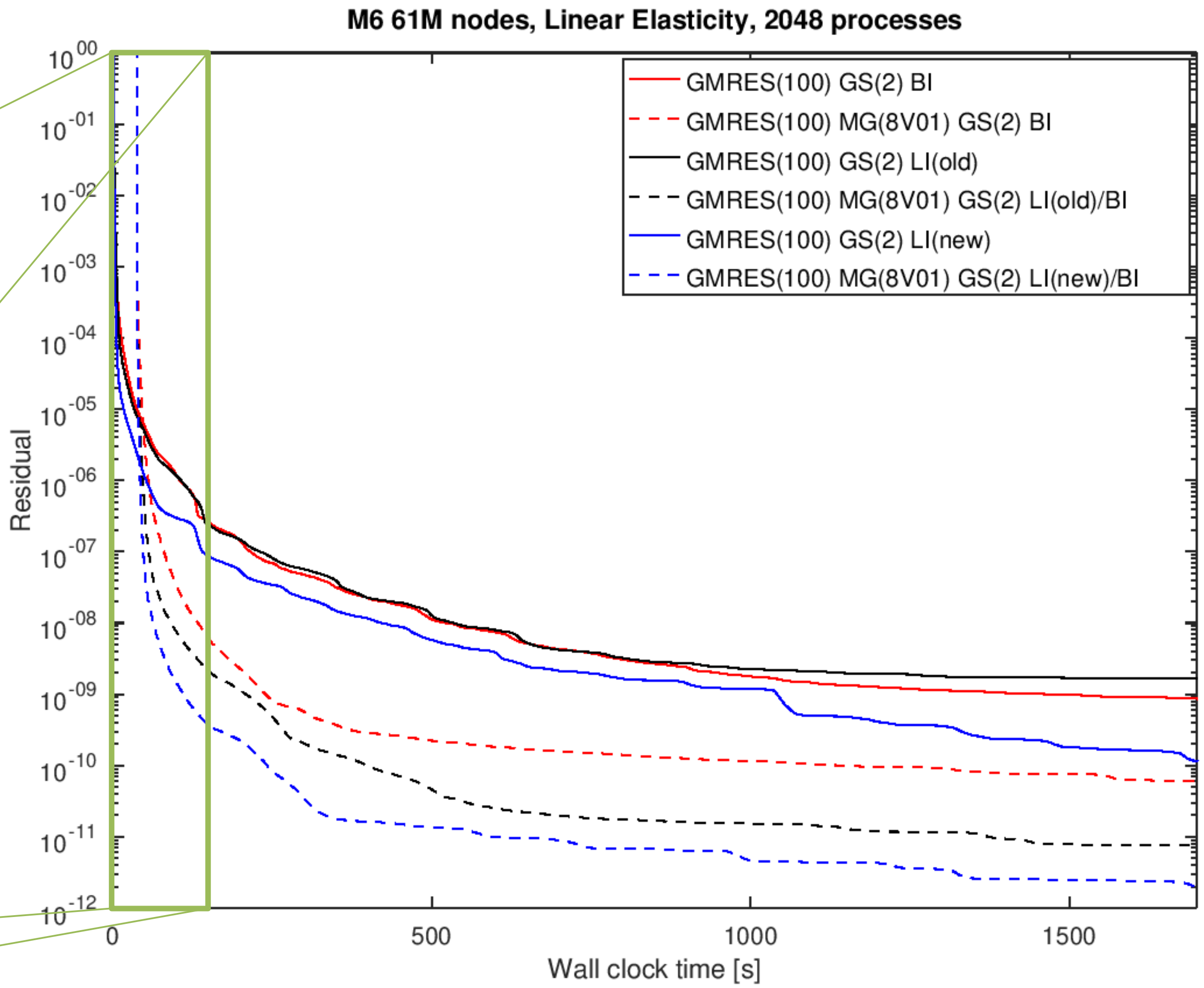




Multigrid is faster

- Effort for LI is similar to BI, Lines detection not expensive
- Multigrid setup is more expensive, but still pays off due to much increased convergence speed





Summary

Summary

- Line-Implicit Methods are applicable for Mesh Deformation
- New lines detection method was developed and verified with good results
- Lines-implicit method is important for fast convergence in anisotropic meshes
- Multigrid method boosts performance significantly, especially for large meshes

Thank You!

Questions?