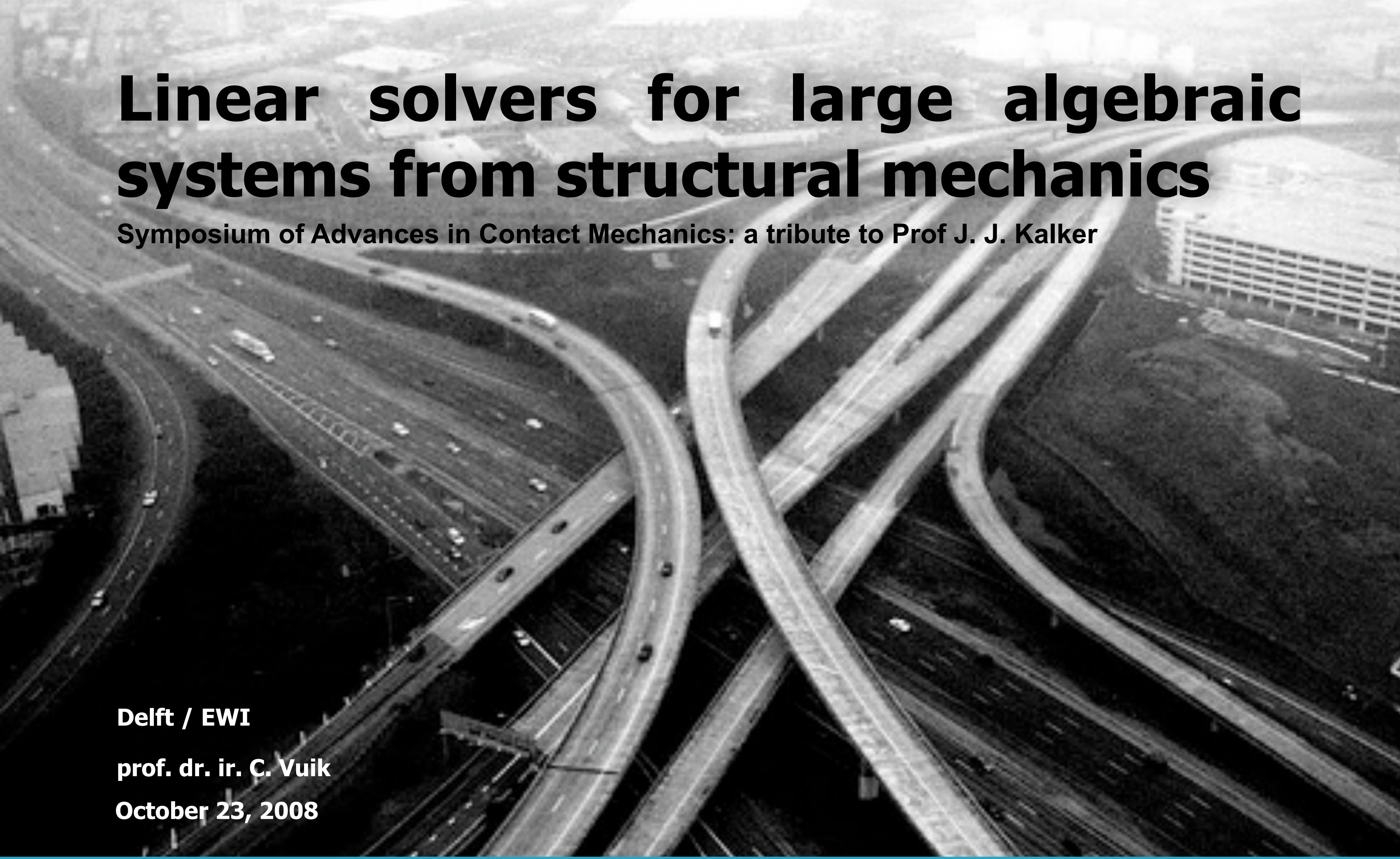


Linear solvers for large algebraic systems from structural mechanics

Symposium of Advances in Contact Mechanics: a tribute to Prof J. J. Kalker

Delft / EWI

prof. dr. ir. C. Vuik

October 23, 2008

• **Department of Applied Mathematical Analysis**

TU Delft

Delft University of Technology

Outline

- Structural mechanics
  - Computational framework
  - Finite element discretization
- Numerical linear algebra
  - Overview numerical methods for linear systems
  - Parallel direct solver
  - Combining methods to create new solver
- Test case
- Discussion

How to compute deformation?
Force balance

- Forces should be in balance,

$$\int_{\Omega} div(\sigma) + \mathbf{f} - \rho \mathbf{g} d\Omega = 0$$

internal force σ, external force $\mathbf{f}$,
gravitation $\mathbf{g}$

- Residual when out of balance,

$$\mathbf{r} = div(\sigma) + \mathbf{f} - \rho \mathbf{g}$$

How to compute deformation?
Virtual work

- Define virtual work,

$$\delta W = \int_v \mathbf{r} \cdot \delta \mathbf{u} dv$$

- Equilibrium

$$\delta W\left(\mathbf{X}\right) = \delta W_{int}\left(\mathbf{X}\right) - \delta W_{ext}\left(\mathbf{X}\right) = 0$$

How to compute deformation?
Linearized virtual work

- First order Taylor,

$$\delta W\left(\mathbf{X}_0\right) + D_{\Delta u}\left[\delta W\left(\mathbf{X}_0\right)\right] = 0$$

- Linearized virtual work

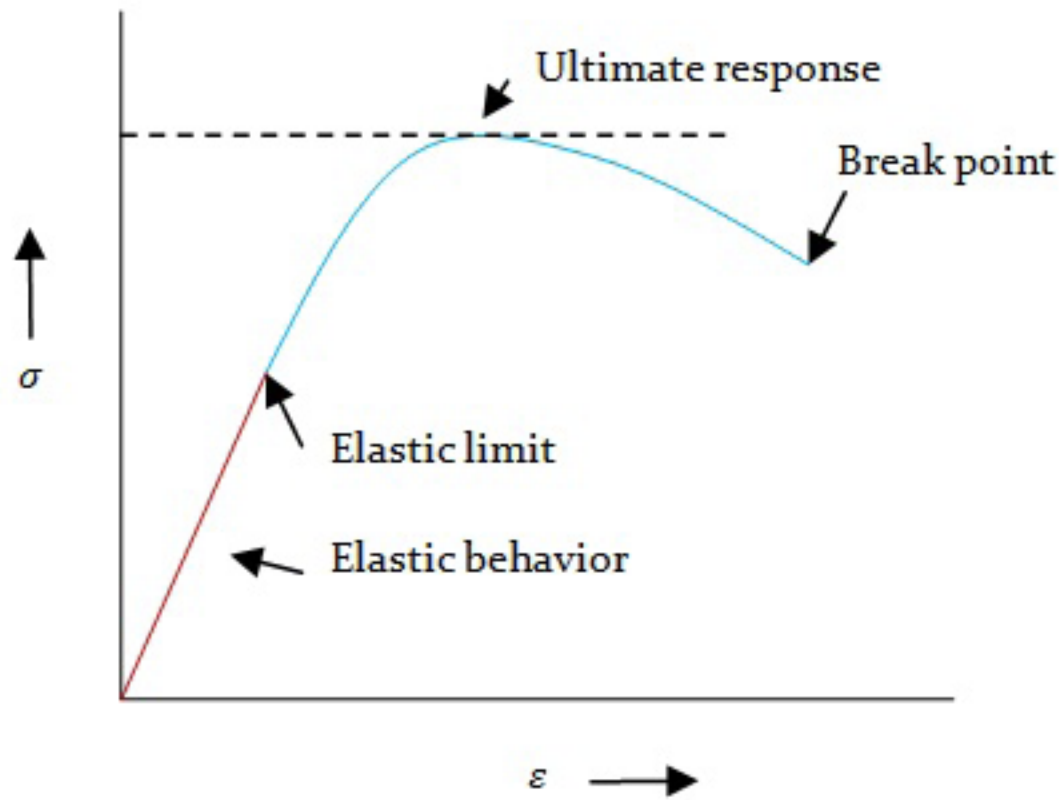
$$\int_V \left(\nabla_0 \Delta \mathbf{u} \cdot \mathbf{S}\right) : \nabla_0 \delta \mathbf{v} dV + \int_V \left(\nabla_0 \Delta \mathbf{u} : \mathbf{F} \cdot \mathbb{C} \cdot \mathbf{F}^T\right) : \nabla_0 \delta \mathbf{v} dV = \delta \mathbf{v} \cdot \mathbf{f}_{ext} - \int_V \mathbf{P} : \nabla_0 \delta \mathbf{v} dV$$

How to compute deformation?
Material response

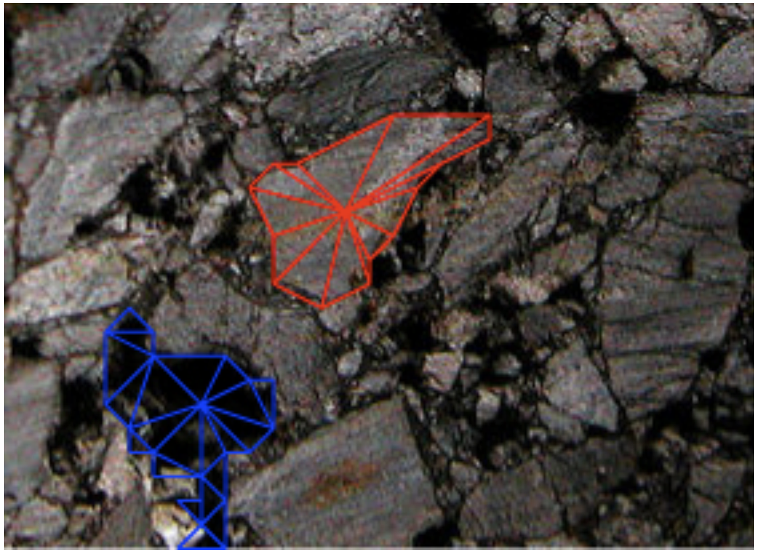
- <u>Three</u> material properties,
  - (Hyper) elasticity
  - Plasticity (permanent deformation)
  - Viscosity (permanent deformation)

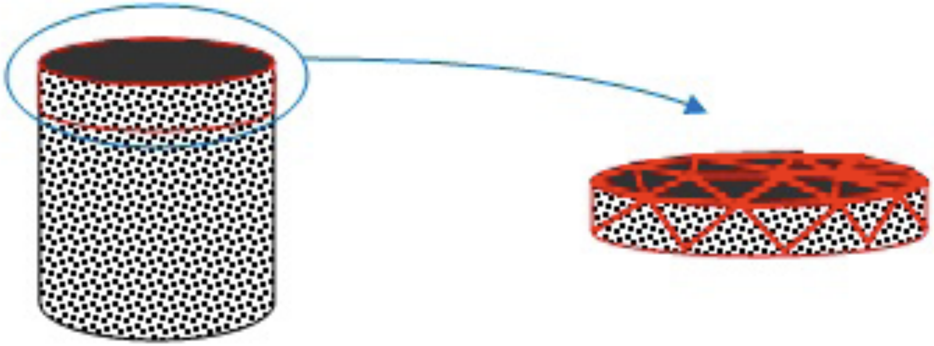
TU Delft

How to compute deformation?
Non-linear material response



Finite element discretization
FE mesh





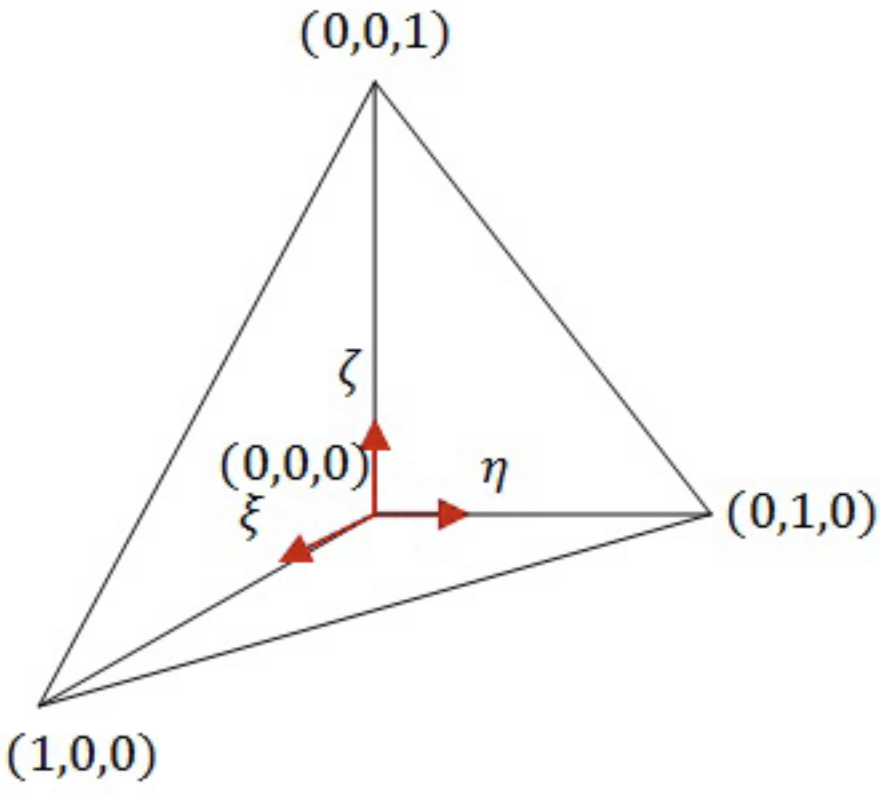
Finite element discretization
Tetrahedral elements

- Introduce local coordinate system,

$$(x, y, z) \rightarrow (\xi, \eta, \zeta)$$

- Transformation between local and global coordinates,

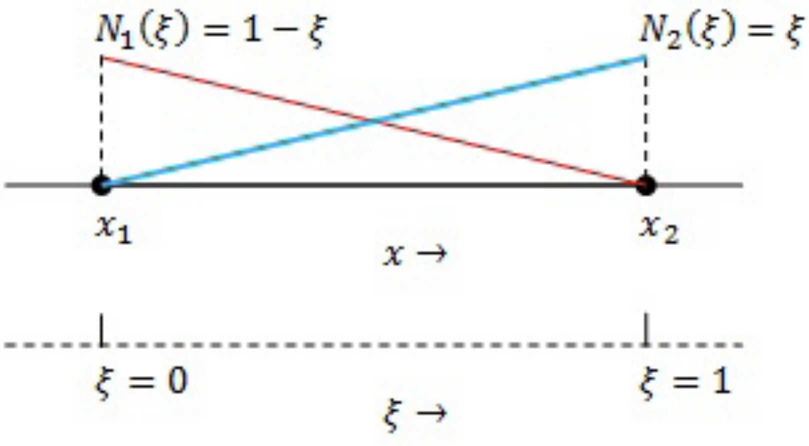
$$\frac{d}{d\xi} = J\frac{d}{dx}$$



Finite element discretization
Shape functions

- 1D Example with linear shape functions,

$$x = N_1(\xi)\, x_1 + N_2(\xi)\, x_2$$



- For 3D case,

$$\mathbf{x} = \sum_{i=1}^{4} \mathbf{N}_i(\xi, \eta, \zeta) \cdot \mathbf{x}_i$$

- For stability and accuracy
  higher order shape functions necessary

Finite element discretization
Numerical integration

- Use Gauss point(s) for numerical integration,

$$\int_{\Omega} f \, d\Omega \simeq f\left(\xi_g, \eta_g, \zeta_g\right) |J| \int_{\Omega} d\Omega$$

Finite element discretization
Stiffness matrix

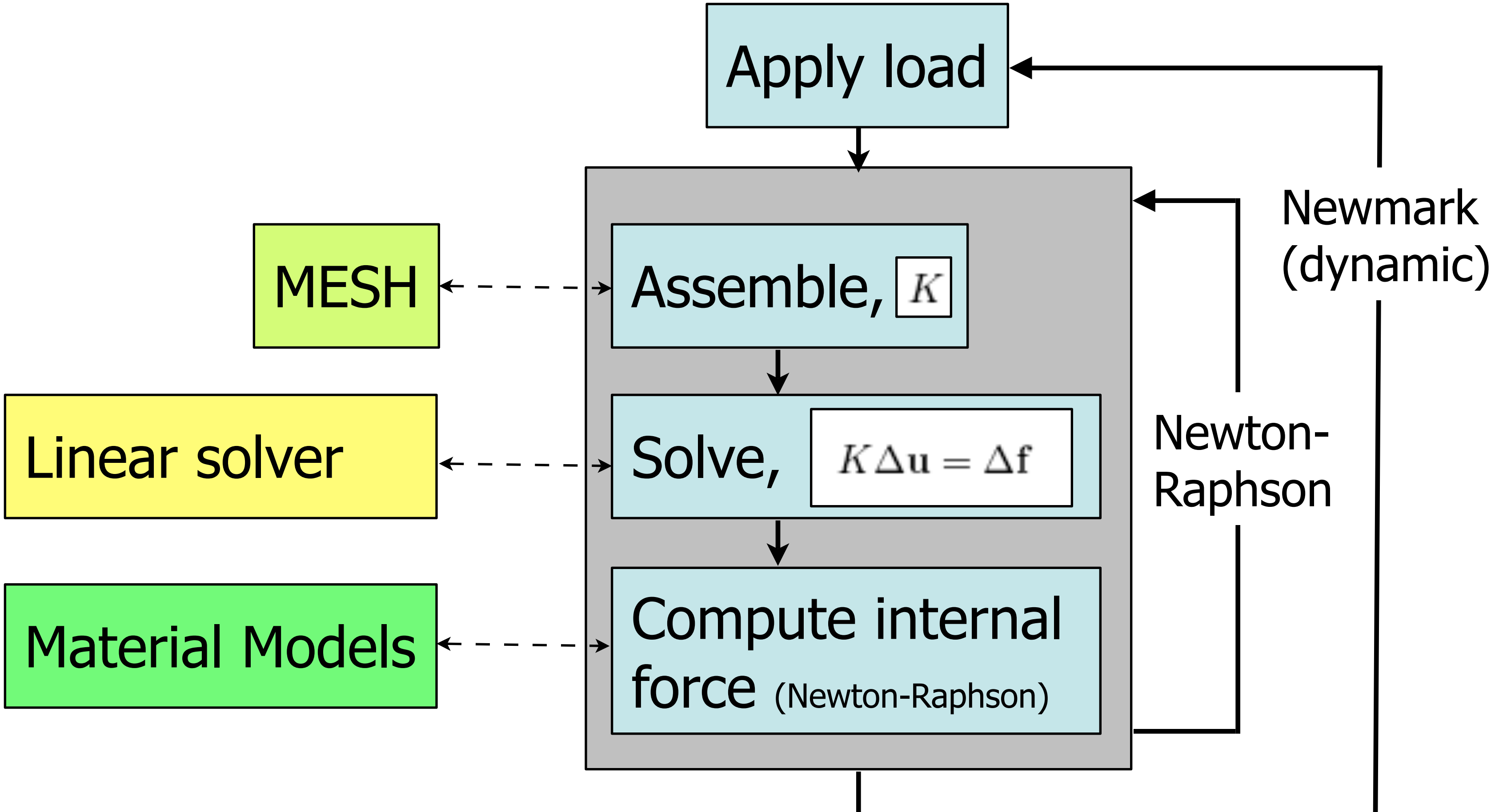
- Discretized, linearized virtual work,

$$K\Delta\mathbf{u} = \mathbf{f}_{ext} - \mathbf{f}_{int}$$

Finite element discretization
Stiffness matrix

- Properties of K,
  - Symmetric
  - Positive definite
  - Sparse
  - No specific pattern of non-zero matrix entries
  - Large differences in entry values due to material properties
  - Changes due to non-linear material properties

How to compute deformation?
Balancing of forces algorithm

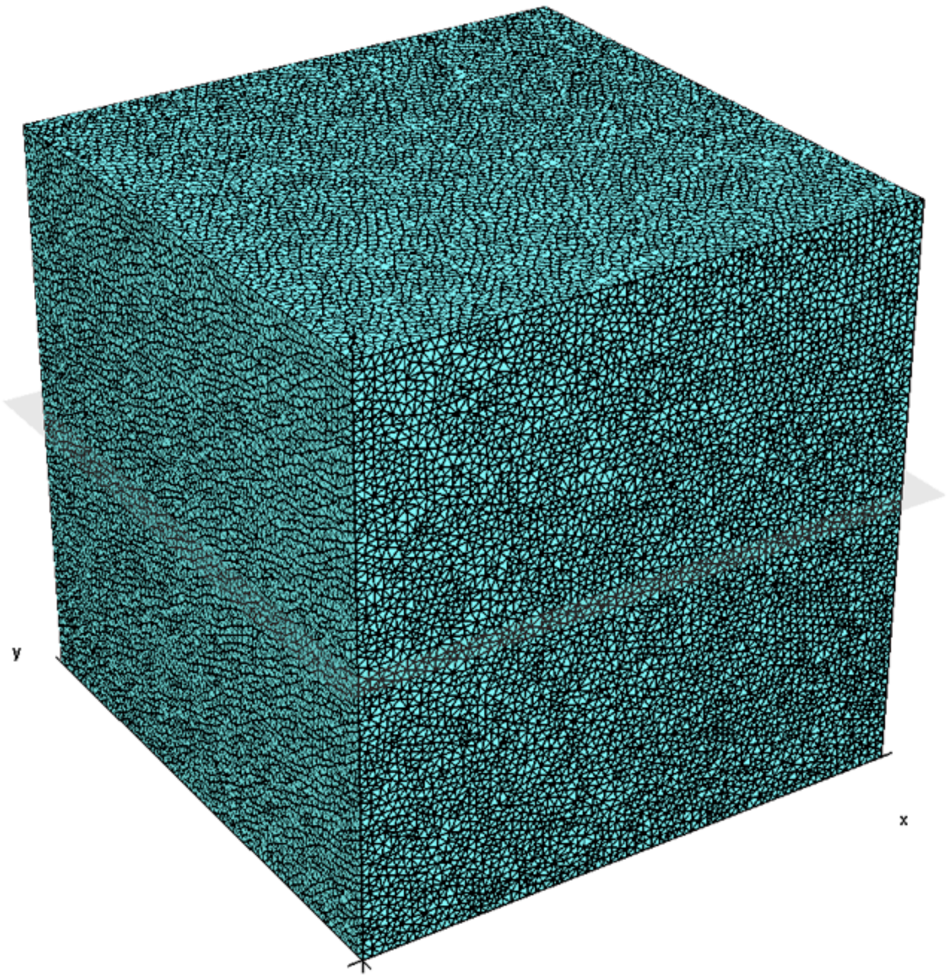


Problems with algorithm
Scale

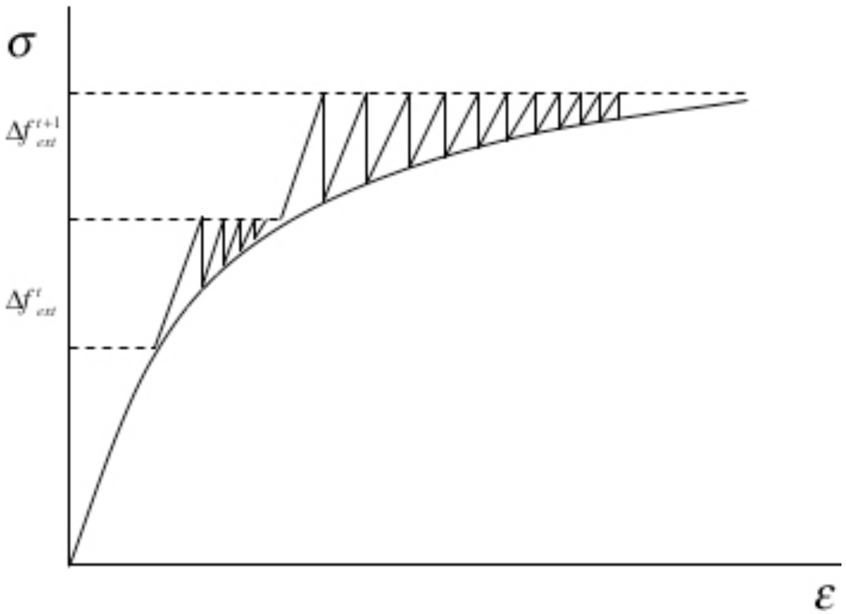
Number of elements:
1.890.057

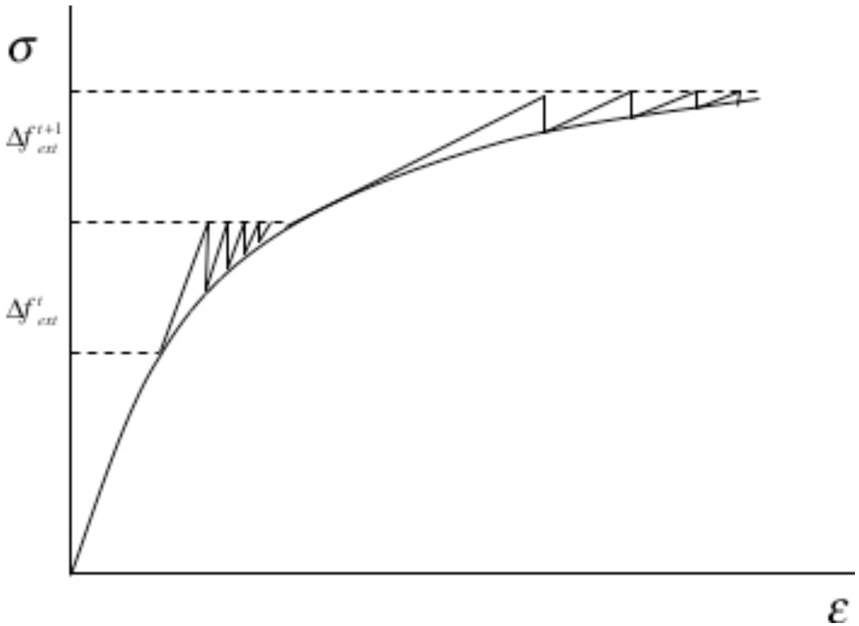
Number of nodes:
307.735

Number of non zero elements in stiffnessmatrix:
21.296.523



Problems with algorithm
Accuracy and approximation





TU Delft

Numerical methods
Overview

- (Parallel) Direct solver
- Iterative solvers,
  - Preconditioning
  - CG method
  - Deflation, Domain Decomposition of Multigrid?

Parallel direct solver
Definition

- Direct solver,

$$x = A^{-1}b$$

- Matrix cannot be singular or ill conditioned, this leads to inaccurate solutions

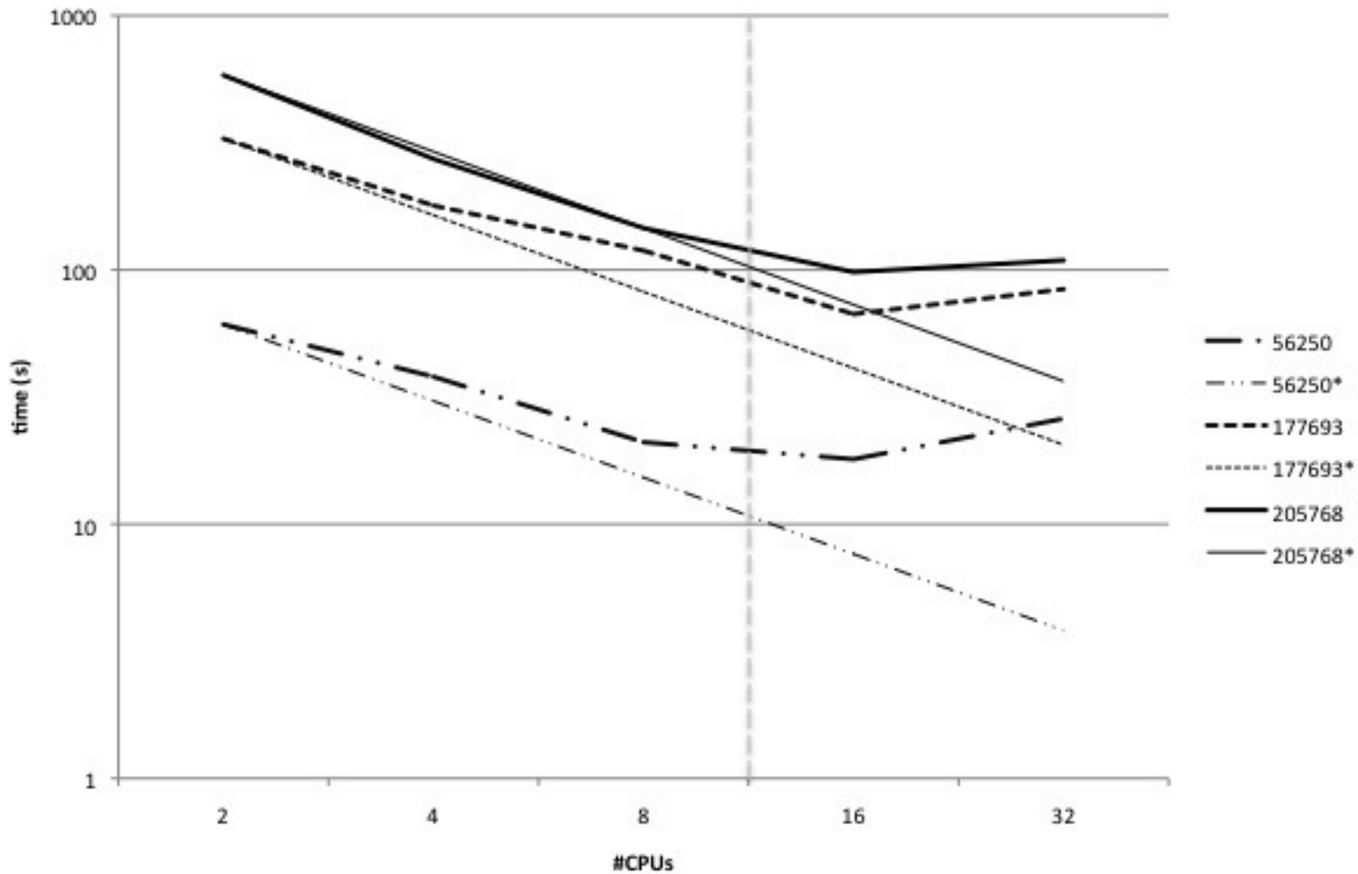
- Large (3D) models yield large linear systems, serial direct solvers lack CPU power and memory

Parallel direct solver
MUMPS

- Solution? Go parallel! Spread work over computing nodes. Adding more nodes implies more CPU power and memory.

- MUMPS project: public domain package and developed by CERFACS, **graal.ens-lyon.fr/MUMPS/**

Parallel direct solver
MUMPS



TU Delft

Iterative methods
Basics

- Solve,

$$Ax = b$$

- Use sequence of approximations of solution x,

$$x_0, x_1, x_2, ..., x_k$$

where,

$$x_{k+1} = x_k + M^{-1}(b - Ax_k)$$

- Choice of M defines iterative method

TU Delft

Iterative methods
Available methods

- Splitting based methods ($M = N - A$),
  - Jacobi
  - Gauss-Seidel
  - SSOR
- Krylov subspace methods
  - CG
  - GMRES
- Multigrid

Iterative methods
Preconditioning

- Condition number,

$$\kappa_p(A) = ||A||_p \left|\left|A^{-1}\right|\right|_p .$$

- For (symmetric) SPD matrices,

$$\kappa_p(A) = \frac{|\lambda_{max}|}{|\lambda_{min}|}$$

- Improve condition of matrix,

$$M^{-1}Ax = M^{-1}b$$

Iterative methods
Preconditioning

- Preconditioner is approximation of original matrix
- Matrix M can be any constant linear solver
- Many choices,
  - Incomplete LU or Cholesky decomposition,
  - Basic iterative methods (GS, Jacobi)
  - Multigrid
  - Domain decomposition
  - Deflation

Iterative methods
Conjugate gradient (CG)

- Krylov subspace,

$$x_0 + span\left\{M^{-1}r_0, M^{-1}A\left(M^{-1}r_0\right), ..., \left(M^{-1}A\right)^{i-1}\left(M^{-1}r_0\right)\right\}$$

- Good performance for well conditioned SPD matrices
- Slow converging components corresponds to smallest eigenvalues of A
- Preconditioner necessary for ill conditioned systems

Iterative methods
Multigrid

- Idea: Approximation of (smooth) error of the solution on coarser grids. Back propagate error to fine grid:

$$Ax = b \rightarrow \Delta x = x - \tilde{x}$$

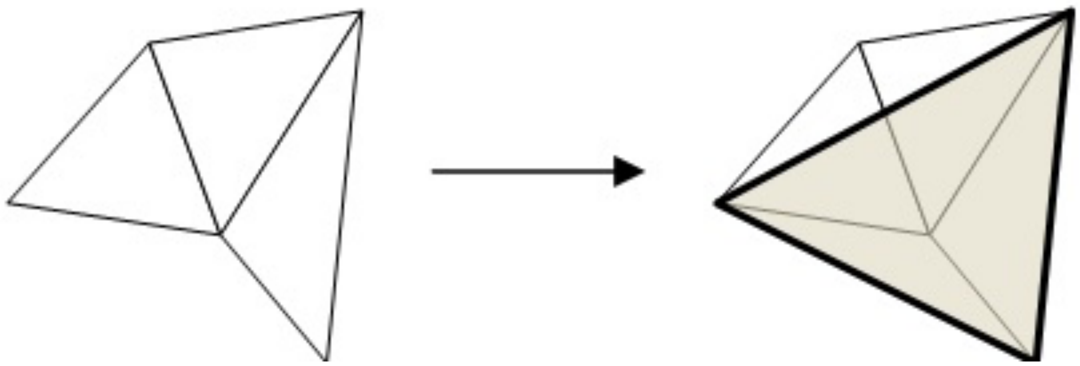
$$r_h = A_h \Delta x_h \rightarrow I_h^H \rightarrow r_H = A_H \Delta x_H$$

$$\tilde{x}_h^{k+1} = \tilde{x}_h^k + I_H^h \Delta x_H$$

- Benefit: Reduction of size the system that has to be solved with direct solver.

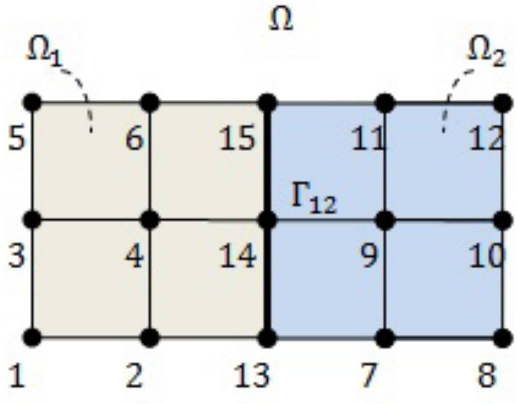
Iterative methods
Multigrid

- How to choose grid operators I_h^H, I_H^h?
- How to choose coarse grid cells on unstructured grids?



Iterative solvers
Domain decomposition

- Divide large problem into subdomains, divide work load and easy parallelizable.



- Rewrite original system,

$$\Omega_i, \forall i \in \{1, 2, ..., s\}$$

$$A \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} \text{ with } A = \begin{pmatrix} B & E \\ F & C \end{pmatrix}$$

where $\mathbf{y}, \mathbf{g}$ correspond to interface nodes

TU Delft

Iterative solvers
Domain decomposition

- Schur complement S,

$$\begin{aligned}\left(C - FB^{-1}E\right)\mathbf{y} &= \mathbf{g} - FB^{-1}\mathbf{f} \\ S\mathbf{y} &= \mathbf{g}'\end{aligned}$$

- Solve $\mathbf{y}$ and obtain $\mathbf{x}$ from,

$$\mathbf{x} = B^{-1}\left(\mathbf{f} - E\mathbf{y}\right)$$

Iterative solvers
Domain decomposition

- How to choose subdomains?

- How to solve on subdomains?

Iterative solvers
Deflation

- Filter out the eigenvalues that belong to the slow converging components of for e.g. the CG method
- Deflation components,

$$
\begin{aligned}
Z &\in \mathbb{R}^{n\times k}, \quad k < n-d, \quad \text{deflation subspace matrix}\\
E &= Z^T A Z \in \mathbb{R}^{k\times k}, \quad \text{inversion Galerkin matrix or coarse matrix}\\
Q &= Z E^{-1} Z^T \in \mathbb{R}^{n\times n}, \quad \text{correction matrix}\\
P &= I - AQ \in \mathbb{R}^{n\times n}, \quad \text{deflation matrix}\\
PAx &= Pb
\end{aligned}
$$

d, number of zero eigenvalues

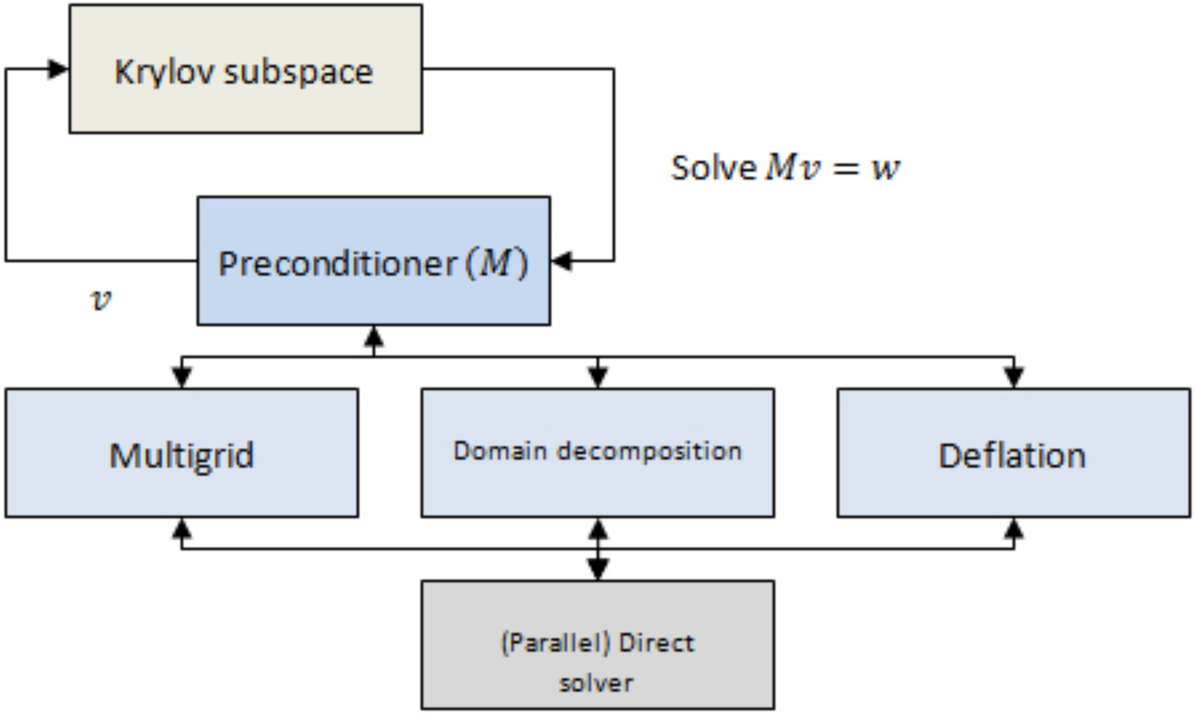
k, number of deflation vectors

TU Delft

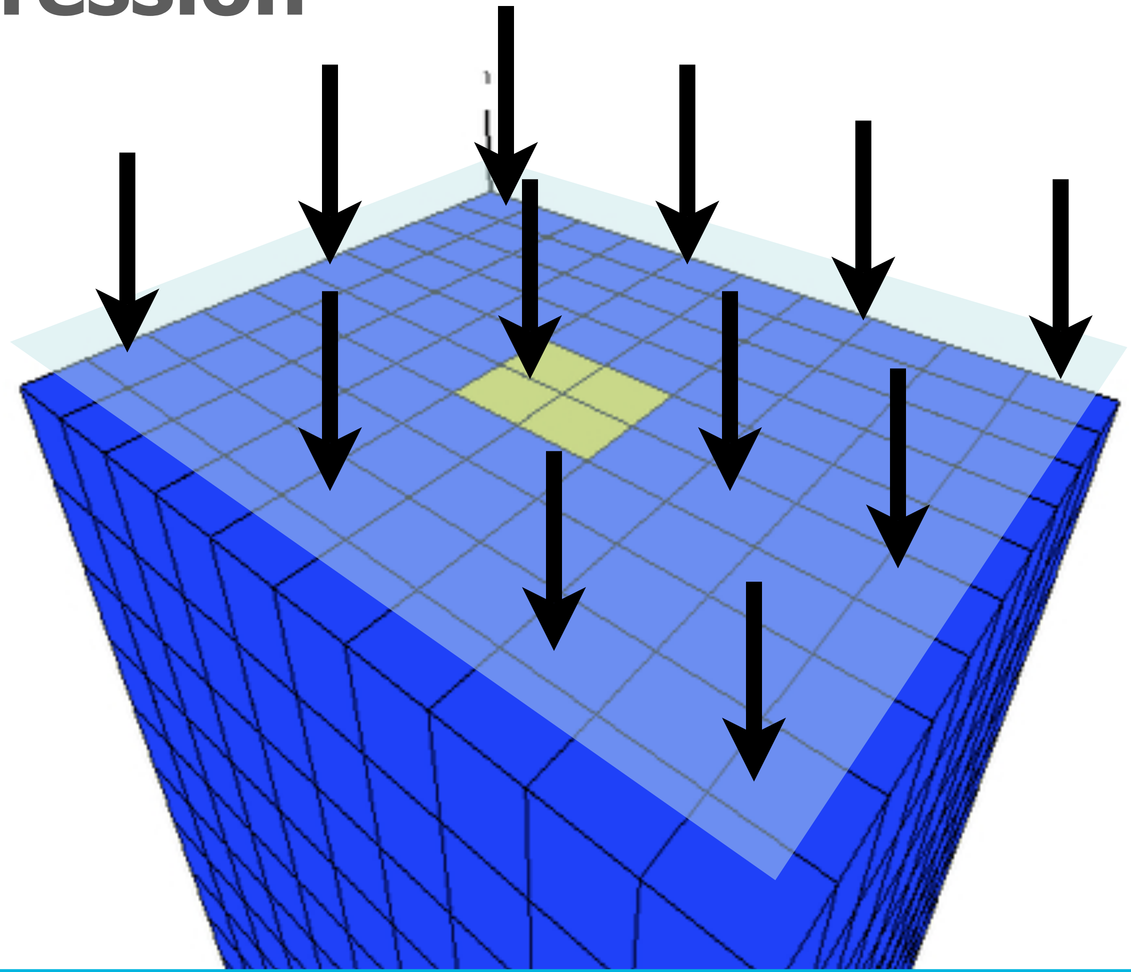
Iterative solvers
Deflation

- Preferably the deflation vectors are the eigenvectors corresponding to the smallest eigenvalues (think of condition number)
- Computation of deflation vectors is expensive, use approximations,
  - Physical : interface elements with high discontinuities
  - Analytical : use information CG, previous time steps, FE discretization etc.

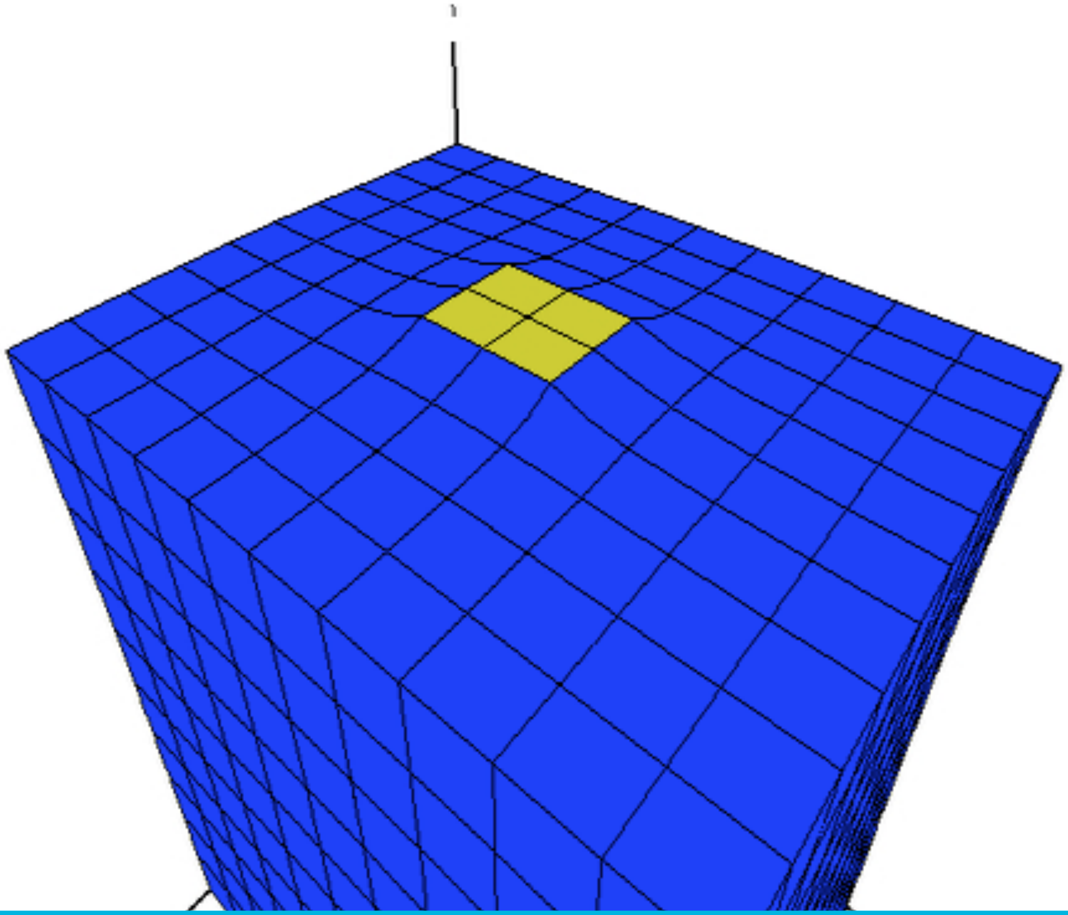
Hybrid solver
Combining numerical methods



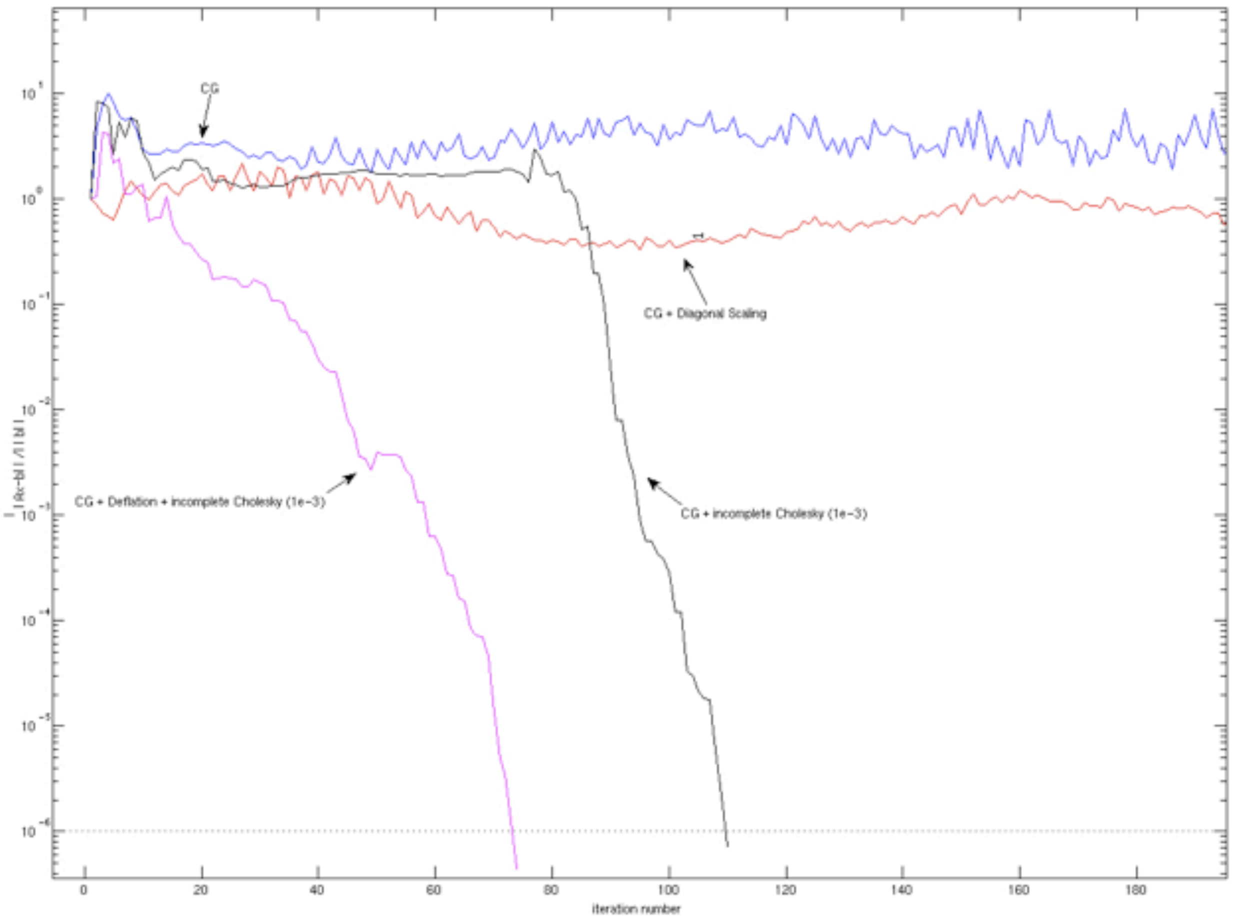
Test case
Compression



Test case
Compression



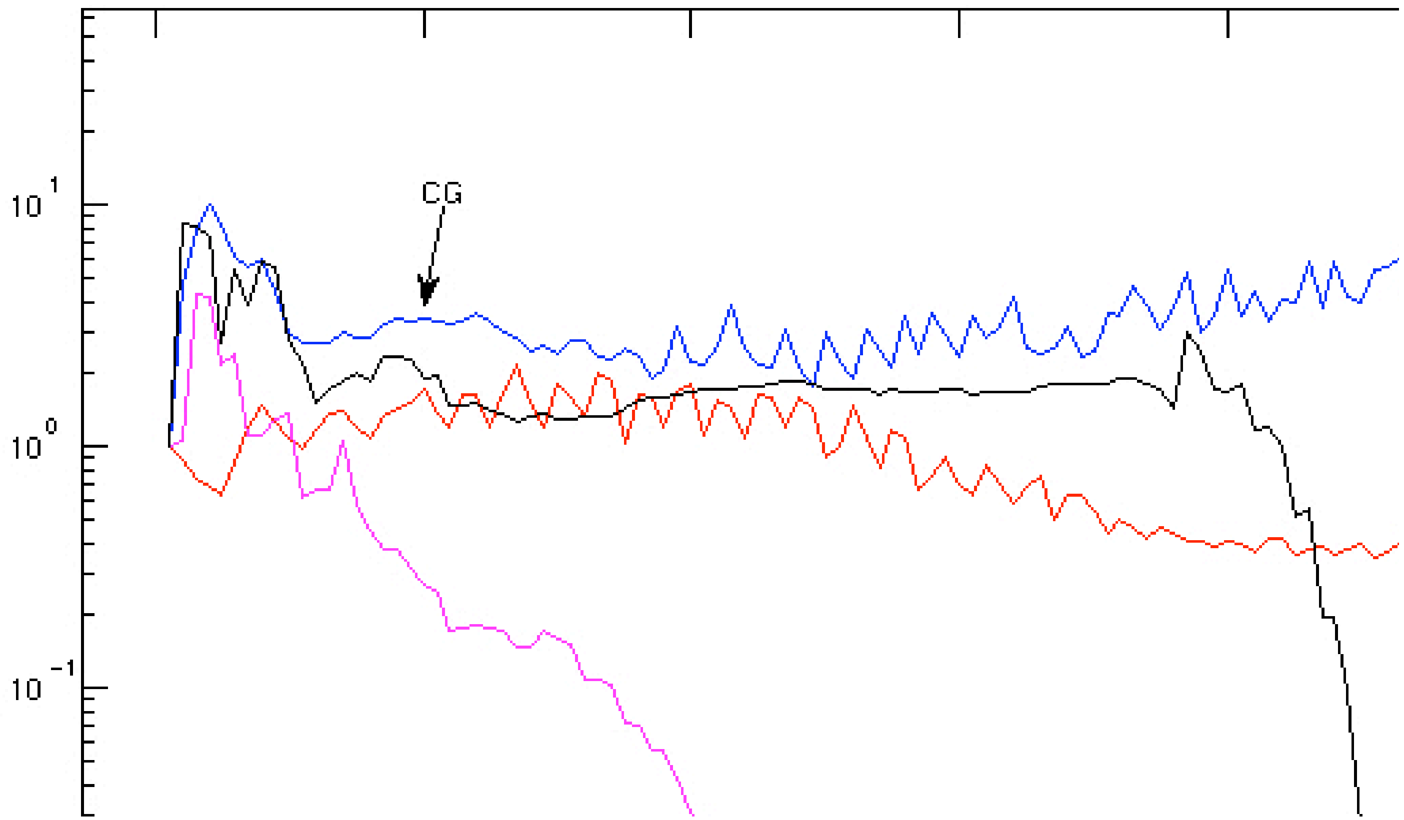
Test case
Deflation + CG + preconditioning



TU Delft

Test case

Deflation + CG + preconditioning



TU Delft

Future research
People

- Civil Engineering (group Scarpas),
  - A. Scarpas
  - C. Kasbergen
- Applied Mathematics (group Vuik),
  - C. Vuik
  - M.B van Gijzen
  - T.B Jönsthövel

Discussion

Q+A