STRUCTURAL MECHANICS

RUHR-UNIVERSITÄT BOCHUM

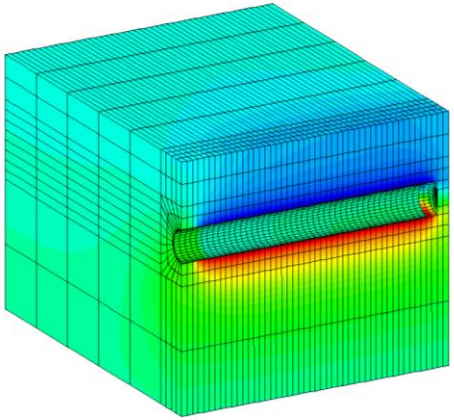
RUB

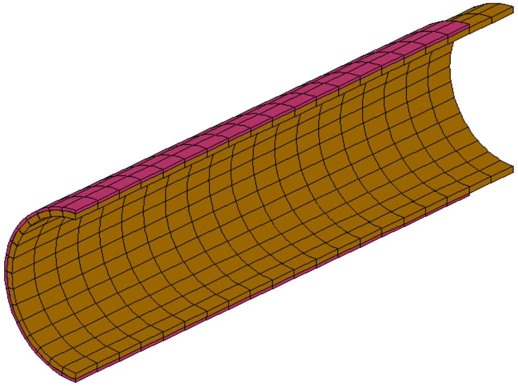
PARALLEL SIMULATION IN TUNNEL ENGINEERING

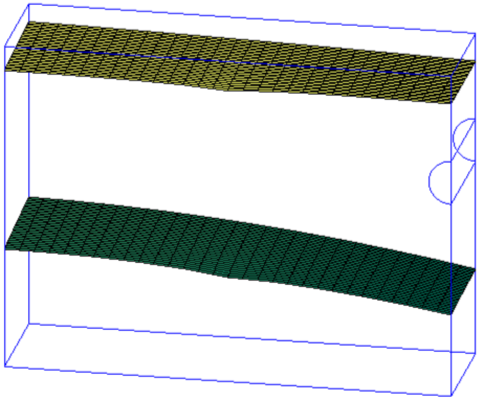
APPLICATION

Hoang-Giang Bui and Günther Meschke
Institute for Structural Mechanics
Ruhr University Bochum

PETSc User Meeting 2016
Vienna, Austria
28-30 June 2016







Outline

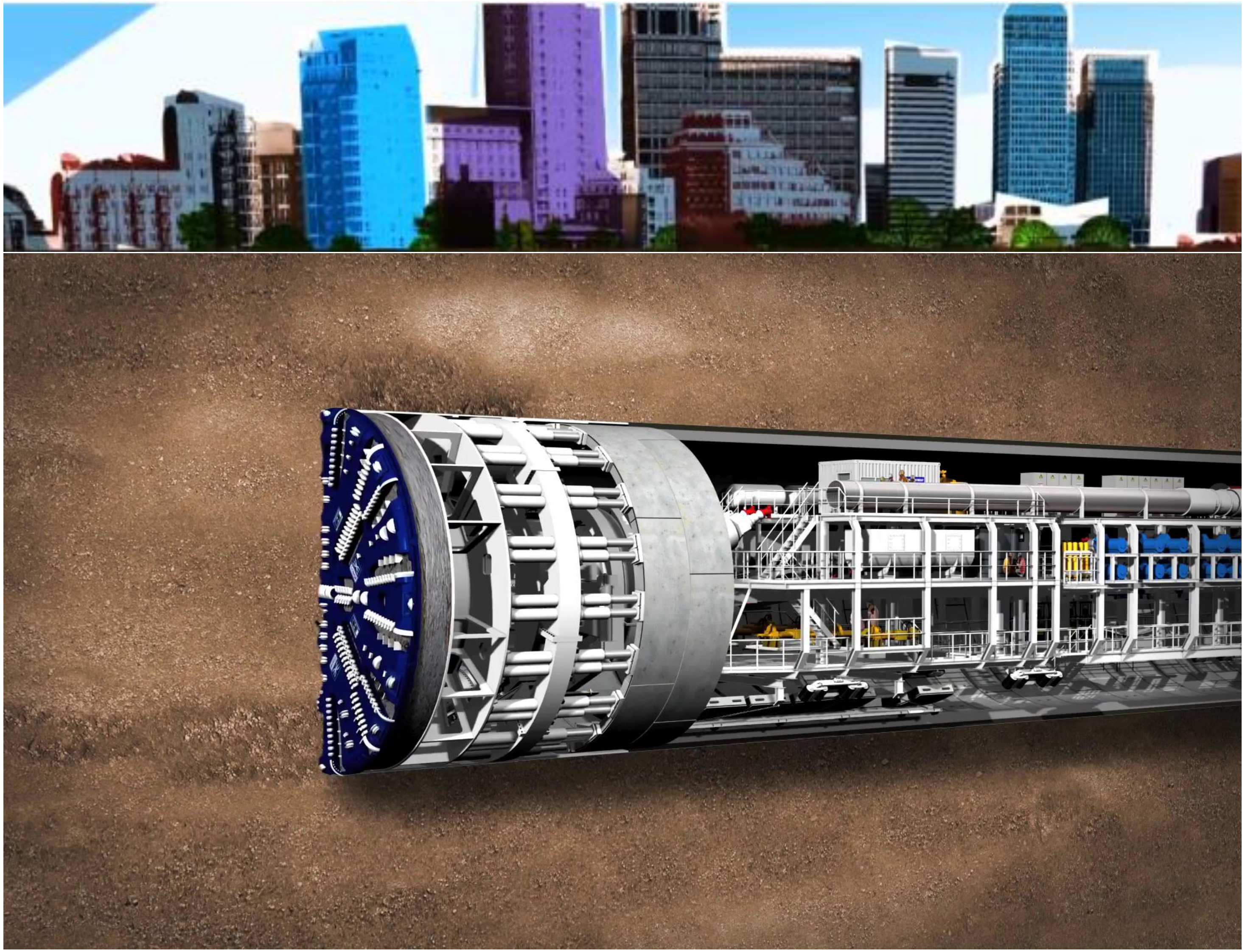
- Motivation
- Problem Description
  - Tunnel Simulation Model
  - Governing Equations
  - Discretization
- Software Overview & Parallellization Implementation
- Numerical Examples
- Conclusions

Motivation

- What is a tunnel for urban infrastructure?
  - Nothing but a tube under ground
  - Used for train traffic (frequently)
  - Or urban street (likely for the above-ground tunnel)

- Why do we do tunnel simulation?
  - Understand the impact of tunnel construction on existing urban infrastructure (i.e buildings)
  - Compute the risk factor
  - Optimize the construction process parameters
  - Build a prediction model

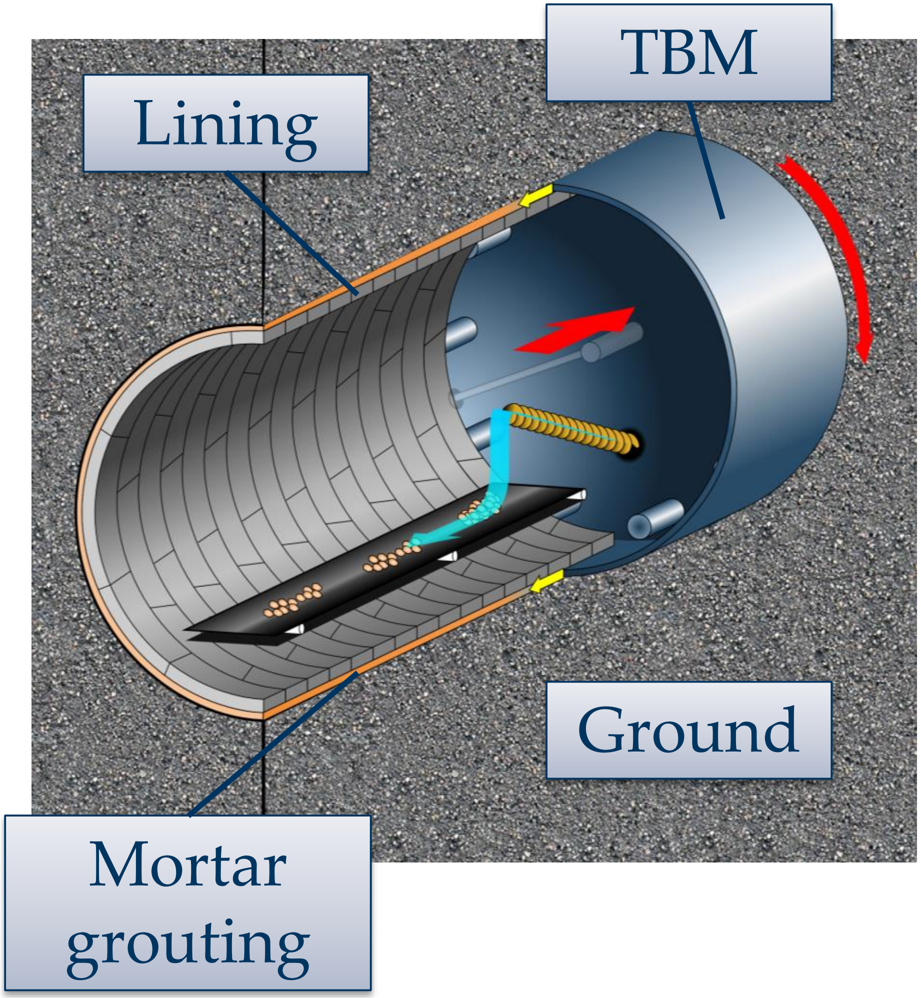
Modern Tunneling Concept: Mechanized Tunneling



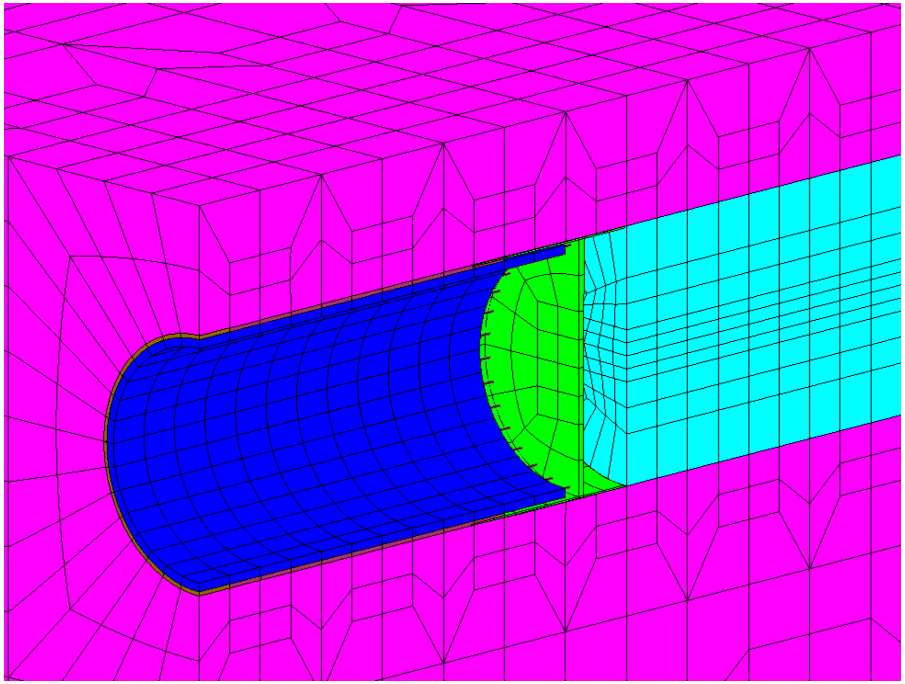
PROBLEM DESCRIPTION

Geometry Description of the Tunnel

Representative Model



FEM Model



Governing Equations

- PDEs

$$div\,[\boldsymbol{\sigma}^s - p^w \mathbf{1}] + [(1-n)\rho^s + \boldsymbol{n}\rho^w]\boldsymbol{g} = 0$$

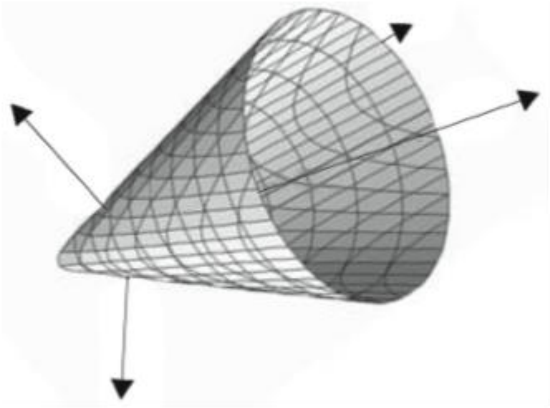
$$div\,\dot{\boldsymbol{u}}^s + div\left[\frac{K}{\mu^w}(-grad\,p^w + \rho^w \boldsymbol{g})\right] = 0$$

- Stress-Strain relationship

Elastoplastic

$$\boldsymbol{\sigma}^s = C^e : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p)$$

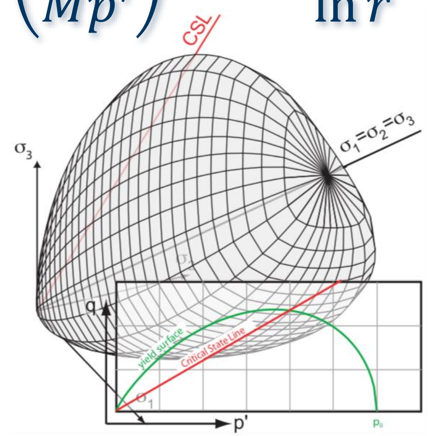
$$\sqrt{J_2\left(\frac{1}{2}\boldsymbol{\sigma}^s : \boldsymbol{\sigma}^s\right)} + \eta I_3(\boldsymbol{\sigma}^s) - \xi c(\dot{\boldsymbol{\varepsilon}}^p) \leq 0$$



Critical State

$$\boldsymbol{\sigma}^s = C^e(p') : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p)$$

$$\left(\frac{q}{Mp'}\right)^n + \frac{ln\left(\frac{p'}{p'_{oj}}\right)}{\ln r} \leq 0$$



Discretization

- Initial boundary value problem

$$find\ u^h, p^h \in U^h, P^h, \mathcal{L}\left(u^h, p^h, v^h, q^h\right) = f\left(v^h, q^h\right)\ \forall v^h \in V^h, q^h \in Q^h$$

- Q2-P1 discretization

$$u^h = \sum^{\mathcal{V}(\mathcal{E})} N_i \hat{u}_i \qquad\qquad p^h = \sum^{\mathcal{V}(\mathcal{E}^c)} N_i \hat{p}_i$$

- Galerkin method:

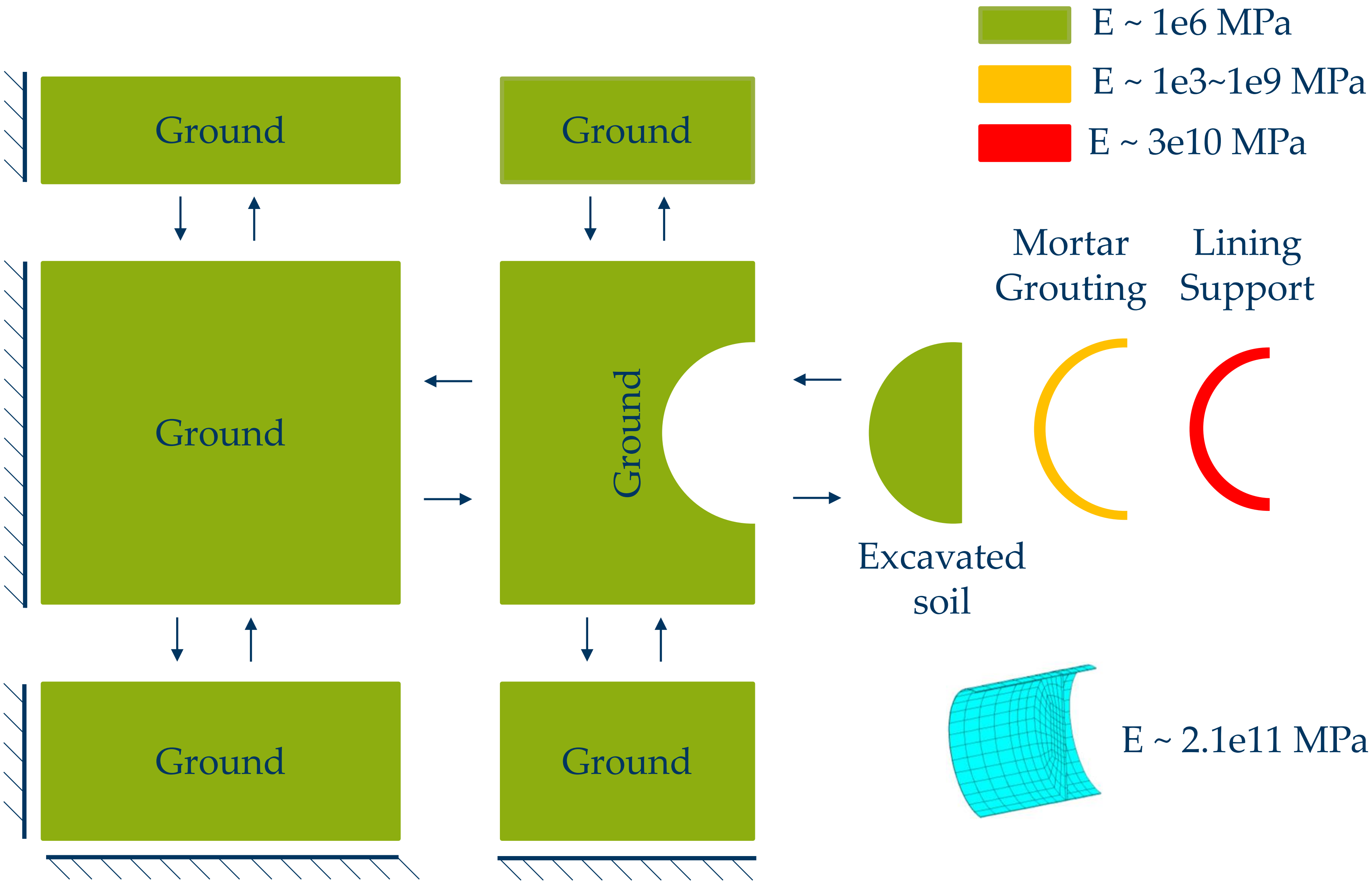
$$K_{ij} = \mathcal{L}\left(N_i, N_j\right) = \begin{bmatrix} K_{uu} & K_{wu} \\ K_{uw} & K_{ww} \end{bmatrix}_{ij}$$

In which:

$$K_{uu}^e = \int_{\Omega^e} B^T D_e B |J| dX \qquad K_{uw}^e = -\int_{\Omega^e} B^T I_v N_p |J| dX$$

$$K_{wu}^e = 0 \qquad K_{ww}^e = -\int_{\Omega^e} B^T \frac{K}{\mu^w} B |J| dX$$

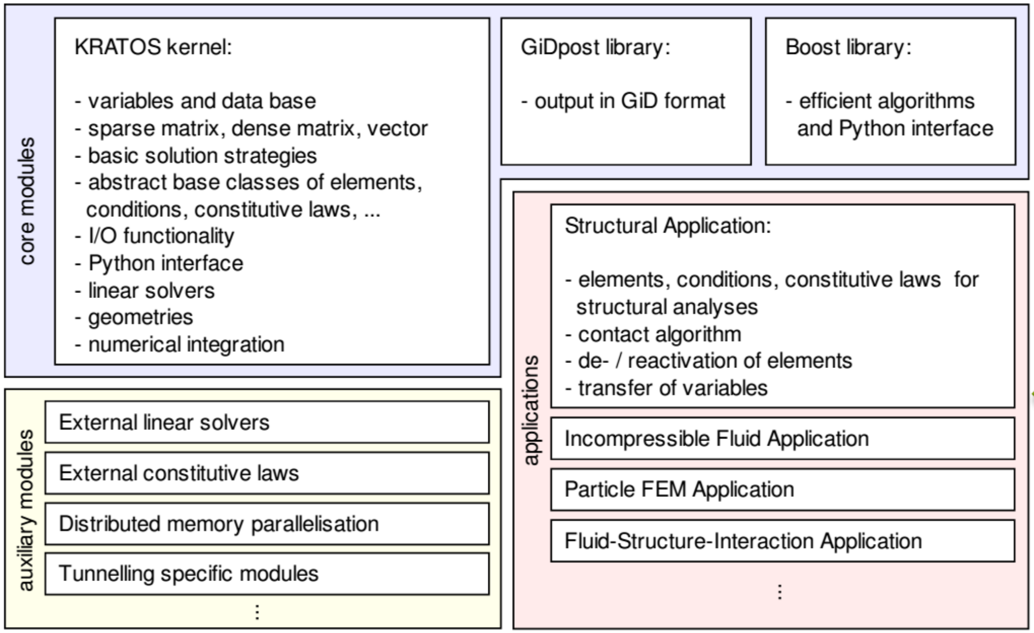
Material Inhomogeneity



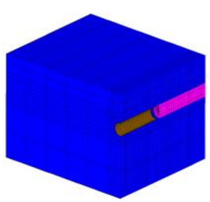
PARALELLIZATION

Software Infrastructure

KRATOS MULTI-PHYSICS [1]



- MortarApplication
  + Mortar Tying/Contact
- MetisApplication
  + Domain Decomposition
- DistributedBuildersApplication
  + Parallel Assembly (MPI)
- PetscSolversApplication
  + Parallel Linear Algebra (solver/preconditioner)



[1] Dadvand & Rossi et al, An Object-oriented Environment for Developing Finite Element Codes for Multi-disciplinary Applications, DOI 10.1007/s11831-010-9045-2
[2] Balay et al, PETSc user manual

Software Infrastructure

Heavy used of:
+ boost shared_ptr
+ Template
+ boost ublas
Python interface

core modules

KRATOS kernel:

- variables and data base
- sparse matrix, dense matrix, vector
- basic solution strategies
- abstract base classes of elements, conditions, constitutive laws, ...
- I/O functionality
- Python interface
- linear solvers
- geometries
- numerical integration

Software Design

STRUCTURAL MECHANICS

MortarApplication
+ Mortar Tying/Contact

MetisApplication
+ Domain Decomposition

DistributedBuildersApplication
+ Parallel Assembly (MPI)

PetscSolversApplication
+ Parallel Linear Algebra
(solver/preconditioner)

+ domain decomposition

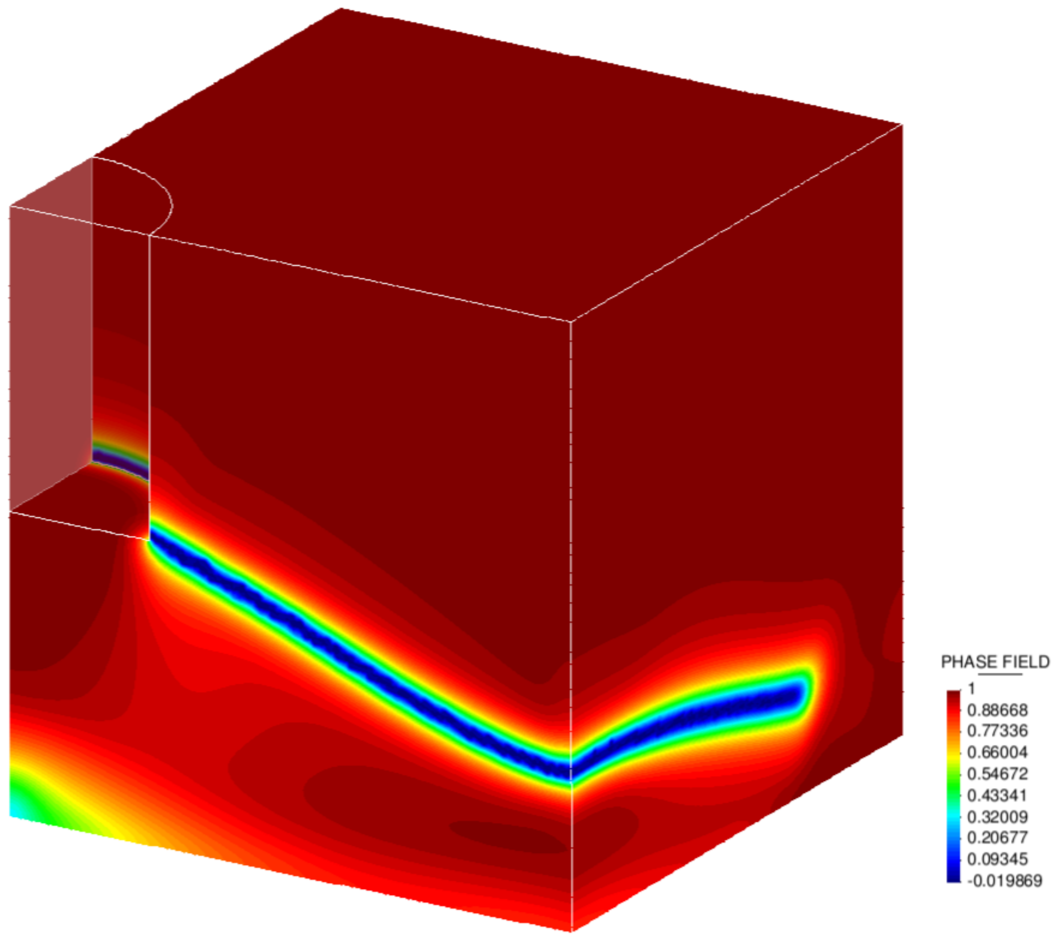
+ wrappers for Mat, Vec
+ preallocation
+ ghost layer communication

+ construct preconditioner
+ call KSPSolve

NUMERICAL EXAMPLES

Numerical Example 1

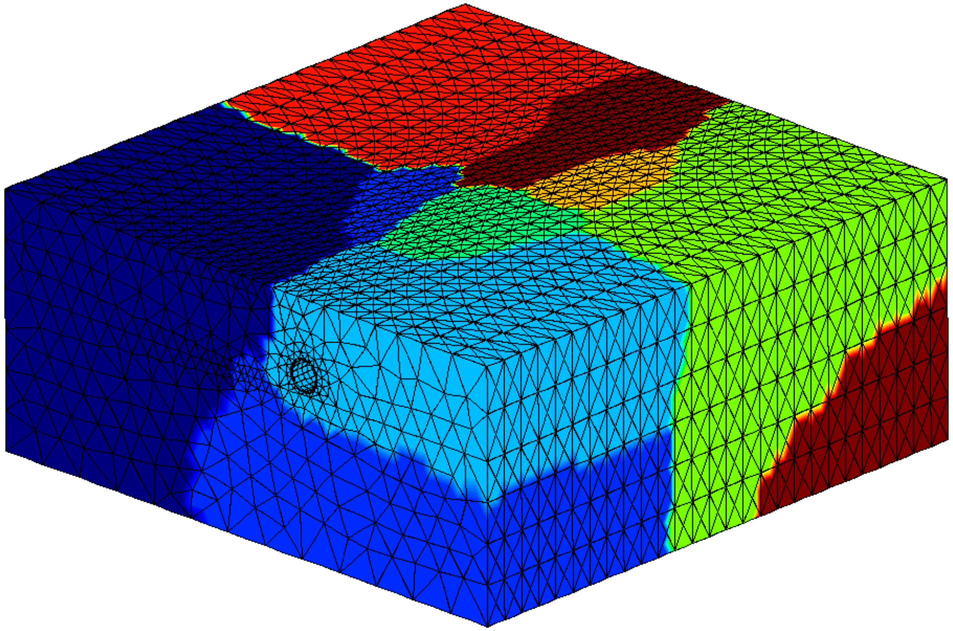
- Fracture Simulation using Phase Field method
  - # nodes: 1.759.004
  - # elements: 10.188.671
  - # dofs: 5.277.012
  - MPI processes = 64
  - GMRES + BoomerAMG
  - Staggered solver
    - Displacement field: ~120s
    - Phase field fracture: ~18s



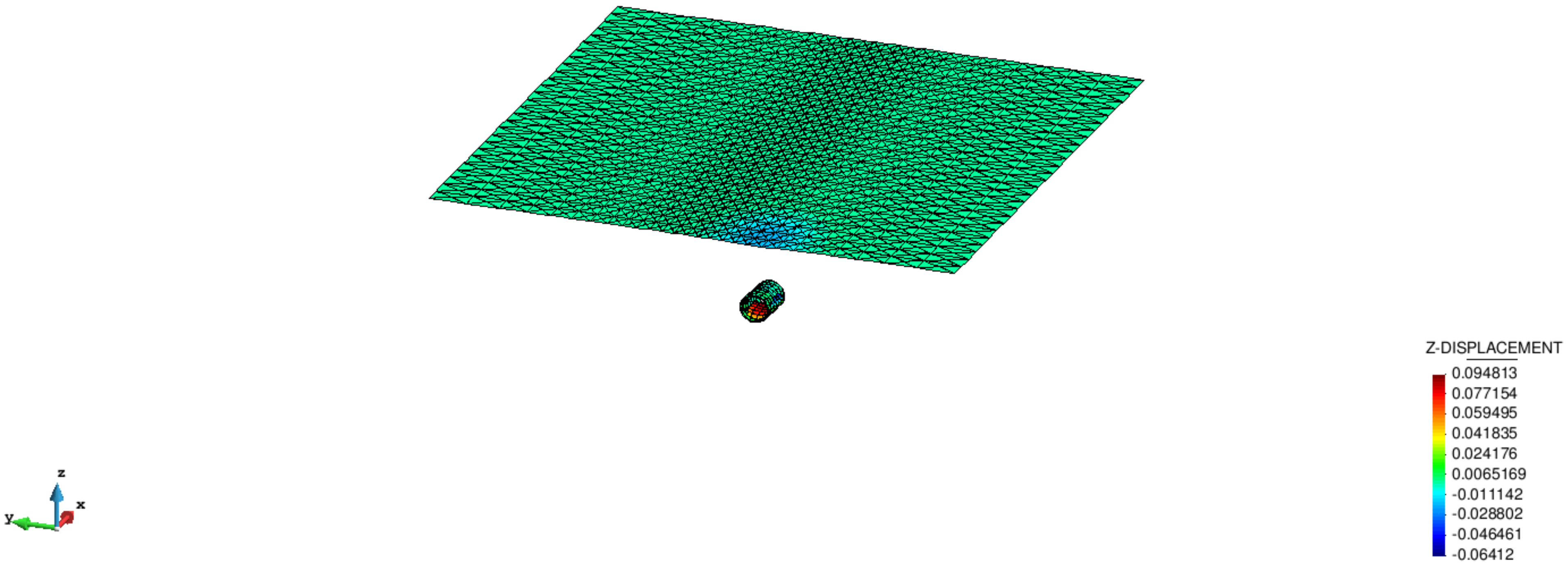
Numerical Example 2

- Reference Tunnel Project 1

  - One phase (pure displacement) discretization
  - No contact between the Tunnel Boring Machine and the soil
  - 433.298 nodes
  - 303.398 tets
  - 1.205.521 dofs
  - 8 Mpi processes
  - GMRES + BoomerAMG + diagonal scaling



Numerical Example 2

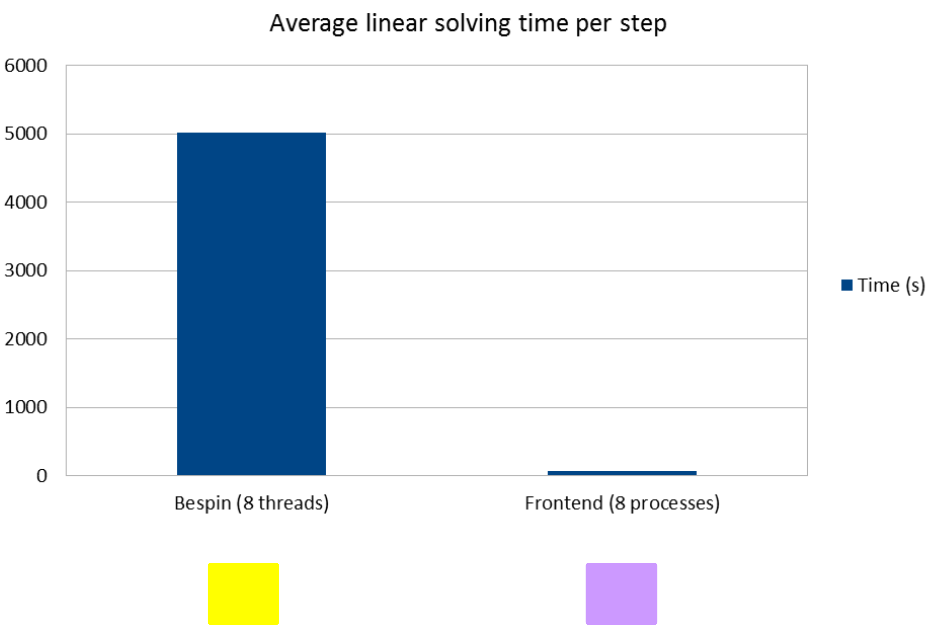


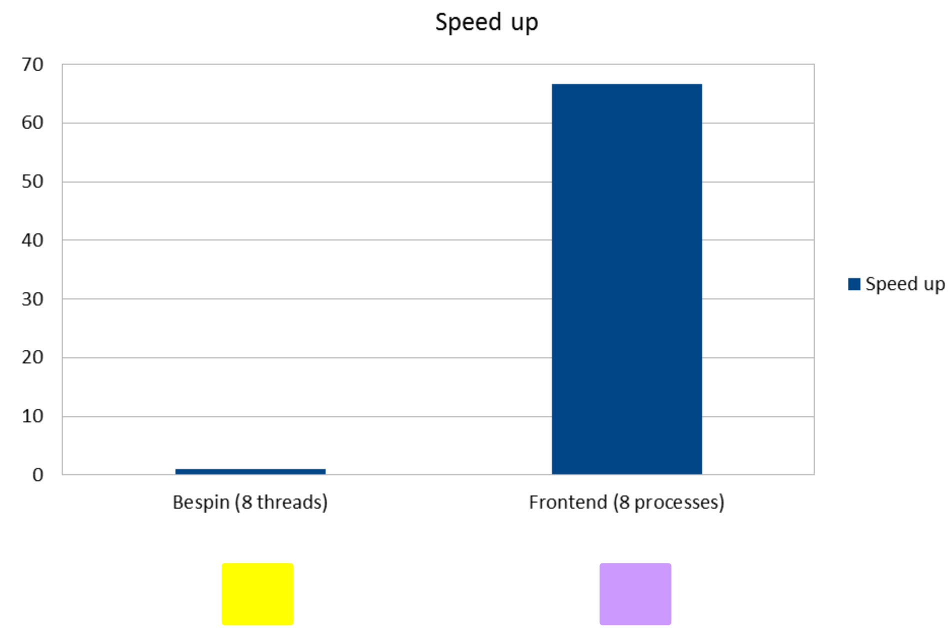
Numerical Example 2

- Reference Project 1

MKL Pardiso

PETSc - BoomerAMG

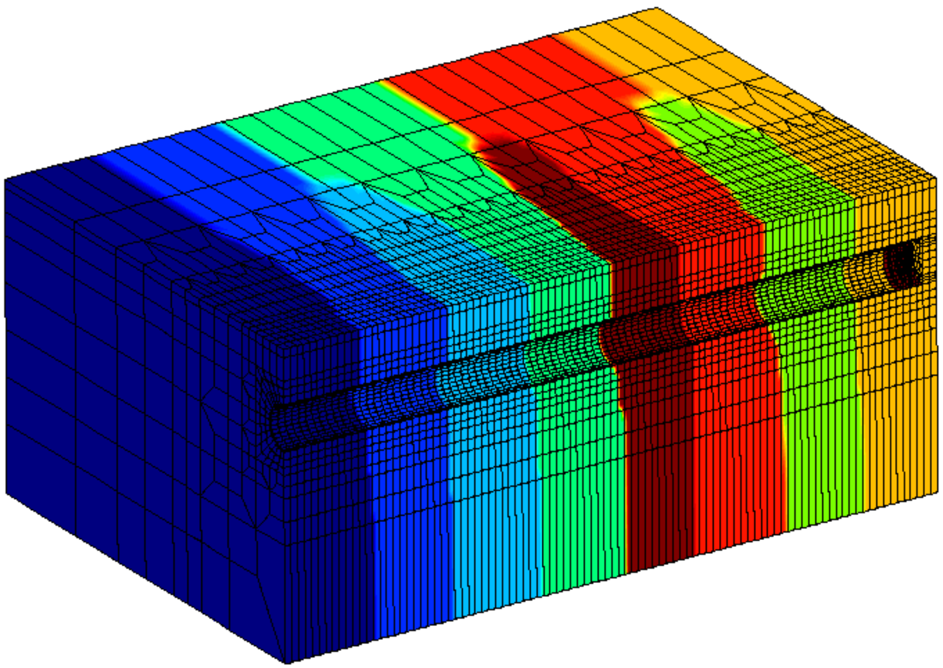




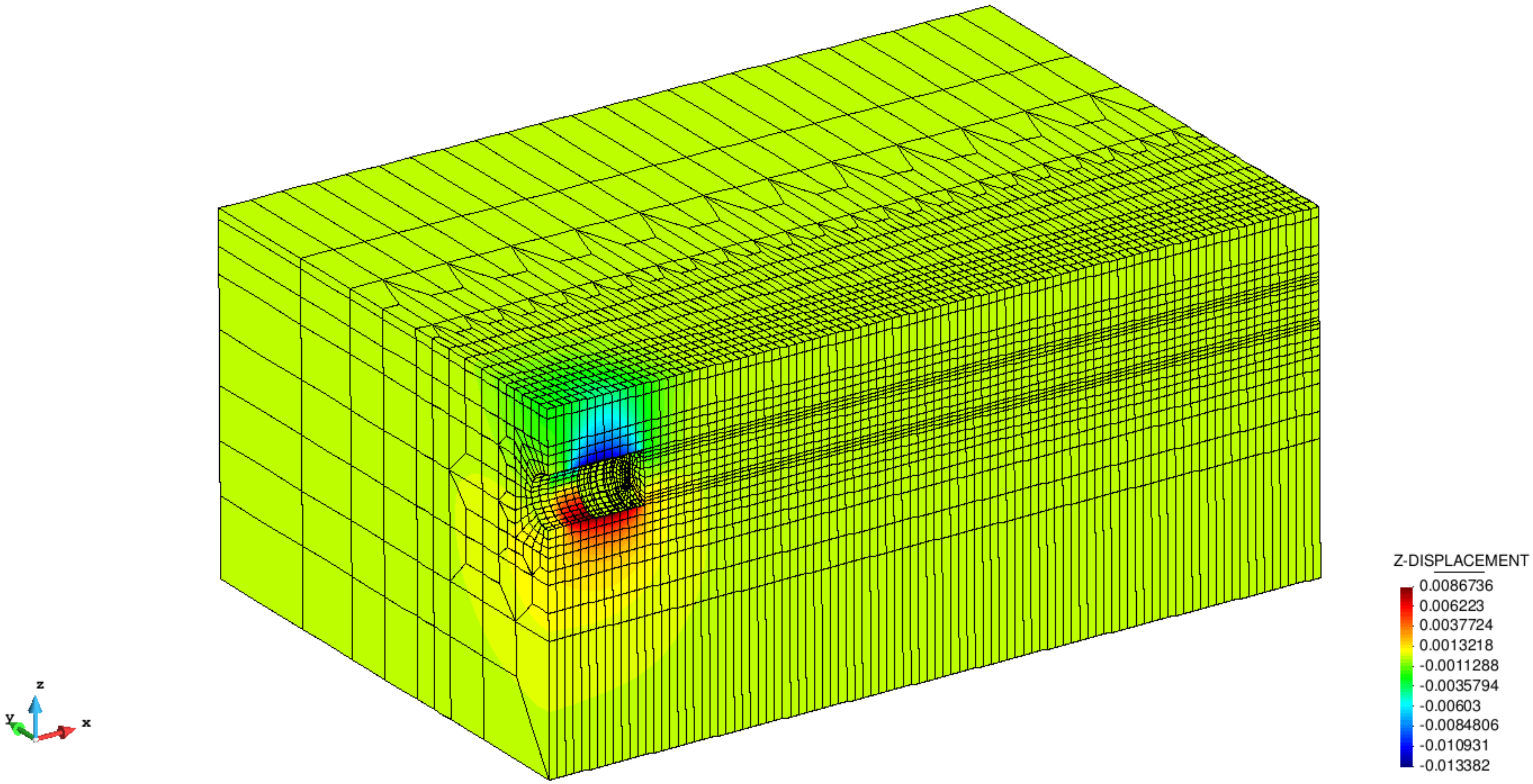
Saving: 2d -> 3hrs

Numerical Example 3

- Reference Project 2
  - Two phase discretization
  - Critical State Soil Model
  - Penalty Contact between the Tunnel Boring Machine and the soil
  - 8 Mpi processes
  - Direct solver: MUMPS



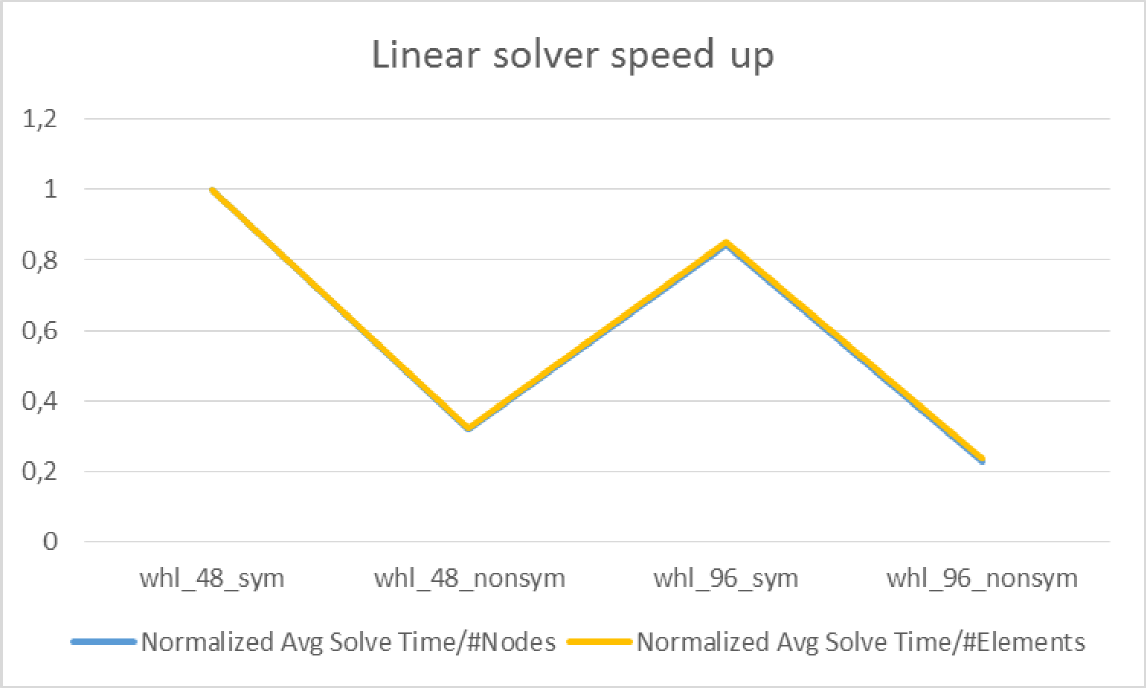
Numerical Example 3



Numerical Example 3

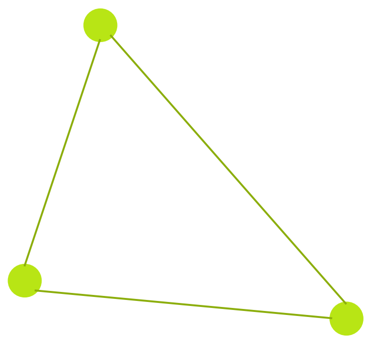
- Timing with different mesh

#nodes	88724	171882	173024	335202
#elements	10256	20240	20240	40480
Avg Solve Time (s)	66,79	406,1	154,81	1108,5
Normalized Avg Solve Time/#Nodes	1	0,319	0,841	0,228
Normalized Avg Solve Time/#Elements	1	0,325	0,851	0,238



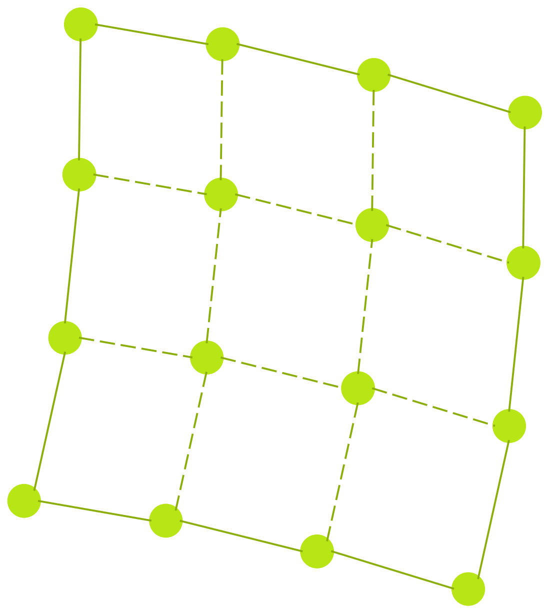
Tunnel Simulation With Isogeometric Method

Standard Isoparametric Finite Element

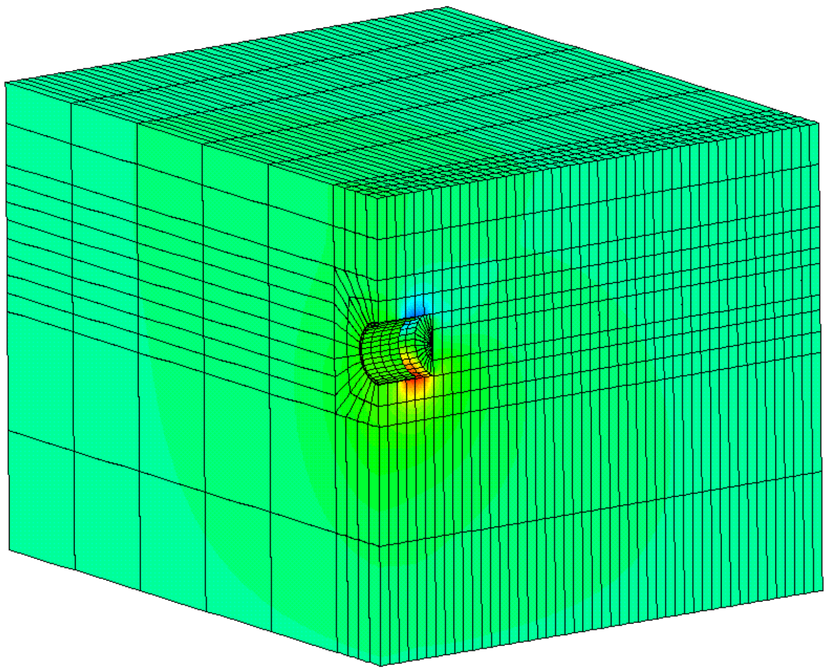


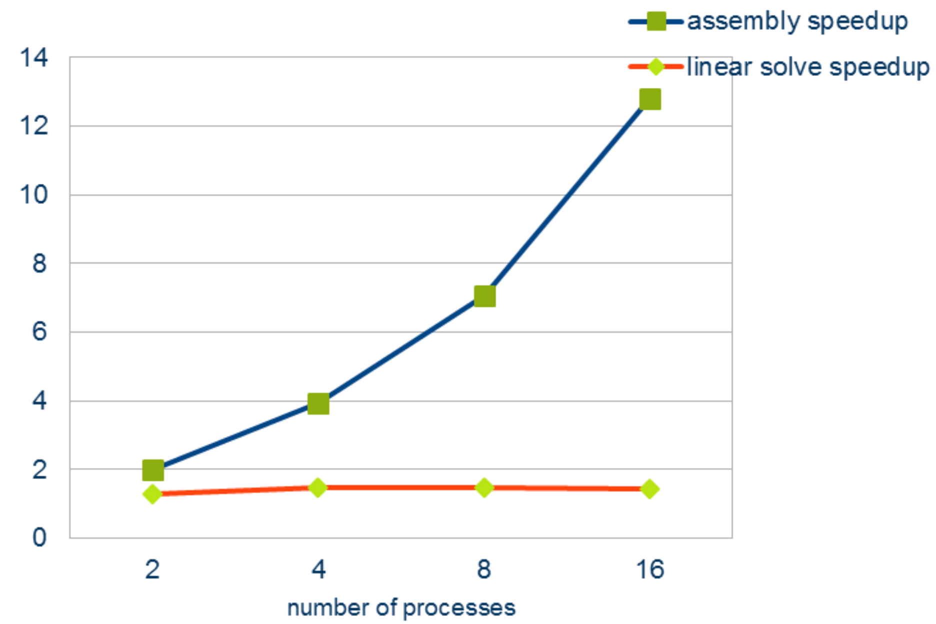
VS

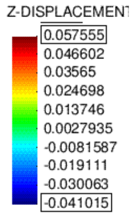
Isogeometric Finite Element



Tunnel Simulation With Isogeometric Method







Conclusions

- The paralellization works and produces expected results
- One phase solve => AMG works as expected
- MUMPS does not scale
- Future works:
  - Development of block preconditioner for two-phase and contact problem
  - Tuning multigrid solver
  - Integrate SNES

THANK YOU !

QUESTIONS ?

This work is part of sub-project C1, within Collaborative Research Centre SFB837 - http://sfb837.sd.rub.de - Interaction Modeling in Mechanized Tunneling - Ruhr University Bochum, Germany

DFG Deutsche Forschungsgemeinschaft