

PERMON Library for Quadratic Programming

Jakub Kruzik, Marek Pecha, David Horak, Zdenek Dostal, Martin Cermak, David Hrbac,
Vaclav Hapla
jakub.kruzik@ugn.cas.cz

June 7, 2023
PETSc User Meeting 2023, Chicago



Institute of Geonics
Czech Academy of Sciences



Department
of Applied Mathematics

Department of applied mathematics
VSB - Technical University of Ostrava

Outline

1. PERMON introduction
2. Total FETI (TFETI)
3. Applications

PERMON Toolbox

- Collection of C libraries
- Based on/extends PETSc
- Developed since 2011
- Published under FreeBSD license
- <https://github.com/permon>



- Collection of C libraries
- Based on/extends PETSc
- Developed since 2011
- Published under FreeBSD license
- <https://github.com/permon>
- PERMON
 - General, massively parallel, QP solution framework
 - QP problems, transformations, solvers



- Collection of C libraries
- Based on/extends PETSc
- Developed since 2011
- Published under FreeBSD license
- <https://github.com/permon>
- PERMON
 - General, massively parallel, QP solution framework
 - QP problems, transformations, solvers
 - FETI DDM implementation



- Collection of C libraries
- Based on/extends PETSc
- Developed since 2011
- Published under FreeBSD license
- <https://github.com/permon>
- PERMON
 - General, massively parallel, QP solution framework
 - QP problems, transformations, solvers
 - FETI DDM implementation
- PermonSVM
 - Library and binaries for the solution of linear SVMs



QP Examples

- Linear elasticity
- Contact problems with friction
- Elasto-plasticity
- Shape optimization
- Vehicle Routing Problems
- Support Vector Machines
- Least-squares regression
- Data fitting
- Data mining
- Medical imaging
- Control systems
- Ice-sheet melting and climate changes modelling

$$\begin{aligned} & \operatorname{argmin} \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{b} \\ & \mathbf{B}_E \mathbf{x} = \mathbf{c}_E \\ & \mathbf{B}_I \mathbf{x} \leq \mathbf{c}_I \\ & \mathbf{l}_b \leq \mathbf{x} \leq \mathbf{u}_b \end{aligned}$$

Workflow - QP and QPS Objects

- Workflow:

- QP problem specification

```
QP qp; QPCreate(comm,&qp); QPSetOperator(qp,A); QPSetRhs(qp,b);  
QPSetEq(qp,Beq,ceq); QPSetBox(qp,is,lb,ub);
```

Workflow - QP and QPS Objects

- Workflow:

- QP problem specification

```
QP qp; QPCreate(comm,&qp); QPSetOperator(qp,A); QPSetRhs(qp,b);  
QPSetEq(qp,Beq,ceq); QPSetBox(qp,is,lb,ub);
```

- QP transformations (optional)

Workflow - QP and QPS Objects

- Workflow:

- QP problem specification

```
QP qp; QPCreate(comm,&qp); QPSetOperator(qp,A); QPSetRhs(qp,b);  
QPSetEq(qp,Beq,ceq); QPSetBox(qp,is,lb,ub);
```

- QP transformations (optional)
 - Solve (automatic/manual choice of solver)

```
QPS qps; QPSCreate(&qps); QPSSetQP(qps,qp); QPSSetFromOptions(qps);  
QPSSolve(qps); QPGetSolutionVector(qp,&x);
```

Workflow - QP and QPS Objects

- Workflow:

- QP problem specification

```
QP qp; QPCreate(comm,&qp); QPSetOperator(qp,A); QPSetRhs(qp,b);  
QPSetEq(qp,Beq,ceq); QPSetBox(qp,is,lb,ub);
```

- QP transformations (optional)
 - Solve (automatic/manual choice of solver)

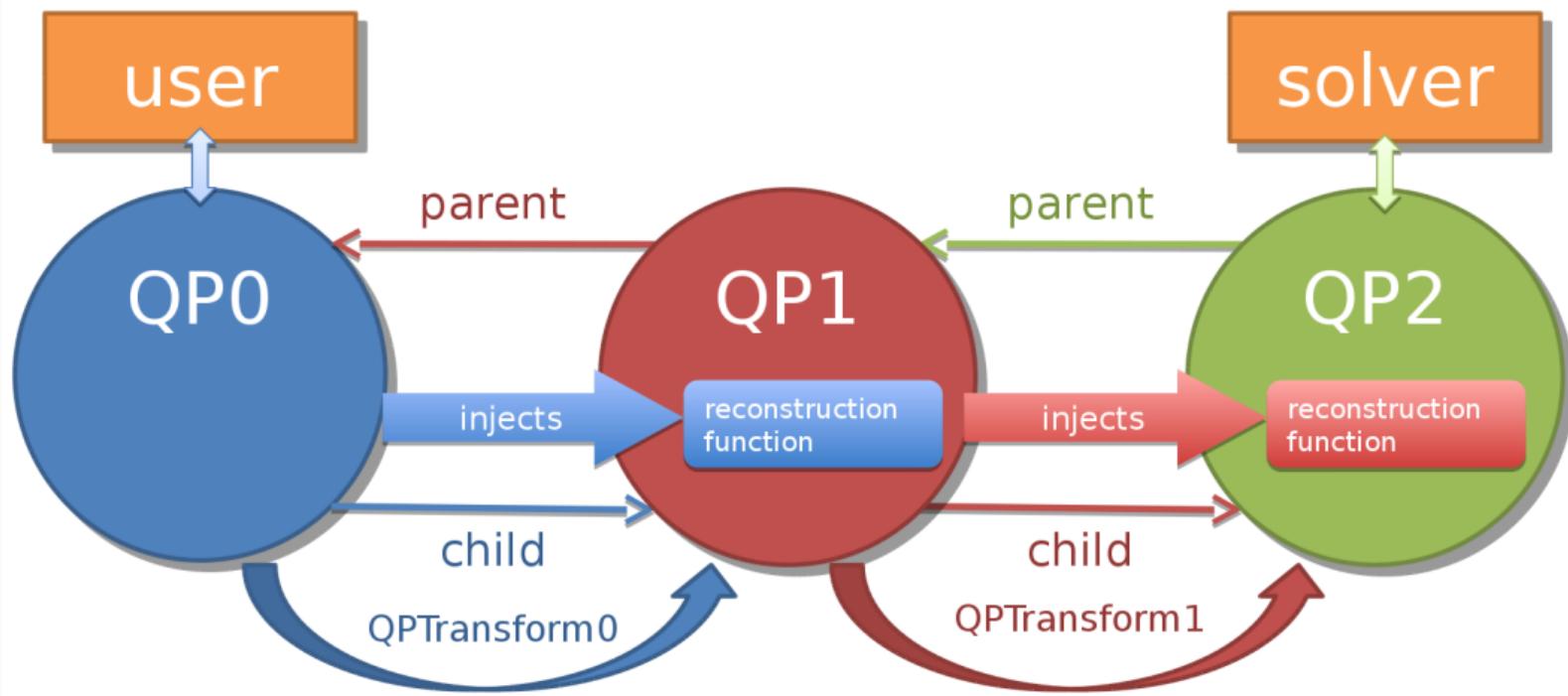
```
QPS qps; QPSCreate(&qps); QPSSetQP(qps,qp); QPSSetFromOptions(qps);  
QPSSolve(qps); QPGetSolutionVector(qp,&x);
```

- Available solvers:

- PETSc KSP (unconstrained)
 - SMALXE, PCPG (equality)
 - MPGP variants, [APGD, PBBF, SPG-QP], PETSc TAO,... (QPC (bound, box,...))

QPT - QP Transform

Transforms QP into a simpler one



Available QP Transformations

```
QPTDualize(QP qp,MatInvType invType,MatRegularizationType regType)
QPTEnforceEqByPenalty(QP qp,PetscReal rho_user,PetscBool
rho_direct)
QPTEnforceEqByProjector(QP qp)
QPTFreezeIneq(QP qp)
QPTHomogenizeEq(QP qp)
QPTMatISToBlockDiag(QP qp)
QPTNormalizeHessian(QP qp)
QPTNormalizeObjective(QP qp)
QPTOrthonormalizeEq(QP qp,MatOrthType type,MatOrthForm form)
QPTRemoveGluingOfDirichletDofs(QP qp)
QPTScale(QP qp)
QPTScaleObjectiveByScalar(QP qp,PetscScalar scale_A,PetscScalar
scale_b)
QPTSplitBE(QP qp)
```

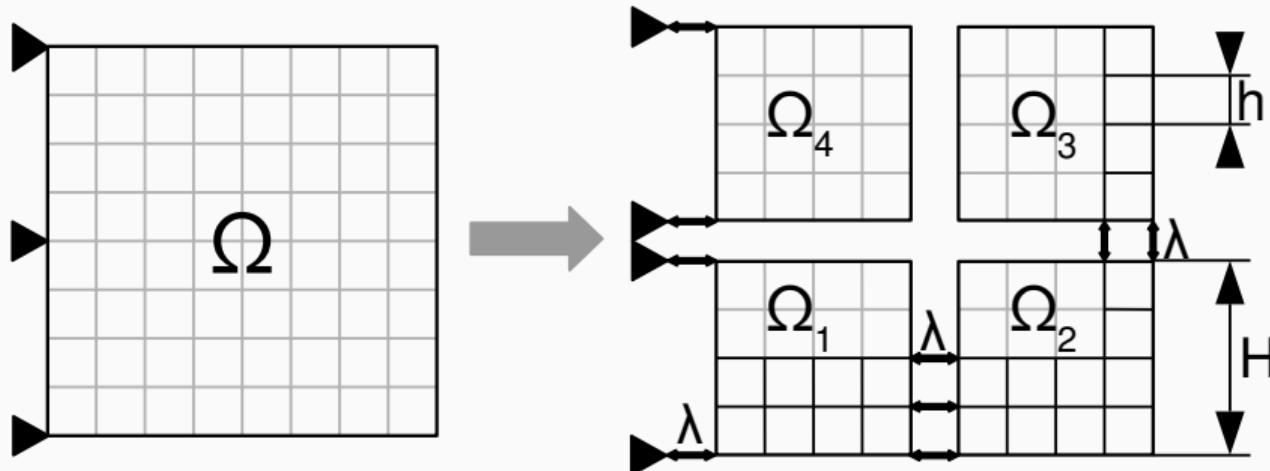
Installing PERMON

1. have PETSc installed and `PETSC_DIR` and `PETSC_ARCH` set
2. `git clone https://github.com/permon/permon`
3. `cd permon`
4. `export PERMON_DIR=${pwd}`
5. `make`

The library is in `$PERMON_DIR/$PETSC_ARCH/lib/libpermon.{so,a}`
See `$PERMON_DIR/src/tutorials/` for examples

Finite Element Tearing and Interconnecting

Total Finite Element Tearing and Interconnecting (TFETI)



Domain Decomposition (Tearing) – Ω into N_s subdomains



Lagrange multipliers (Interconnecting)

$$\mathbf{K} = \text{diag}(\mathbf{K}^1, \dots, \mathbf{K}^{N_S}), \quad \mathbf{R} = \text{diag}(\mathbf{R}^1, \dots, \mathbf{R}^{N_S}),$$

$$\mathbf{f} = [(\mathbf{f}^1)^T, \dots, (\mathbf{f}^{N_S})^T]^T$$

Primal problem

$$\underset{\mathbf{u}}{\text{argmin}} \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u} - \mathbf{f}^T \mathbf{u} \quad \text{s.t.} \quad \begin{aligned} \mathbf{B}_I \mathbf{u} &\leq \mathbf{c}_I \\ \mathbf{B}_E \mathbf{u} &= \mathbf{c}_E \end{aligned}$$

TFETI – Formulation

$$\mathbf{K} = \text{diag}(\mathbf{K}^1, \dots, \mathbf{K}^{N_S}), \quad \mathbf{R} = \text{diag}(\mathbf{R}^1, \dots, \mathbf{R}^{N_S}),$$

$$\mathbf{f} = [(\mathbf{f}^1)^T, \dots, (\mathbf{f}^{N_S})^T]^T$$

Primal problem

$$\underset{\mathbf{u}}{\operatorname{argmin}} \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u} - \mathbf{f}^T \mathbf{u} \quad \text{s.t.} \quad \begin{aligned} \mathbf{B}_I \mathbf{u} &\leq \mathbf{c}_I \\ \mathbf{B}_E \mathbf{u} &= \mathbf{c}_E \end{aligned}$$

Lagrangian

$$\mathbf{B} = [\mathbf{B}_E^T, \mathbf{B}_I^T]^T, \quad \mathbf{c} = [\mathbf{c}_E^T, \mathbf{c}_I^T]^T, \quad \boldsymbol{\lambda} = [\boldsymbol{\lambda}_E^T, \boldsymbol{\lambda}_I^T]^T$$

$$L(\mathbf{u}, \boldsymbol{\lambda}) = \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u} - \mathbf{u}^T \mathbf{f} + (\mathbf{B} \mathbf{u} - \mathbf{c})^T \boldsymbol{\lambda}.$$

TFETI – Formulation

KKT

$$\nabla_{\mathbf{u}} L(\mathbf{u}, \boldsymbol{\lambda}) = \mathbf{K}\mathbf{u} - \mathbf{f} + \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{o}, \quad (1)$$

$$\nabla_{\boldsymbol{\lambda}_E} L(\mathbf{u}, \boldsymbol{\lambda}) = \mathbf{B}_E \mathbf{u} - \mathbf{c}_E = \mathbf{o}, \quad (2)$$

$$\nabla_{\boldsymbol{\lambda}_I} L(\mathbf{u}, \boldsymbol{\lambda}) = \mathbf{B}_I \mathbf{u} - \mathbf{c}_I \leq \mathbf{o}, \quad (3)$$

$$\boldsymbol{\lambda}_I^T (\mathbf{B}_I \mathbf{u} - \mathbf{c}_I) = \mathbf{o}, \quad (4)$$

$$\boldsymbol{\lambda}_I \geq \mathbf{o} \quad (5)$$

TFETI – Formulation

KKT

$$\nabla_{\mathbf{u}} L(\mathbf{u}, \boldsymbol{\lambda}) = \mathbf{K}\mathbf{u} - \mathbf{f} + \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{o}, \quad (1)$$

$$\nabla_{\boldsymbol{\lambda}_E} L(\mathbf{u}, \boldsymbol{\lambda}) = \mathbf{B}_E \mathbf{u} - \mathbf{c}_E = \mathbf{o}, \quad (2)$$

$$\nabla_{\boldsymbol{\lambda}_I} L(\mathbf{u}, \boldsymbol{\lambda}) = \mathbf{B}_I \mathbf{u} - \mathbf{c}_I \leq \mathbf{o}, \quad (3)$$

$$\boldsymbol{\lambda}_I^T (\mathbf{B}_I \mathbf{u} - \mathbf{c}_I) = \mathbf{o}, \quad (4)$$

$$\boldsymbol{\lambda}_I \geq \mathbf{o} \quad (5)$$

(1) is solvable iff

$$(\mathbf{f} - \mathbf{B}^T \boldsymbol{\lambda}) \in \text{Im } \mathbf{K} \Leftrightarrow \mathbf{R}^T (\mathbf{f} - \mathbf{B}^T \boldsymbol{\lambda}) = \mathbf{o},$$

where $\mathbf{R} = \text{span}\{\text{Ker } \mathbf{K}\}$. Then

TFETI – Formulation

KKT

$$\nabla_{\mathbf{u}} L(\mathbf{u}, \boldsymbol{\lambda}) = \mathbf{K}\mathbf{u} - \mathbf{f} + \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{o}, \quad (1)$$

$$\nabla_{\boldsymbol{\lambda}_E} L(\mathbf{u}, \boldsymbol{\lambda}) = \mathbf{B}_E \mathbf{u} - \mathbf{c}_E = \mathbf{o}, \quad (2)$$

$$\nabla_{\boldsymbol{\lambda}_I} L(\mathbf{u}, \boldsymbol{\lambda}) = \mathbf{B}_I \mathbf{u} - \mathbf{c}_I \leq \mathbf{o}, \quad (3)$$

$$\boldsymbol{\lambda}_I^T (\mathbf{B}_I \mathbf{u} - \mathbf{c}_I) = \mathbf{o}, \quad (4)$$

$$\boldsymbol{\lambda}_I \geq \mathbf{o} \quad (5)$$

(1) is solvable iff

$$(\mathbf{f} - \mathbf{B}^T \boldsymbol{\lambda}) \in \text{Im } \mathbf{K} \Leftrightarrow \mathbf{R}^T (\mathbf{f} - \mathbf{B}^T \boldsymbol{\lambda}) = \mathbf{o},$$

where $\mathbf{R} = \text{span}\{\text{Ker } \mathbf{K}\}$. Then

$$\mathbf{K}\mathbf{u} - \mathbf{f} + \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{K}\mathbf{R}\boldsymbol{\alpha} \Leftrightarrow \mathbf{u} = \mathbf{K}^\dagger (\mathbf{f} - \mathbf{B}^T \boldsymbol{\lambda}) + \mathbf{R}\boldsymbol{\alpha},$$

where $\mathbf{K}\mathbf{K}^\dagger \mathbf{K} = \mathbf{K}$.

TFETI – Formulation

Substituting \mathbf{u} into KKT

$$[\nabla_{\boldsymbol{\lambda}} L(\mathbf{u}, \boldsymbol{\lambda})]_E = \left[-BK^\dagger B^T \boldsymbol{\lambda} + (BK^\dagger \mathbf{f} - \mathbf{c}) + BR\boldsymbol{\alpha} \right]_E = \mathbf{o},$$

$$[\nabla_{\boldsymbol{\lambda}} L(\mathbf{u}, \boldsymbol{\lambda})]_I = \left[-BK^\dagger B^T \boldsymbol{\lambda} + (BK^\dagger \mathbf{f} - \mathbf{c}) + BR\boldsymbol{\alpha} \right]_I \leq \mathbf{o},$$

$$\boldsymbol{\lambda}_I^T \left[-BK^\dagger B^T \boldsymbol{\lambda} + (BK^\dagger \mathbf{f} - \mathbf{c}) + BR\boldsymbol{\alpha} \right]_I = \mathbf{o},$$

TFETI – Formulation

Substituting \mathbf{u} into KKT

$$\begin{aligned} [\nabla_{\boldsymbol{\lambda}} L(\mathbf{u}, \boldsymbol{\lambda})]_E &= \left[-\mathbf{B}\mathbf{K}^\dagger \mathbf{B}^T \boldsymbol{\lambda} + (\mathbf{B}\mathbf{K}^\dagger \mathbf{f} - \mathbf{c}) + \mathbf{B}\mathbf{R}\boldsymbol{\alpha} \right]_E = \mathbf{o}, \\ [\nabla_{\boldsymbol{\lambda}} L(\mathbf{u}, \boldsymbol{\lambda})]_I &= \left[-\mathbf{B}\mathbf{K}^\dagger \mathbf{B}^T \boldsymbol{\lambda} + (\mathbf{B}\mathbf{K}^\dagger \mathbf{f} - \mathbf{c}) + \mathbf{B}\mathbf{R}\boldsymbol{\alpha} \right]_I \leq \mathbf{o}, \\ \boldsymbol{\lambda}_I^T \left[-\mathbf{B}\mathbf{K}^\dagger \mathbf{B}^T \boldsymbol{\lambda} + (\mathbf{B}\mathbf{K}^\dagger \mathbf{f} - \mathbf{c}) + \mathbf{B}\mathbf{R}\boldsymbol{\alpha} \right]_I &= \mathbf{o}, \end{aligned}$$

are together with $\boldsymbol{\lambda}_i \geq \mathbf{o}$ and $\mathbf{R}^T(\mathbf{f} - \mathbf{B}^T \boldsymbol{\lambda}) = \mathbf{o}$ KKT conditions for

$$\underset{\boldsymbol{\lambda}}{\operatorname{argmax}} -\frac{1}{2} \boldsymbol{\lambda}^T \mathbf{B}\mathbf{K}^\dagger \mathbf{B}^T \boldsymbol{\lambda} + \boldsymbol{\lambda}^T (\mathbf{B}\mathbf{K}^\dagger \mathbf{f} - \mathbf{c}) - \frac{1}{2} \mathbf{f}^T \mathbf{K}^\dagger \mathbf{f} \quad \text{s.t. } \boldsymbol{\lambda}_I \geq \mathbf{o}, \quad \mathbf{R}^T \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{R}^T \mathbf{f}$$

with Lagrangian

$$\Lambda(\boldsymbol{\lambda}, \boldsymbol{\alpha}) = -\frac{1}{2} \boldsymbol{\lambda}^T \mathbf{B}\mathbf{K}^\dagger \mathbf{B}^T \boldsymbol{\lambda} + \boldsymbol{\lambda}^T (\mathbf{B}\mathbf{K}^\dagger \mathbf{f} - \mathbf{c}) - \frac{1}{2} \mathbf{f}^T \mathbf{K}^\dagger \mathbf{f} + \boldsymbol{\alpha}^T (\mathbf{R}^T \mathbf{B}^T \boldsymbol{\lambda} - \mathbf{R}^T \mathbf{f})$$

TFETI – Dual Formulation

$$\operatorname{argmax}_{\boldsymbol{\lambda}} -\frac{1}{2} \boldsymbol{\lambda}^T \mathbf{B} \mathbf{K}^\dagger \mathbf{B}^T \boldsymbol{\lambda} + \boldsymbol{\lambda}^T (\mathbf{B} \mathbf{K}^\dagger \mathbf{f} - \mathbf{c}) - \frac{1}{2} \mathbf{f}^T \mathbf{K}^\dagger \mathbf{f} \quad \text{s.t.} \quad \boldsymbol{\lambda}_I \geq \mathbf{o}, \quad \mathbf{R}^T \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{R}^T \mathbf{f}$$

$$\begin{aligned}\mathbf{F} &= \mathbf{B} \mathbf{K}^\dagger \mathbf{B}^T, & \mathbf{G} &= \mathbf{R}^T \mathbf{B}^T \\ \mathbf{e} &= \mathbf{R}^T \mathbf{f}, & \mathbf{d} &= \mathbf{B} \mathbf{K}^\dagger \mathbf{f} - \mathbf{c}\end{aligned}$$

Dual problem

$$\operatorname{argmin}_{\boldsymbol{\lambda}} \frac{1}{2} \boldsymbol{\lambda}^T \mathbf{F} \boldsymbol{\lambda} - \boldsymbol{\lambda}^T \mathbf{d} \quad \text{s.t.} \quad \begin{aligned}\boldsymbol{\lambda}_I &\geq 0 \\ \mathbf{G} \boldsymbol{\lambda} &= \mathbf{e}\end{aligned}$$

TFETI – Improving Formulation

$$\boldsymbol{\lambda} = \widehat{\boldsymbol{\lambda}} + \widetilde{\boldsymbol{\lambda}} \quad \text{s.t.} \quad \mathbf{G}\widetilde{\boldsymbol{\lambda}} = \mathbf{e}$$

Homogenize equality constraint

$$\operatorname{argmin}_{\widehat{\boldsymbol{\lambda}}} \frac{1}{2} \widehat{\boldsymbol{\lambda}}^T \mathbf{F} \widehat{\boldsymbol{\lambda}} - \widehat{\boldsymbol{\lambda}}^T (\mathbf{d} - \mathbf{F} \widetilde{\boldsymbol{\lambda}}) \quad \text{s.t.} \quad \begin{aligned} \widehat{\boldsymbol{\lambda}}_I &\geq -\widetilde{\boldsymbol{\lambda}}_I \\ \mathbf{G}\widehat{\boldsymbol{\lambda}} &= \mathbf{o} \end{aligned}$$

TFETI – Improving Formulation

$$\boldsymbol{\lambda} = \widehat{\boldsymbol{\lambda}} + \widetilde{\boldsymbol{\lambda}} \quad \text{s.t.} \quad \mathbf{G}\widetilde{\boldsymbol{\lambda}} = \mathbf{e}$$

Homogenize equality constraint

$$\operatorname{argmin}_{\widehat{\boldsymbol{\lambda}}} \frac{1}{2} \widehat{\boldsymbol{\lambda}}^T \mathbf{F} \widehat{\boldsymbol{\lambda}} - \widehat{\boldsymbol{\lambda}}^T (\mathbf{d} - \mathbf{F} \widetilde{\boldsymbol{\lambda}}) \quad \text{s.t.} \quad \begin{aligned} \widehat{\boldsymbol{\lambda}}_I &\geq -\widetilde{\boldsymbol{\lambda}}_I \\ \mathbf{G}\widehat{\boldsymbol{\lambda}} &= \mathbf{o} \end{aligned}$$

$$\mathbf{Q} = \mathbf{G}^T (\mathbf{G}\mathbf{G}^T)^{-1} \mathbf{G}, \quad \mathbf{P} = \mathbf{I} - \mathbf{Q}$$

Project onto natural coarse space and enforce equality const. by penalty

$$\operatorname{argmin}_{\widehat{\boldsymbol{\lambda}}} \frac{1}{2} \widehat{\boldsymbol{\lambda}}^T (\mathbf{P}\mathbf{F}\mathbf{P} + \rho \mathbf{Q}) \widehat{\boldsymbol{\lambda}} - \widehat{\boldsymbol{\lambda}}^T \mathbf{P} (\mathbf{d} - \mathbf{F} \widetilde{\boldsymbol{\lambda}}) \quad \text{s.t.} \quad \widehat{\boldsymbol{\lambda}}_I \geq -\widetilde{\boldsymbol{\lambda}}_I$$

TFETI – Improving Formulation

$$\boldsymbol{\lambda} = \widehat{\boldsymbol{\lambda}} + \widetilde{\boldsymbol{\lambda}} \quad \text{s.t.} \quad \mathbf{G}\widetilde{\boldsymbol{\lambda}} = \mathbf{e}$$

Homogenize equality constraint

$$\operatorname{argmin}_{\widehat{\boldsymbol{\lambda}}} \frac{1}{2} \widehat{\boldsymbol{\lambda}}^T \mathbf{F} \widehat{\boldsymbol{\lambda}} - \widehat{\boldsymbol{\lambda}}^T (\mathbf{d} - \mathbf{F} \widetilde{\boldsymbol{\lambda}}) \quad \text{s.t.} \quad \begin{aligned} \widehat{\boldsymbol{\lambda}}_I &\geq -\widetilde{\boldsymbol{\lambda}}_I \\ \mathbf{G}\widehat{\boldsymbol{\lambda}} &= \mathbf{o} \end{aligned}$$

$$\mathbf{Q} = \mathbf{G}^T (\mathbf{G}\mathbf{G}^T)^{-1} \mathbf{G}, \quad \mathbf{P} = \mathbf{I} - \mathbf{Q}$$

Project onto natural coarse space and enforce equality const. by penalty

$$\operatorname{argmin}_{\widehat{\boldsymbol{\lambda}}} \frac{1}{2} \widehat{\boldsymbol{\lambda}}^T (\mathbf{P}\mathbf{F}\mathbf{P} + \rho \mathbf{Q}) \widehat{\boldsymbol{\lambda}} - \widehat{\boldsymbol{\lambda}}^T \mathbf{P} (\mathbf{d} - \mathbf{F} \widetilde{\boldsymbol{\lambda}}) \quad \text{s.t.} \quad \widehat{\boldsymbol{\lambda}}_I \geq -\widetilde{\boldsymbol{\lambda}}_I$$

$$\kappa(\mathbf{P}\mathbf{F}\mathbf{P} | \text{Im } \mathbf{P}) \leq \text{const} \frac{H}{h}$$

- User provides:
 - stiffness matrix
 - load vector
 - \mathbf{B}_E matrix or l2g mapping, (optionaly \mathbf{B}_I)
 - \mathbf{R} (kernel of stiffness matrix) or it can be generated by **MUMPS**

- User provides:
 - stiffness matrix
 - load vector
 - B_E matrix or l2g mapping, (optionaly B_I)
 - R (kernel of stiffness matrix) or it can be generated by **MUMPS**
- Applies QP transform chain:
 - `QPTDualize(qp);`
 - (`QPTOrthonormalizeEq(qp)` if we have inequality)
 - `QPTHomogenizeEq(qp);`
 - `QPTEnforceEqByProjector(qp);`
 - (`QPTEnforceEqByPenalty(qp)` if we have inequality)

Using PERMON FETI - The Easy Way

1. take a code that is able to generate MATIS
2. add `#include <permonksp.h>`
3. change `PetscInitialize` to `PermonInitialize`
4. change `PetscFinalize` to `PermonFinalize`
5. have **MUMPS** installed
6. compile and link with PERMON
7. `mpirun -n /app ... --ksp_type feti ...`

QPChainViewKKT(QP, PetscViewer) / -qp_chain_view_kkt

```
QP Object: QP_0x84000004_3 (dual_prpnlt_) 4 MPI processes
  type: QP
    #7 in chain, derived by QPTEnforceEqByPenalty
    || x|| = 2.82655518e+01    max( x) = 4.55e+00 = x(234)    min( x) = ...
    || b|| = 4.69041793e-02    max( b) = 5.06e-03 = b(134)    min( b) = ...
    r = ||A*x - b - lambda_lb|| = 0.00e+00    r0/||b|| = 0.00e+00
    r = ||min(x-lb,0)|| = 0.00e+00    r/||b|| = 0.00e+00
    r = ||min(lambda_lb,0)|| = 1.71e-08    r/||b|| = 3.64e-07
    r = |lambda_lb'* (lb-x)| = 1.51e-07    r/||b|| = 3.22e-06
-----
QP Object: QP_0x84000004_4 (dual_proj_) 4 MPI processes
  type: QP
    #6 in chain, derived by QPTEnforceEqByProjector
    || x|| = 2.82655518e+01    max( x) = 4.55e+00 = x(234)    min( x) = ...
    || b|| = 4.69023653e-02    max( b) = 5.05e-03 = b(548)    min( b) = ...
    ||cE|| = 0.00e-00    max(cE) = 0.00e-00 = cE(0)    min(cE) = ...
    r = ||A*x - b + (B'*lambda) - lambda_lb|| = 3.30e-18    r0/||b|| = 7.04e-17
    r = ||BE*x|| = 2.69e-08    r/||b|| = 5.73e-07
```

QPChainViewKKT(QP, PetscViewer) / -qp_chain_view_kkt

```
QP Object: QP_0x84000004_8 (dual_) 4 MPI processes
type: QP
#2 in chain, derived by QPTDualize
|| x|| = 4.47234996e+01    max( x) = 4.55e+00 = x(234)    min( x) = ...
|| b|| = 6.70029516e-02    max( b) = 5.39e-03 = b(807)    min( b) = ...
||cE|| = 1.55000000e+01    max(cE) = 3.65e+00 = cE(0)    min(cE) = ...
r = ||A*x - b + (B'*lambda) - lambda_lb|| = 0.00e+00    r0/||b|| = 0.00e+00
r = ||BE*x-cE||           = 2.22e-08    r/||b|| = 3.31e-07
r = ||min(x-lb,0)||       = 0.00e+00    r/||b|| = 0.00e+00
r = ||min(lambda_lb,0)||   = 1.71e-08    r/||b|| = 2.55e-07
r = |lambda_lb'*(lb-x)|  = 1.51e-07    r/||b|| = 2.25e-06
-----
...

```

QPChainViewKKT(QP, PetscViewer) / -qp_chain_view_kkt

```
QP Object: QP_0x84000004_10 4 MPI processes
type: QP
#0 in chain, derived by
|| x|| = 4.10833800e-01    max( x) = 4.07e-03 = x(2425)    min( x) = ...
|| b|| = 3.72252508e+01    max( b) = 0.00e+00 = b(0)    min( b) = ...
||cE|| = 0.00e-00    max(cE) = 0.00e-00 = cE(0)    min(cE) = ...
||cI|| = 0.00e-00    max(cI) = 0.00e-00 = cI(0)    min(cI) = ...
r = ||A*x - b + B'*lambda|| = 2.22e-08    r0/||b|| = 5.96e-10
r = ||BE*x||          = 2.72e-08    r/||b|| = 7.30e-10
r = ||max(BI*x,0)||   = 5.26e-09    r/||b|| = 1.41e-10
r = ||min(lambda_I,0)|| = 0.00e+00    r/||b|| = 0.00e+00
r = |lambda_I'*(BI*x)|= 5.44e-08    r/||b|| = 1.46e-09
```

QPSViewConvergence(QPS, PetscViewer) / -qps_view_convergence

QPS Object: 4 MPI processes

type: smalxe

last QPSSolve CONVERGED due to CONVERGED_RTOL, KSPReason=2, required 11 iterations

all 1 QPSSolves from last QPSReset/QPSResetStatistics have required 11 iterations

tolerances: rtol=1.0e-06, abstol=1.0e-50, dtol=1.0e+50, maxits=100

smalxe specific:

Total number of inner iterations 103

#hits of M1, eta: 3, 1

#updates of M1, rho: 3, 6

inner QPS Object: (smalxe_) 4 MPI processes

type: mpgp

last QPSSolve CONVERGED due to CONVERGED_HAPPY_BREAKDOWN, KSPReason=7, required 1 it...

all 11 QPSSolves from last QPSReset/QPSResetStatistics have required 103 iterations

tolerances: rtol=1.0e-05, abstol=3.6e-12, dtol=1.0e+50, maxits=10000

mpgp specific:

from the last QPSReset:

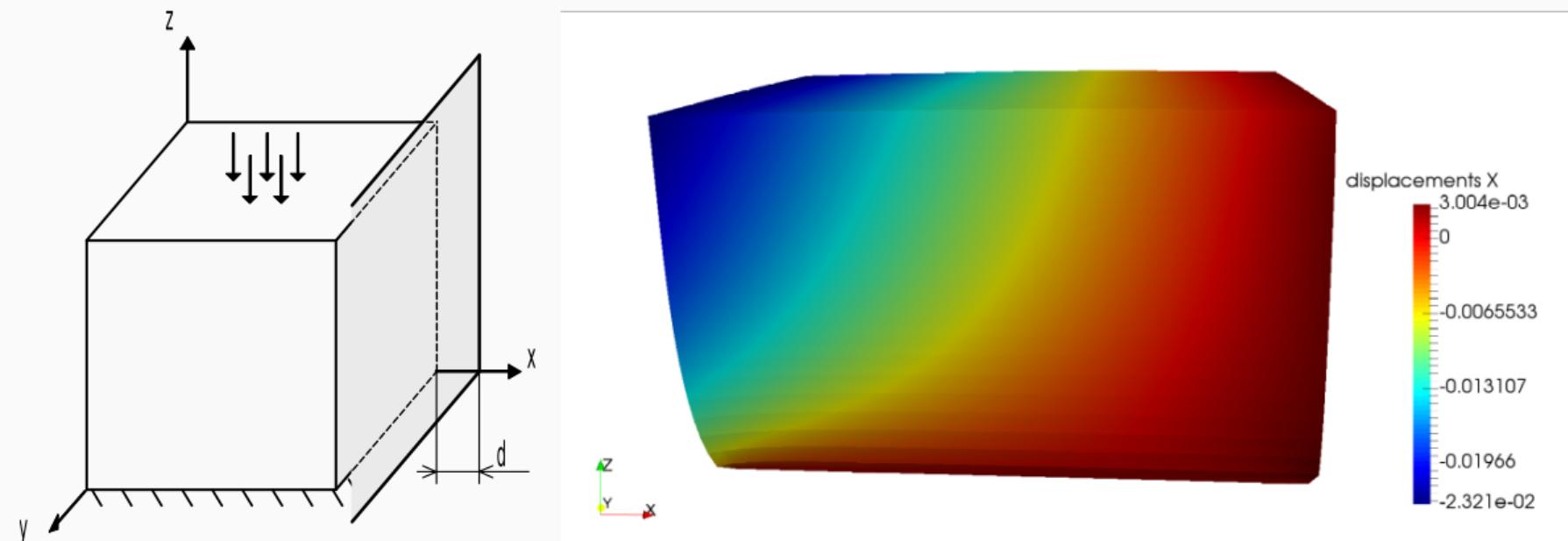
number of Hessian multiplications 119

number of CG steps 97

number of expansion steps 5

number of proportioning steps 1

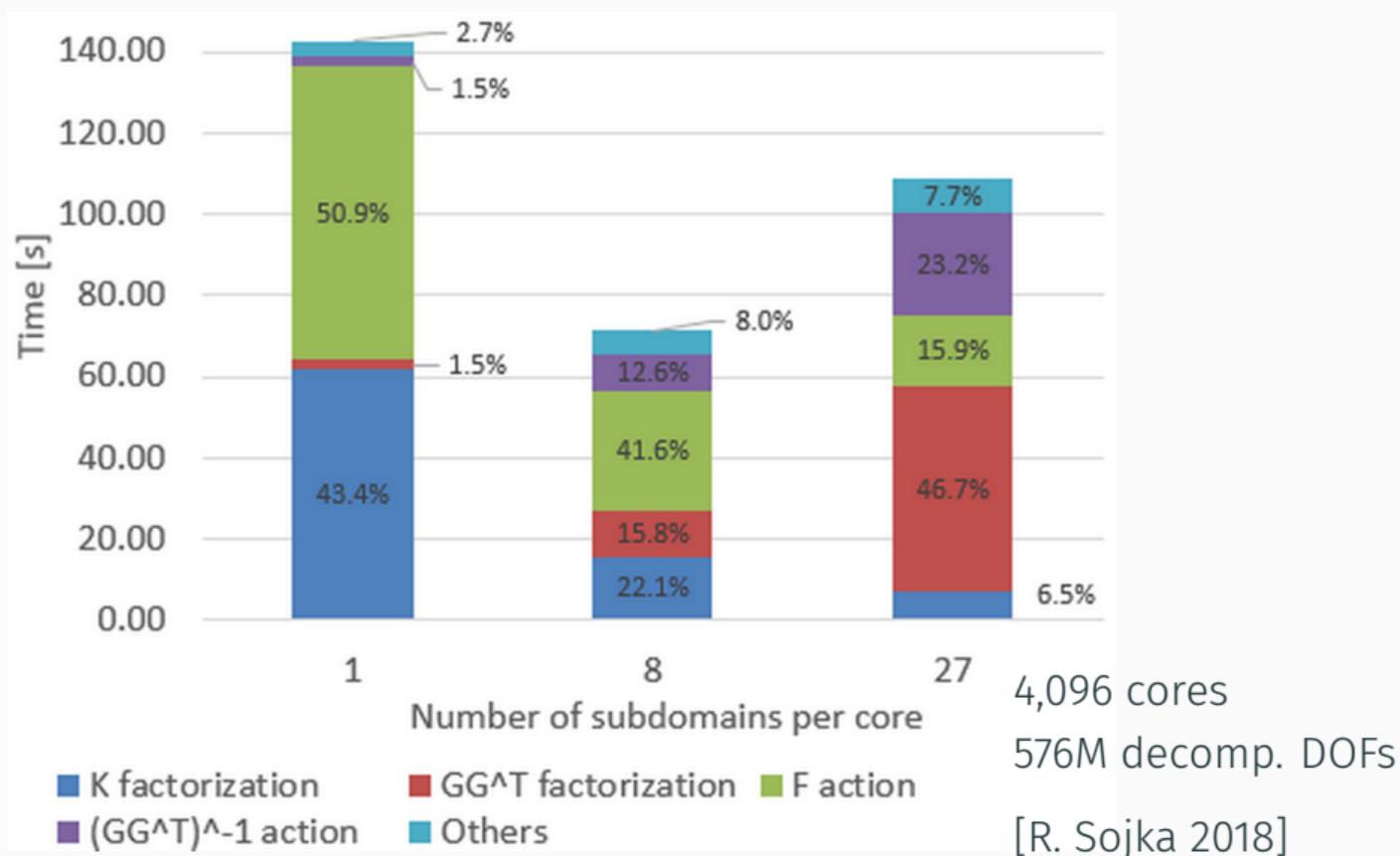
A Model Benchmark - Linear Elasticity (with contact)



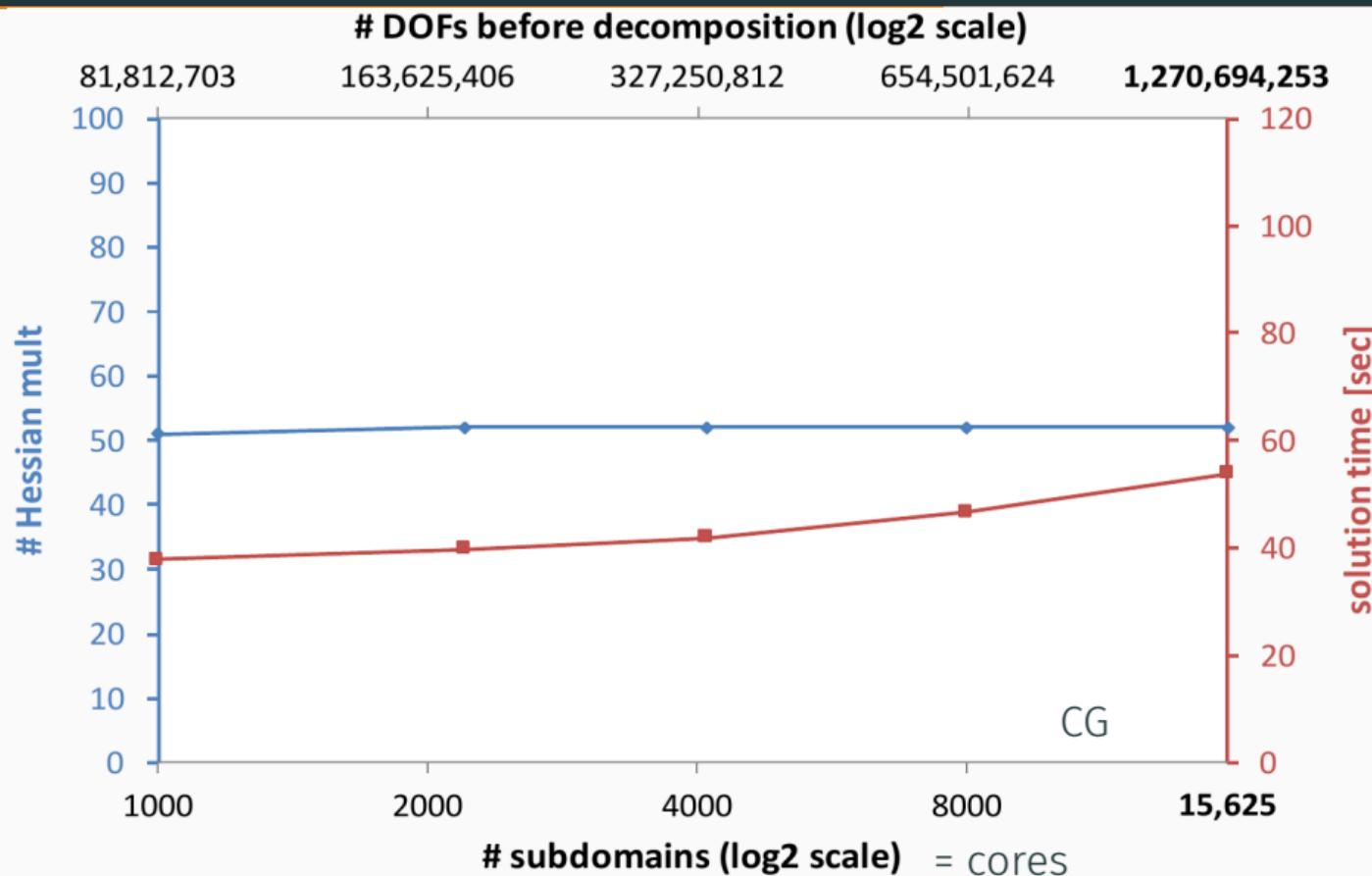
$$f_z = -465 \text{ N/mm}^2,$$

$$E = 2 \cdot 10^5 \text{ MPa}, \quad \mu = 0.33$$

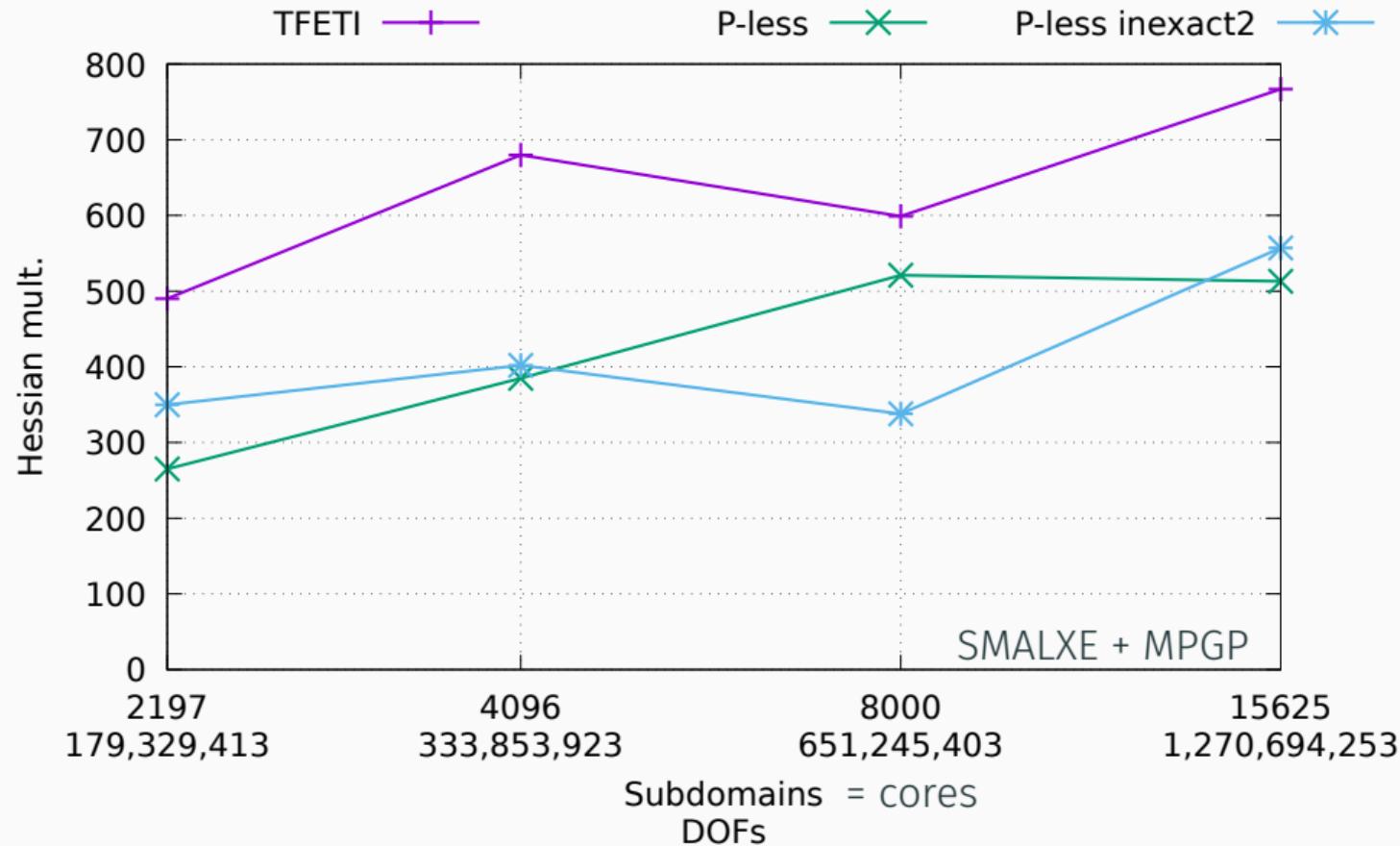
FETI Trade-off - Lin. Elasticity Cube (no contact) - ARCHER (EPCC)



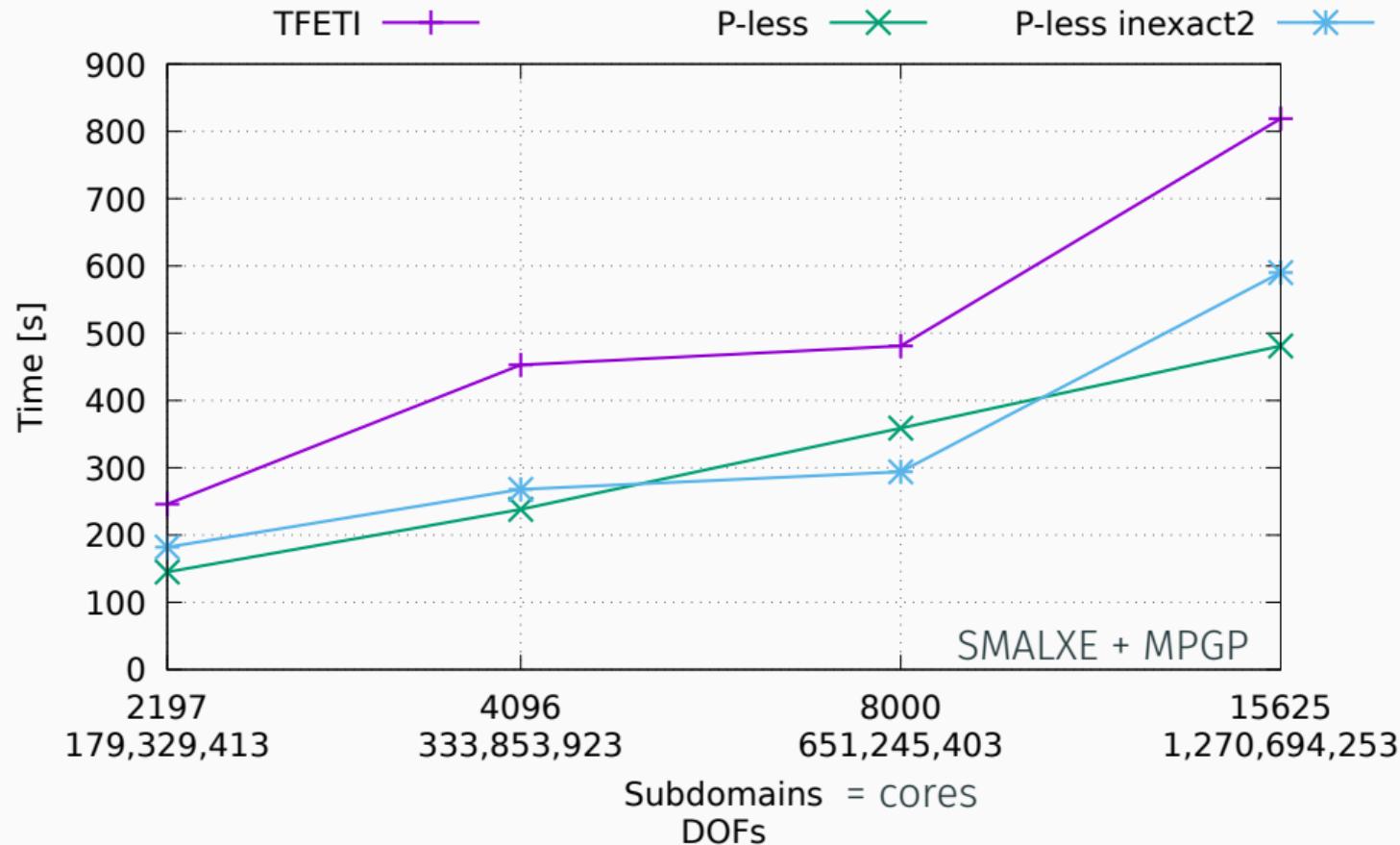
Weak Scaling, Linear Elasticity Problem @ Archer (EPCC)



Numerical Weak Scaling, Contact Problem @ Archer (EPCC)

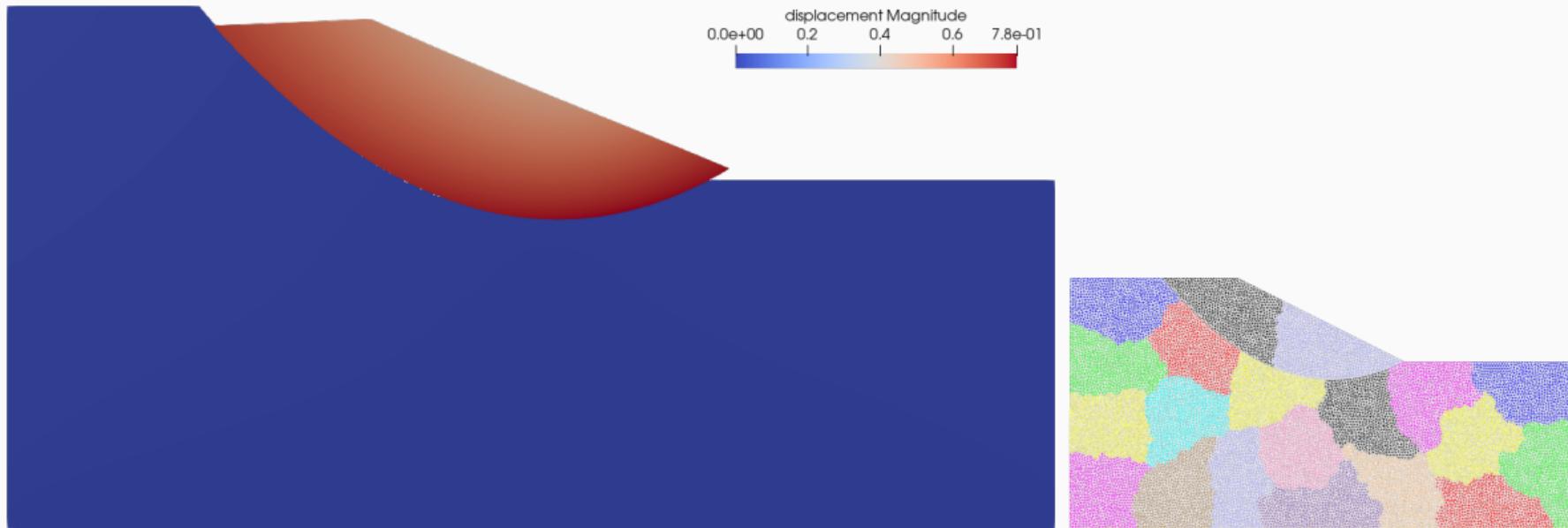


Weak Scaling, Contact Problem @ Archer (EPCC)



PERMON in Real World Applications

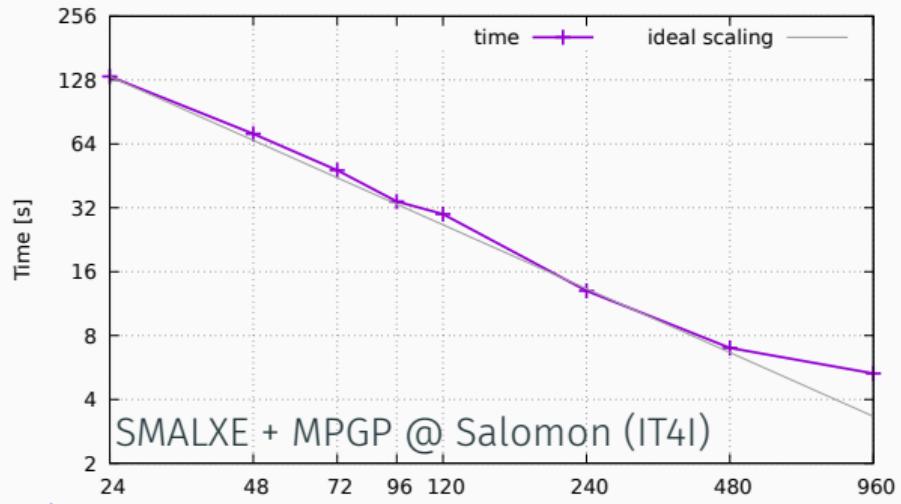
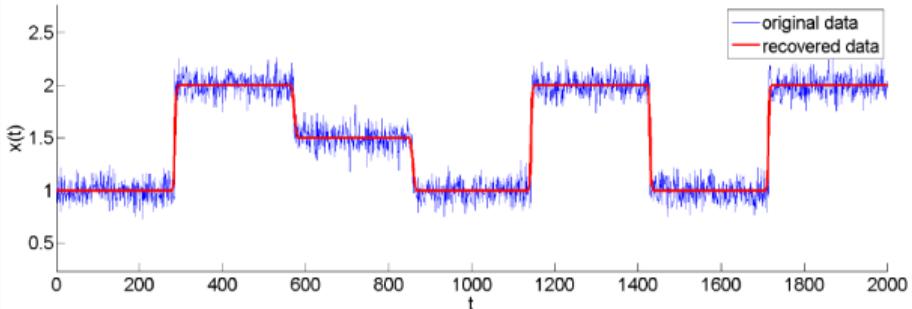
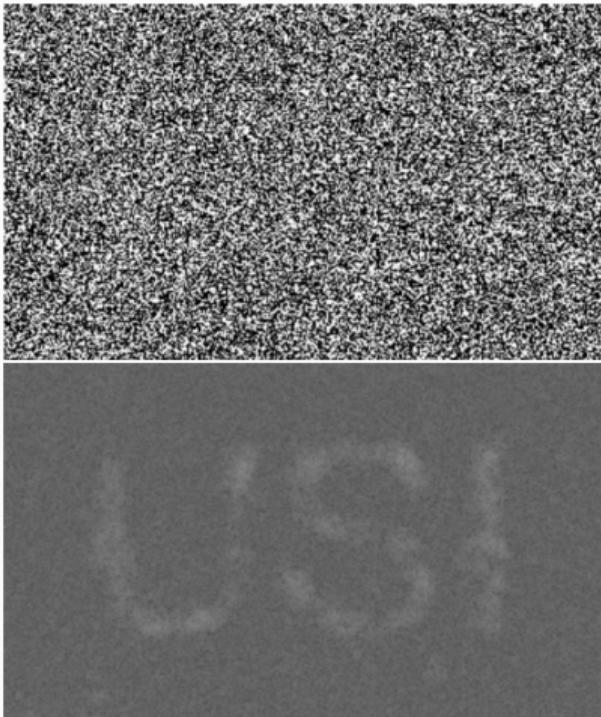
Slope Stability Problem Solved by FETI with Contact



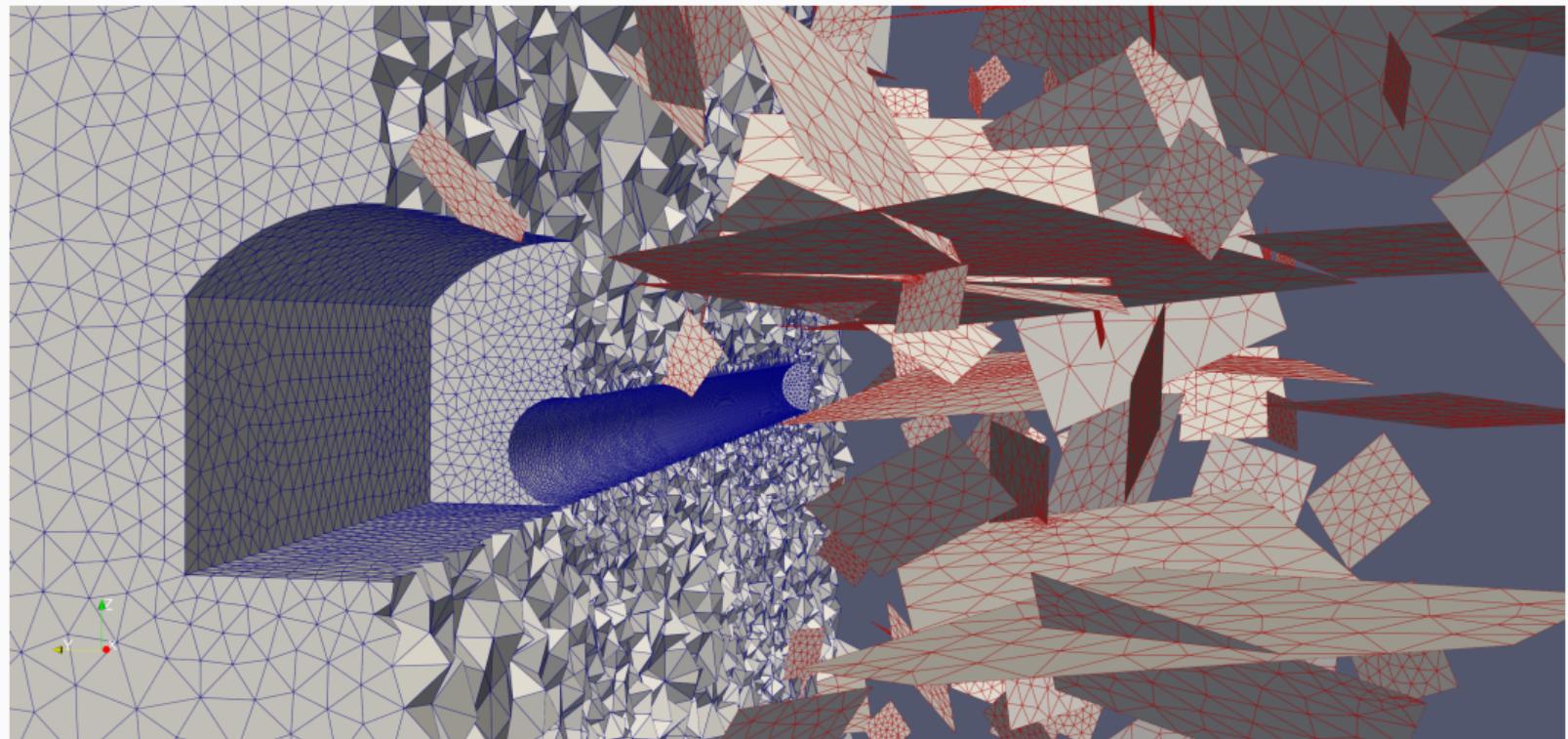
1. elasto-plastic analysis of a slope to determine failure surface
2. FETI to naturally model discontinuity in displacement field

Time Series De-noising in Economics

by Lukáš Pospíšil, USI



Borehole Excavation - Hydromechanical Model with Discrete Fracture Matrix

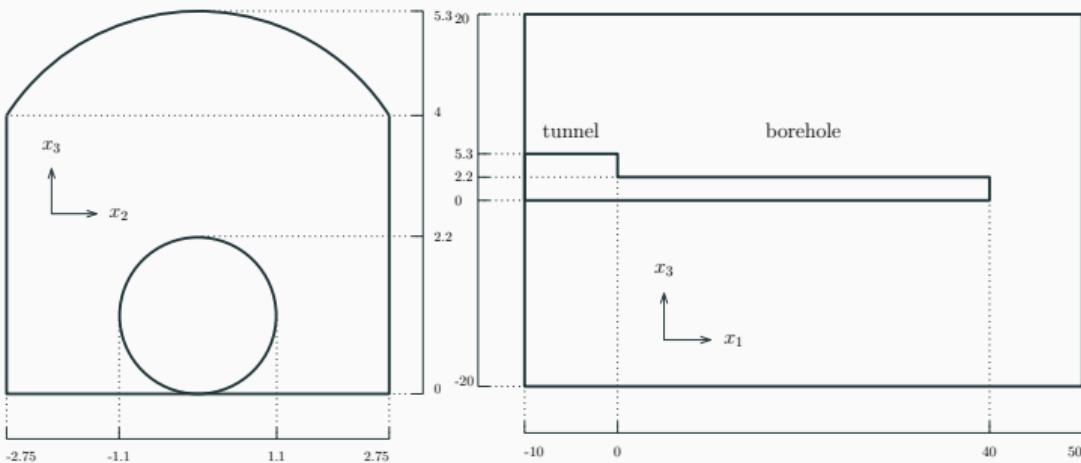
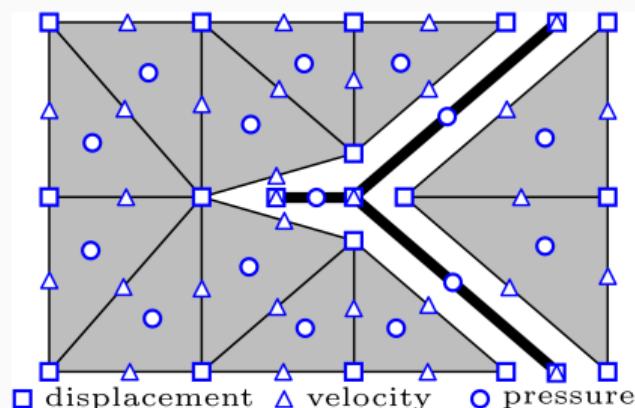


J. Stebel et al., *On the parallel solution of hydro-mechanical problems with fracture networks and contact conditions*, to appear in Computers & Structures

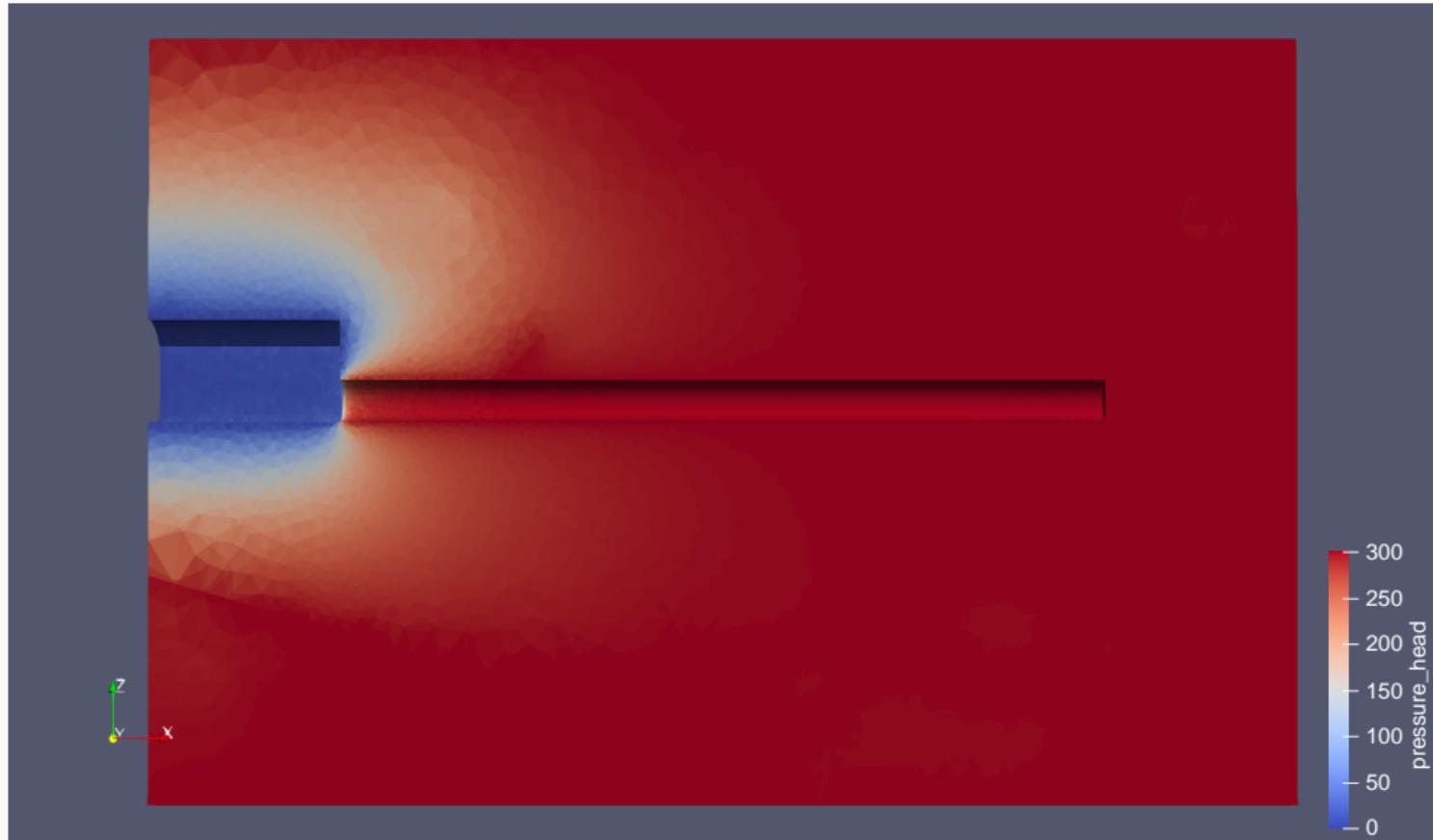
Borehole Excavation - Hydromechanical Model with Discrete Fracture Matrix

- using Flow123d - <https://flow123d.github.io>
- fractures are thin porous elastic medium satisfying impenetrability condition
- fixed-stress splitting method
 - hydraulic subproblem (CG + HYPRE BoomerAMG)
 - mechanical contact problem (dualization + MPRGP)
- depth 300 m, realistic model parameters

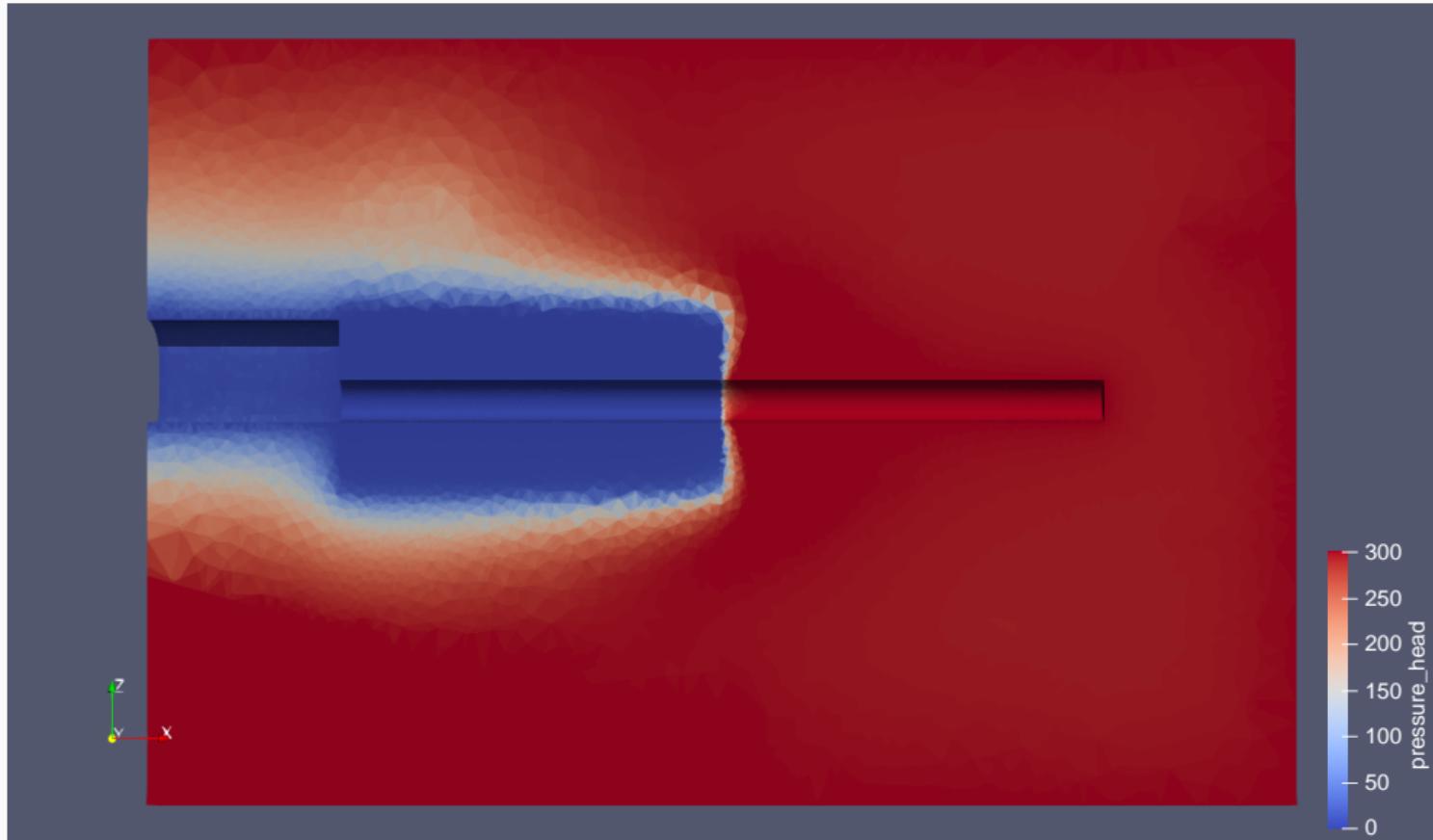
$$\Delta t = \begin{cases} 1, & t \in [0, 40) \\ 1 \rightarrow 15, & t \in [40, 120) \\ 15, & t \in [120, 360) \end{cases}$$



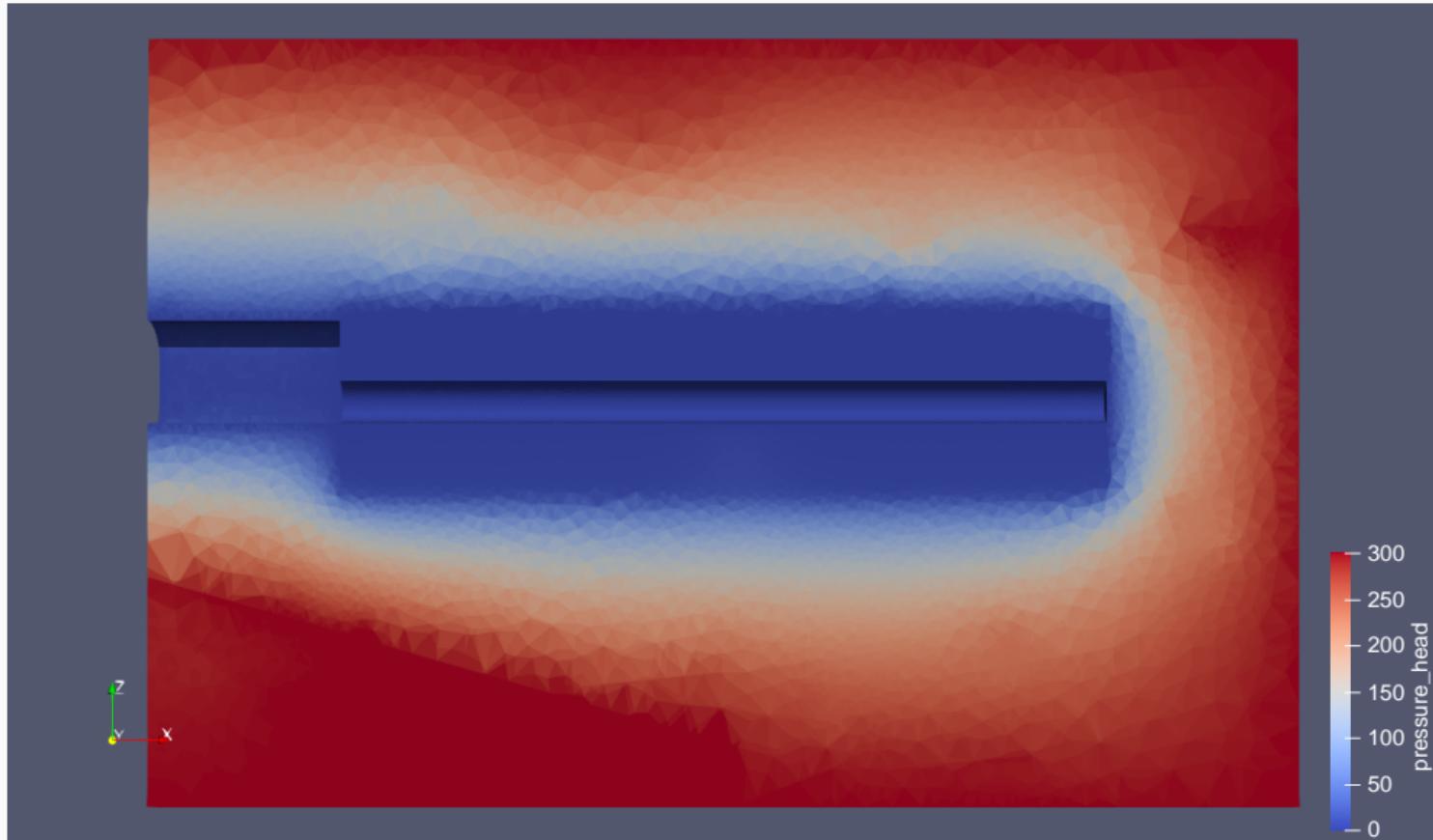
Borehole Excavation - 400 Fractures - Pressure Head [m] at $t = 0$



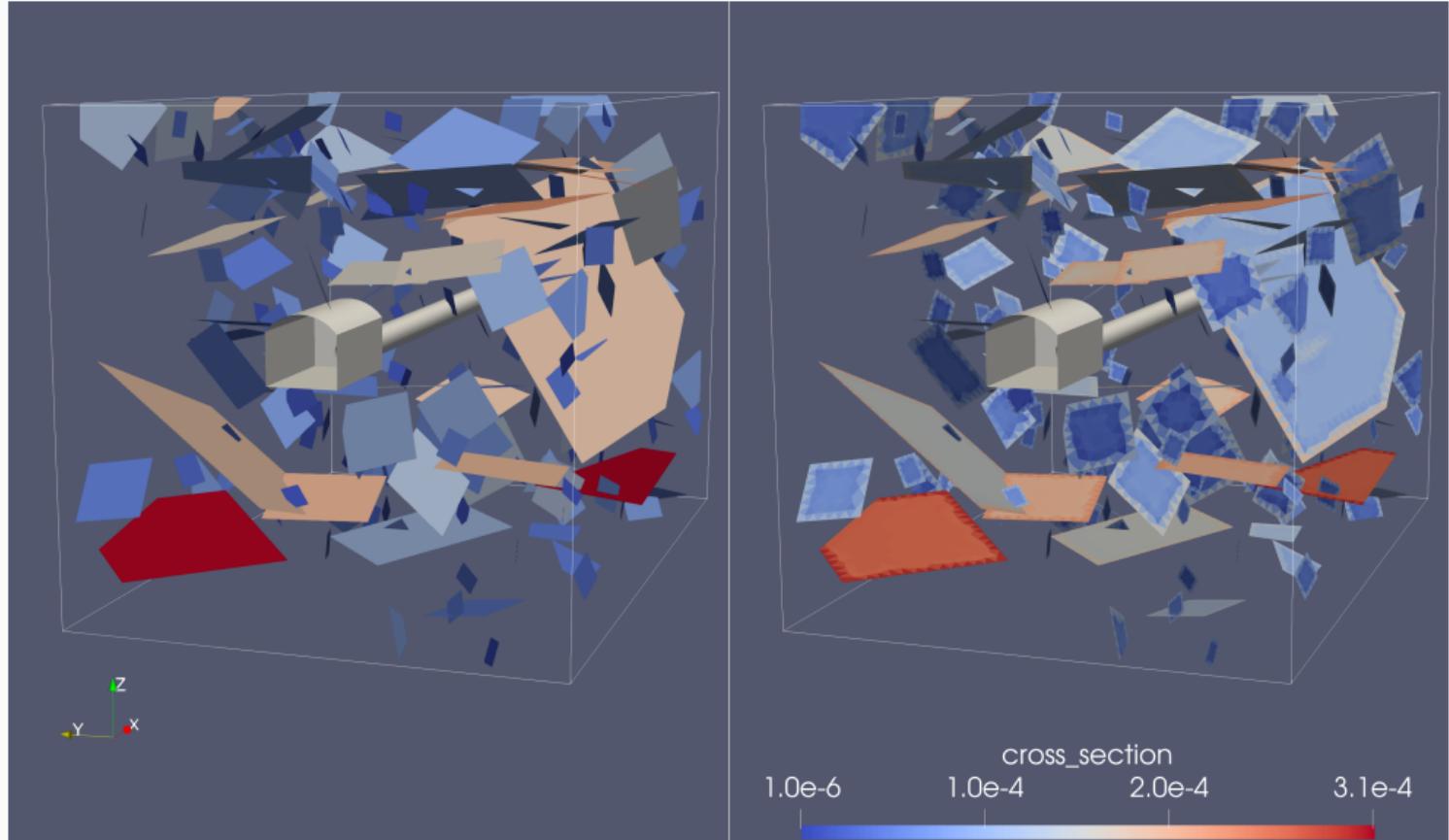
Borehole Excavation - 400 Fractures - Pressure Head [m] at $t = 20$



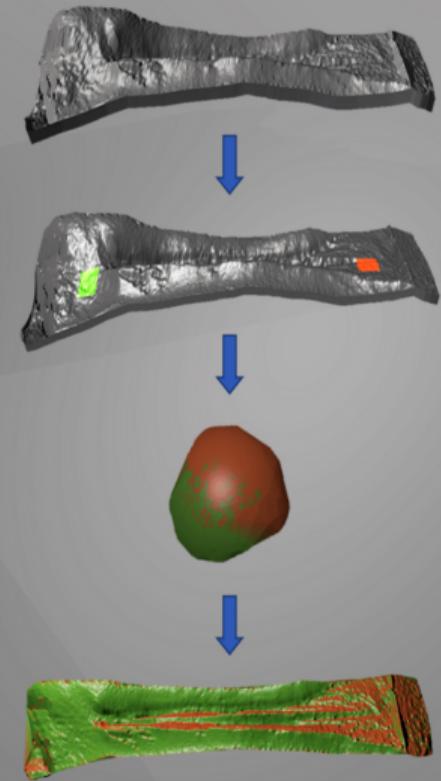
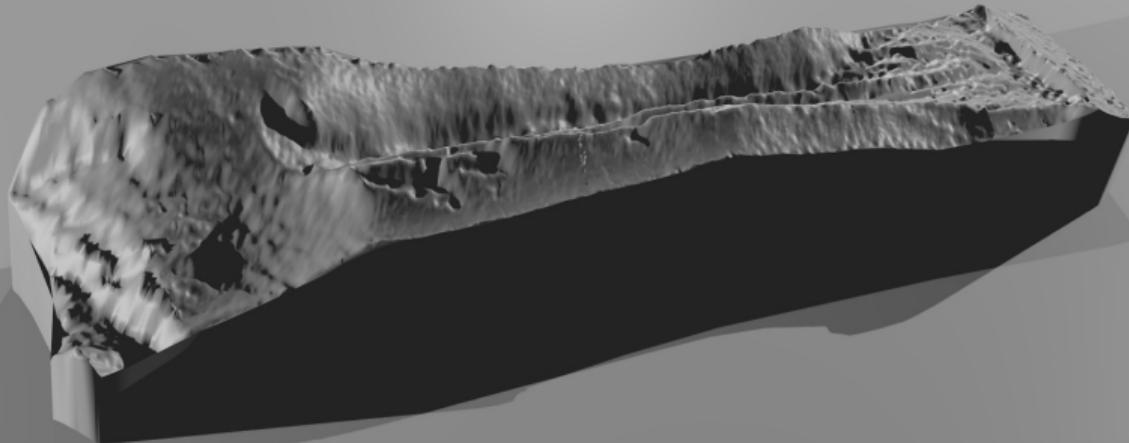
Borehole Excavation - 400 Fractures - Pressure Head [m] at $t = 360$



Borehole Excavation - 200 Fractures - Cross-section [m] at $t = 0$ and $t = 360$



Detecting Brittle and Ductile Fractures using PermonSVM



Ground truth learning
detecting the brittle and ductile fractures

Conclusion and Outlook

- Easy to use if you are familiar with PETSc
- Research tool for QP, FETI, SVM
- Solving real world applications

Conclusion and Outlook

- Easy to use if you are familiar with PETSc
- Research tool for QP, FETI, SVM
- Solving real world applications
- Planned improvements in the near future:
 - Documentation
 - **QPChainUpdate(...)** - propagate changes in input data through QP chain
 - Quadratic constraint QPC
 - SMALXE stopping criteria
 - Multiple subdomains per core and Hybrid FETI/BETI

Thank you for your attention!

Any questions?

Jakub Kruzik, Marek Pecha, David Horak, Zdenek Dostal, Martin Cermak, David Hrbac, Vaclav Hapla
jakub.kruzik@ugn.cas.cz



Institute of Geonics
Czech Academy of Sciences



Department
of Applied Mathematics

Department of applied mathematics
VSB - Technical University of Ostrava