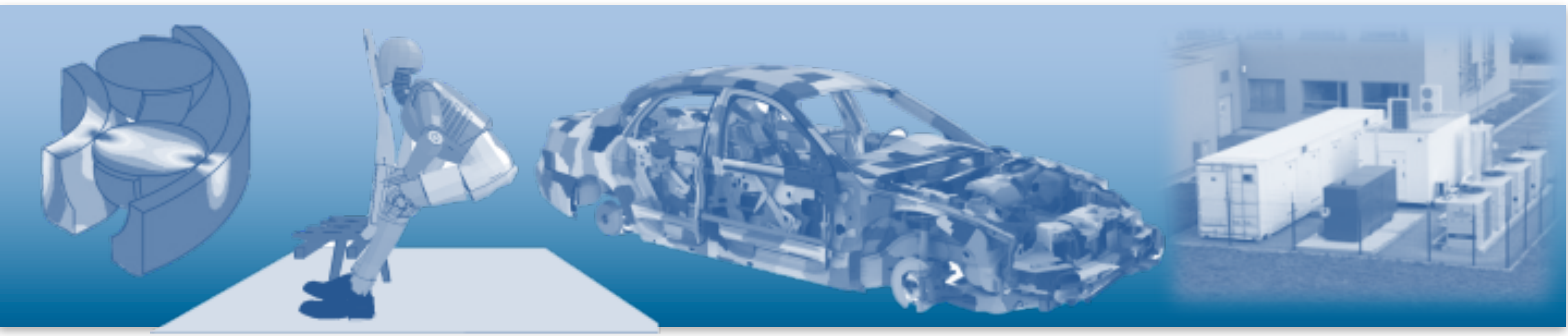
PERMON libraries for large scale quadratic programming

M. Čermák, V. Hapla, D. Horák, J. Kružík, L. Pospíšil*,
R. Sojka, A. Vašatová + prof. Z. Dostál



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

* Università della Svizzera italiana, Switzerland

IT4Innovations supercomputing for#industry01

Department of Applied Mathematics

permon.it4i.cz

IT4Innovations

- HPC infrastructure
- 8 research programmes

IT4Innovations
national111010
supercomputing
center1££0#110

2013

2014

2015

Anselm - 94 TFLOPs system operational since June 2013

Salomon – 2 PFLOPs system from June/July 2015

IT4Innovations
supercomputing
for#industry01

Department
of Applied
Mathematics

Salomon in numbers

- 1008 compute nodes
- 24,192 Intel Haswell cores
  - 2 x 12 per node
- 129,024 GB RAM
  - 128GB per node, 5.3GB per core
- 864 2nd generation MICs (KNC)
  - Intel Xeon Phi 7120P
  - 52,704 cores
- Rpeak 2.01 Pflop/s
- Rmax 1.46 Pflop/s (LINPACK)

largest European installation of MICs
⇒ interested in PETSc on MIC

IT4Innovations supercomputing for#industry01

Department of Applied Mathematics

PERMON.it4i.cz

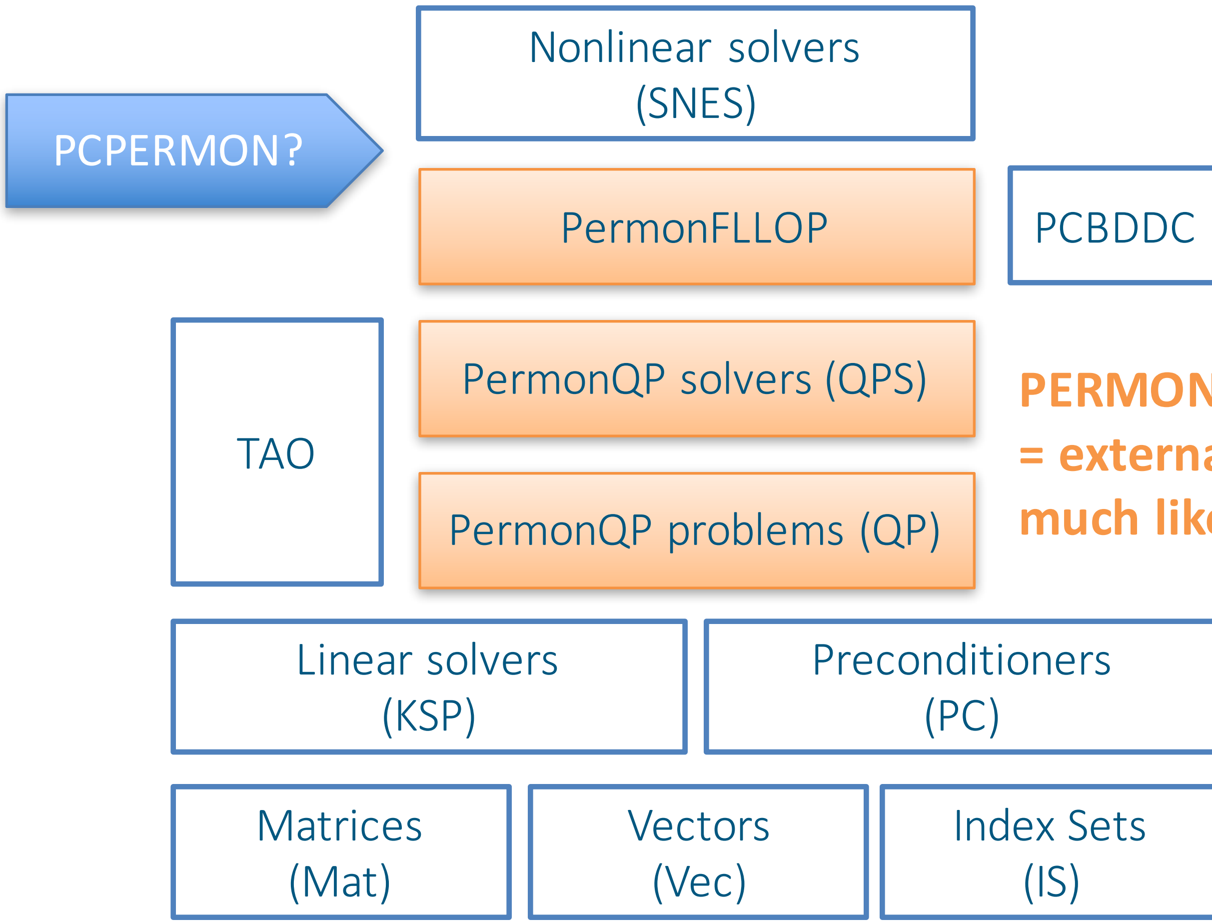
PERMON in a nutshell

Parallel, **E**fficient, **R**obust, **M**odular, **O**bject-oriented, **N**umerical

http://permon.it4i.cz/

- **PermonQP**
  - generic QP solution framework
  - QP problems, transforms, solvers
- **PermonFLLOP**
  - FETI DDM implementation
  - extends PermonQP
- **both**
  - based on/extending **PETSc** open-source framework for sci. comp.
  - together solve large contact mechanics problems
- started in 2011, ~30 000 lines of effective C code

PETSc + PERMON



PERMON
= external PETSc extension
much like SLEPc

IT4Innovations supercomputing for#industry01

Department of Applied Mathematics

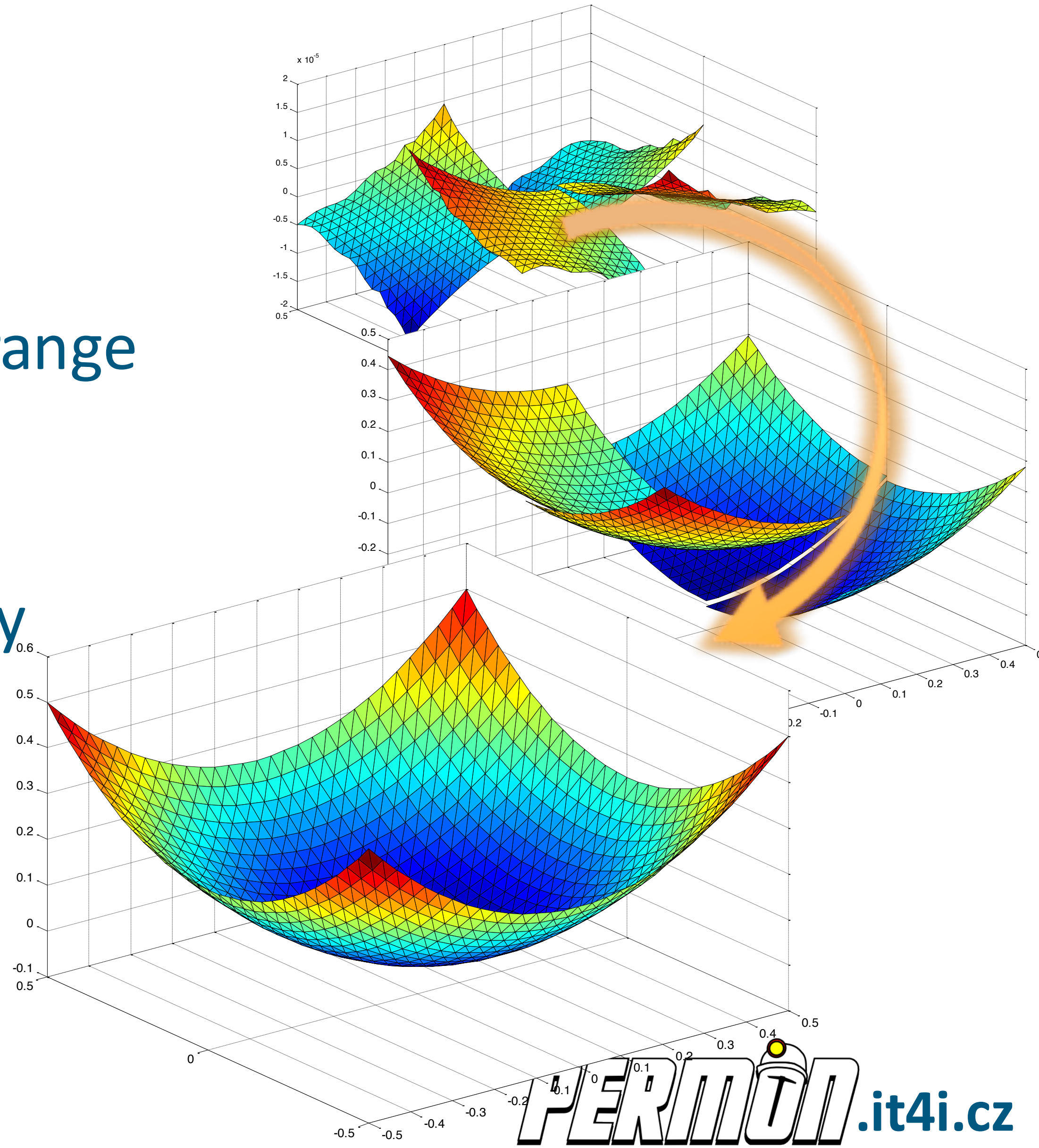
PERMON.it4i.cz

Build

- easy
- just build PETSc
- set PETSC_DIR and PETSC_ARCH
- set PERMON_DIR
- make

PermonFLLOP

- implements FETI non-overlapping DDM
- dual formulation
- unknown optimal Lagrange multipliers found by iterative process
- they enforce continuity across subdomains
- mix iterative and direct solvers



IT4Innovations
supercomputing
for#industry01

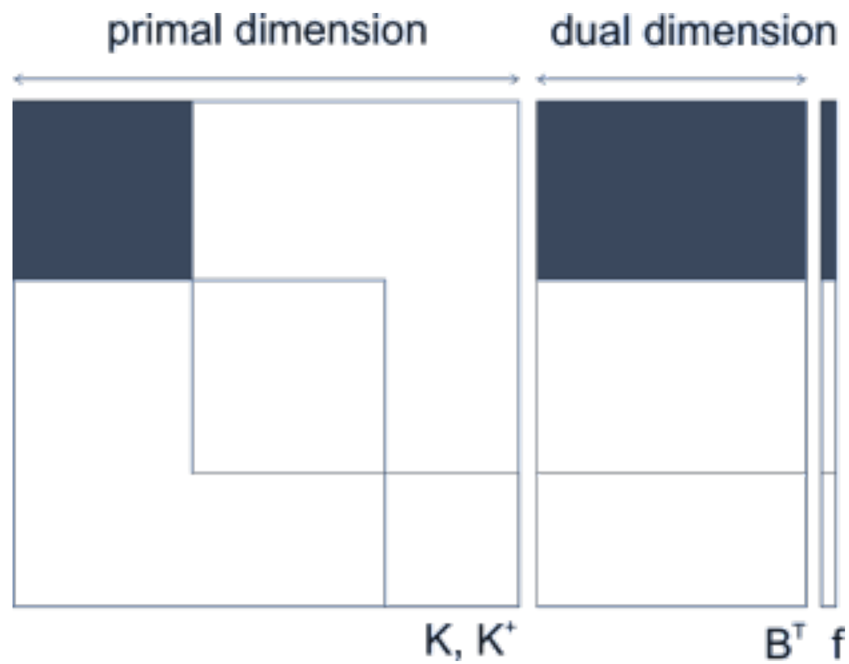
Department
of Applied
Mathematics

PERMON.it4i.cz

FETI: primal formulation

$$\begin{bmatrix} \mathbf{K} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{o} \end{bmatrix}$$

- $\mathbf{K}$ SPD block-diagonal stiffness matrix (blocks ~ subdomains)
- $\mathbf{R}$ columns span kernel of $\mathbf{K}$
- $\mathbf{f}$ load vector
- $\mathbf{B}$ signed boolean gluing matrix (can enforce also Dirichlet BC - TFETI)



IT4Innovations supercomputing for industry

Department of Applied Mathematics

PERMON.it4i.cz

FETI: QP formulations

- $\begin{bmatrix} \mathbf{K} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{o} \end{bmatrix}$ (primal)
- $\begin{bmatrix} \mathbf{BK}^+\mathbf{B}^T & -\mathbf{BR} \\ -\mathbf{R}^T\mathbf{B}^T & \mathbf{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{BK}^+\mathbf{f} \\ -\mathbf{R}^T\mathbf{f} \end{bmatrix}$ (dual)

- $\min_{\mathbf{u}} {}^1\!/_2\, \mathbf{u}^T\mathbf{Ku} - \mathbf{u}^T\mathbf{f}$ s.t. $\mathbf{Bu} = \mathbf{o}$ (primal)

  $\mathbf{F} = \mathbf{BK}^+\mathbf{B}^T,\ \mathbf{d} = \mathbf{BK}^+\mathbf{f},\ \mathbf{G} = -\mathbf{R}^T\mathbf{B}^T,\ \mathbf{e} = -\mathbf{R}^T\mathbf{f}$

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

  $\boldsymbol{\alpha} = -(\mathbf{GG}^T)^{-1}\mathbf{G}(\mathbf{F}\boldsymbol{\lambda} - \mathbf{d})$

- $\min_{\boldsymbol{\lambda}} {}^1\!/_2\, \boldsymbol{\lambda}^T\mathbf{F}\boldsymbol{\lambda} - \boldsymbol{\lambda}^T\mathbf{d}$ $s.t.$ $\mathbf{G}\boldsymbol{\lambda} = \mathbf{e}$ (dual)

FETI: improving the dual formulation

1) $\min_{\boldsymbol{\lambda}} {}^{1}/_{2}\, \boldsymbol{\lambda}^T \mathbf{F} \boldsymbol{\lambda} - \boldsymbol{\lambda}^T \mathbf{d}$ s.t. $\mathbf{G}\boldsymbol{\lambda} = \mathbf{e}$

$\boldsymbol{\lambda} = \overline{\boldsymbol{\lambda}} + \boldsymbol{\lambda}_{\mathrm{P}}$

$\boldsymbol{\lambda}_{\mathrm{P}} = \mathbf{G}^T (\mathbf{G}\mathbf{G}^T)^{-1} \mathbf{e} \in \mathrm{Im}\mathbf{G}^T$

$\bar{\mathbf{d}} = \mathbf{d} - \mathbf{F}\boldsymbol{\lambda}_{\mathrm{P}}$

$\overline{\boldsymbol{\lambda}} \in \mathrm{Ker}\mathbf{G}$ unknown

2) $\min_{\overline{\boldsymbol{\lambda}}} {}^{1}/_{2}\, \overline{\boldsymbol{\lambda}}^T \mathbf{F} \overline{\boldsymbol{\lambda}} - \overline{\boldsymbol{\lambda}}^T \bar{\mathbf{d}}$ s.t. $\mathbf{G}\overline{\boldsymbol{\lambda}} = \mathbf{o}$

$\mathbf{P} = \mathbf{I} - \mathbf{G}^T (\mathbf{G}\mathbf{G}^T)^{-1} \mathbf{G}$, [$(\mathbf{G}\mathbf{G}^T)^{-1}$ is coarse problem]

3) $\min_{\overline{\boldsymbol{\lambda}}} {}^{1}/_{2}\, \overline{\boldsymbol{\lambda}}^T \mathbf{PFP} \overline{\boldsymbol{\lambda}} - \overline{\boldsymbol{\lambda}}^T \mathbf{P}\bar{\mathbf{d}} \iff \mathbf{PFP}\overline{\boldsymbol{\lambda}} = \mathbf{P}$ [solve with CG]

FETI: inequality constraints

- **primal formulation**

  $\min_{\mathbf{u}} \, {}^1\!/_2 \, \mathbf{u}^T \mathbf{K} \mathbf{u} \; - \mathbf{u}^T \mathbf{f}$ s.t. $\mathbf{B}_{\mathrm{E}} \mathbf{u} = \mathbf{o}$ and $\mathbf{B}_{\mathrm{I}} \mathbf{u} \leq \mathbf{c}_{\mathrm{I}}$

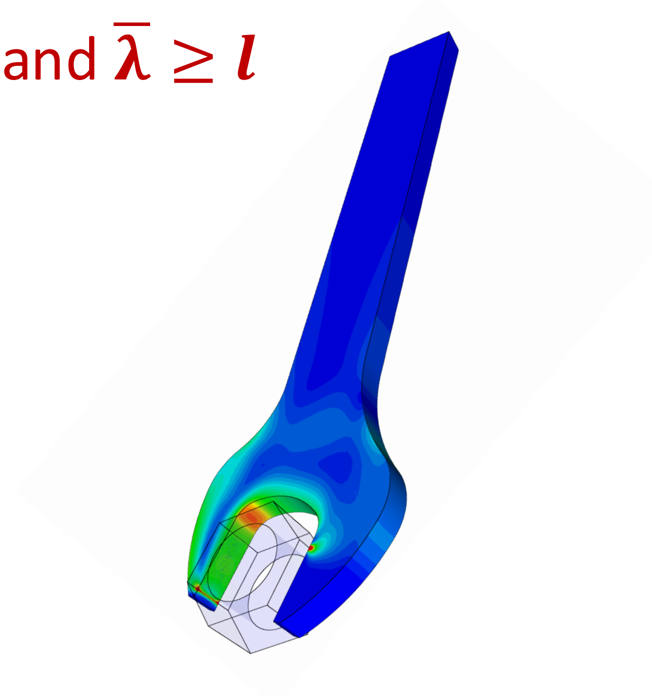
  (nonpenetration - mortar conditions)

- **dual formulation**

  $\min_{\overline{\boldsymbol{\lambda}}} \, {}^1\!/_2 \, \overline{\boldsymbol{\lambda}}^T \mathbf{P} \mathbf{F} \mathbf{P} \overline{\boldsymbol{\lambda}} - \overline{\boldsymbol{\lambda}}^T \mathbf{P} \overline{\mathbf{d}}$ s.t. $\mathbf{G} \overline{\boldsymbol{\lambda}} = \mathbf{o}$ and $\overline{\boldsymbol{\lambda}} \geq \boldsymbol{l}$

  (is no more a linear system !)

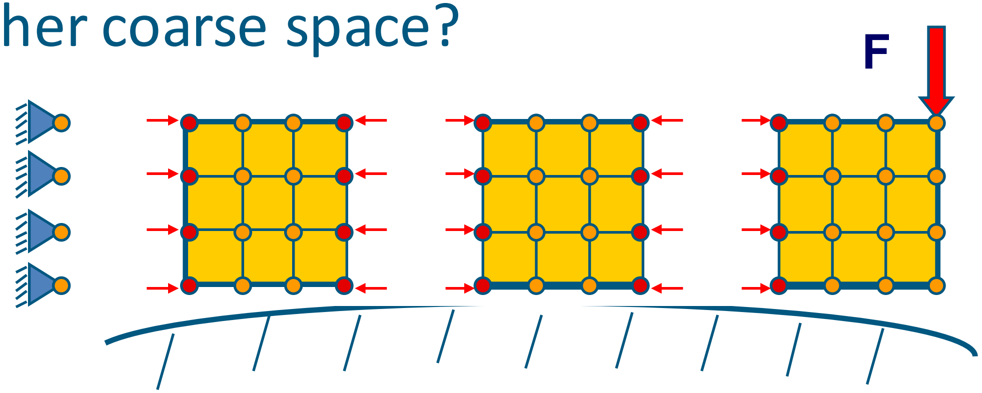
- use SMALXE + MPGP (discussed later)
  to solve this dual formulation



IT4Innovations supercomputing for#industry01
Department of Applied Mathematics
PERMON.it4i.cz

Total FETI (TFETI)

- **Dirichlet** boundary conditions handled **same way as gluing**
- all subdomain **floating**
  - all stiffness matrices singular
  - no need to detect it
  - same nullspaces in typical cases
  - uniform handling
- additional rows of **B** matrix
- effect of richer coarse space?



PermonFLLOP calls PermonQP

```
MPI_Comm comm;
Mat K, BE, BI;
Věc f, cE, cI, x;
QP qp;

QPCreate(comm, &qp);
QPSetOperator(qp, K);
QPSetRhs(qp, f);
QPSetEq(qp, BE, cE);
QPSetIneq(qp, BI, cI);

QPFetiPrepare(qp);

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

```
PetscErrorCode QPFetiPrepare(QP qp)
{
  …
  QPTDualize(qp);
  QPTHomogenizeEq(qp);
  QPTEnforceEqByProjector(qp);
  …
}
```

Sequence of QP transforms

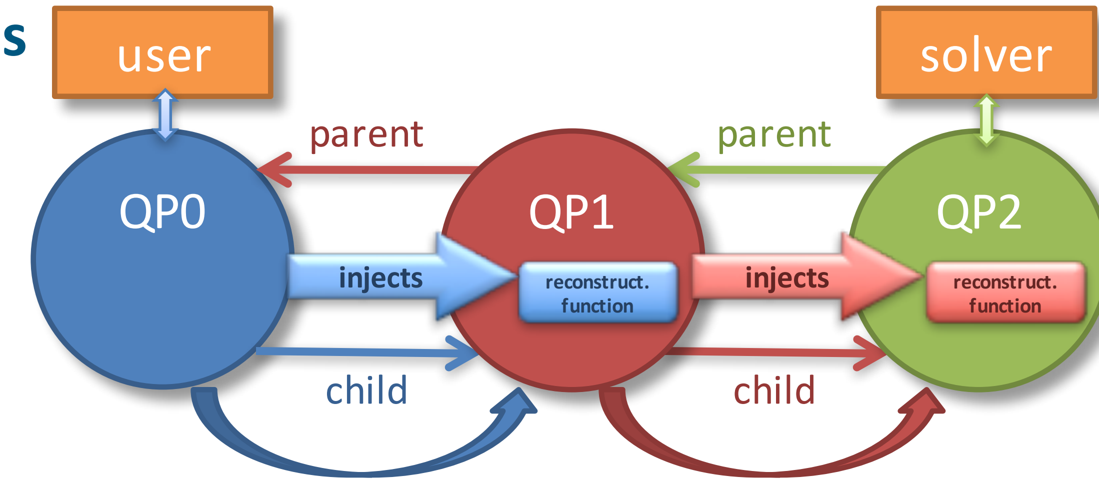
IT4Innovations supercomputing for#industry01

Department of Applied Mathematics

PERMON.it4i.cz

PermonQP

- QP solution **framework**
- **separation of concerns**
  - QP **problems** (QP)
  - QP **transforms** (QPT) and corresponding **backward reconstructions**
  - QP **solvers** (QPS)



- *"gradually get rid of difficulties in the original formulation"*
- modularity/composability
- automatic/manual **choice of solver**
- **applications:** contact problems, data fitting, support vector machines, ...

PermonQP and PETSc TAO

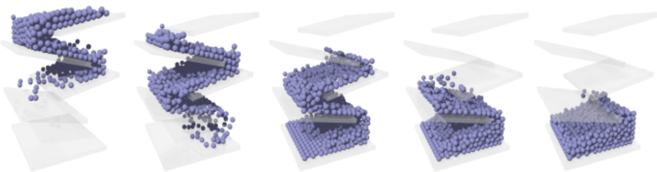
TAO

- general objective function
- only unconstrained and box constrained

PermonQP

- specific objective function (convex quadratic), Mat, Vec
- more like KSP, less like SNES
- can be coupled with FETI DDM (PermonFLLOP)
- more general constraints
  - linear equality
  - linear inequality
  - separable convex constraints – next slide
    - applicable to any subset of indices (IS)
    - simple bound / box constraints
    - quadratic constraints (spherical, elliptical)
    - conical constraints, …

Generalizing box constraints - QPC



- separable convex constraints
- applicable to any subset of indices (IS)
- simple bound / box constraints
- quadratic constraints (spherical, elliptical)
- conical constraints, …

$$\min \, {}^{1}\!/_{2}\, \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{b}$$
s.t. …

QPC
index set

QPCBOUND
lower bound
$\mathbf{x} \geq \boldsymbol{l}$

QPCBOX
lower bound
upper bound
$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$

QPCELLIPTIC
major axis
minor axis
$\left(\frac{x_1}{a}\right)^2 + \left(\frac{x_2}{b}\right)^2 \leq 1$

QPCQUADRATIC
radii
$x_1^2 + x_2^2 \leq r^2$

QPCCONICAL
angles
$x_1^2 + x_2^2 \leq \mu x_3^2$

PermonQP – print KKT satisfaction

```
====================
QPChainViewKKT output
QP Object: 4 MPI proce
  type: QP
  #0 in chain, derived
|| x|| = 4.75e+00    max( x) = 2.00e-06 =   x    )    min( x) = -3.23e-01 =
|| b|| = 4.75e-02    max( b) = -4.17e-04 =           min( b) = -2.50e-03 =
||cE|| = 0.00e-00    max(cE) = 0.00e-00 = cE(0)    min(cE) = 0.00e-00 = cE(
||cI|| = 1.63e+00    max(cI) = 4.34e-01 = cI(0)    min(cI) = 3.01e-01 = cI(
r = ||A*x - b + BE'*lambda_E + BI'*lambda_I|| = 6.78e-06    r0/||b|| = 1.43
r = ||BE*x-cE||             = 1.58e-05    r/||b|| = 3.32e-04
r = ||max(BI*x-cI,0)||      = 0.00e+00    r/||b|| = 0.00e+00
r = ||min(lambda_I,0)||     = 0.00e+00    r/||b|| = 0.00e+00
r = lambda_I'*(BI*x-cI)     = -1.39e-07    r/||b|| = -2.92e-06
------------------
QP Object: 4 MPI processes
  type: QP
  #1 in chain, derived by QPTScal
|| x|| = 4.75e+00    max( x) = 2.0
|| b|| = 4.75e-02    max( b) = -4.
||cE|| = 0.00e-00    max(cE) = 0.0
||cI|| = 1.63e+00    max(cI) = 4.3
r = ||A*x - b + BE'*lambda_E + BI'
r = ||BE*x-cE||             = 1.58e-0
r = ||max(BI*x-cI,0)||      = 0.00e+0
r = ||min(lambda_I,0)||     = 0.00e+0
r = lambda_I'*(BI*x-cI)     = -1.39e-
------------------
QP Object:(dual_) 4 MPI processes
  type: QP
  #2 in chain, derived by QPTDuali
|| x|| = 4.47e-01    max( x) = 1.3
|| b|| = 2.62e+00    max( b) = 3.9
||cE|| = 5.00e-01    max(cE) = -2.
||lb|| = inf    max(lb) = 0.00e+00
r = ||A*x - b + BE'*lambda_E - lam
r = ||BE*x-cE||             = 6.78e-0
r = ||min(x-lb,0)||         = 0.00e+0
r = ||min(lambda_lb,0)||    = 1.67e-05    r/||b|| = 6.37e-06
r = |lambda_lb'*(lb-x)|     = 2.47e-06    r/||b|| = 9.42e-07
------------------
```

-qp_chain_view_kkt

$$\min \tfrac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{b} \text{ s.t. } \mathbf{B}_E\mathbf{x} = \mathbf{c}_E \text{ and } \mathbf{x} \geq \boldsymbol{l}$$

$$\mathbf{A}\mathbf{x} - \mathbf{b} + \mathbf{B}_E^T\boldsymbol{\lambda}_E - \boldsymbol{\lambda}_l = \mathbf{o}$$

$$\mathbf{B}_E\mathbf{x} = \boldsymbol{c}_E$$

$$\mathbf{x} \geq \boldsymbol{l}$$

$$\boldsymbol{\lambda}_l \geq \mathbf{o}$$

$$\boldsymbol{\lambda}_l(\boldsymbol{l} - \mathbf{x}) = \mathbf{o}$$


```
QP Object:(dual_) 4 MPI processes
  type: QP
  #2 in chain, derived by QPTDualize
|| x|| = 4.47e-01    max( x) = 1.30e-01 =   x(112)    min( x) = -1.47e-01 =   x(106)
|| b|| = 2.62e+00    max( b) = 3.90e-01 =   b(21)    min( b) = -5.61e-01 =   b(133)
||cE|| = 5.00e-01    max(cE) = -2.50e-01 = cE(3)    min(cE) = -2.50e-01 = cE(0)
||lb|| = inf    max(lb) = 0.00e+00 = lb(30)    min(lb) = -4.49e+307 = lb(0)
r = ||A*x - b + BE'*lambda_E - lambda_lb|| = 5.31e-05    r0/||b|| = 2.03e-05
r = ||BE*x-cE||             = 6.78e-06    r/||b|| = 2.59e-06
r = ||min(x-lb,0)||         = 0.00e+00    r/||b|| = 0.00e+00
r = ||min(lambda_lb,0)||    = 1.67e-05    r/||b|| = 6.37e-06
r = |lambda_lb'*(lb-x)|     = 2.47e-06    r/||b|| = 9.42e-07
```

PermonQP – how to solve any QP?

- **Dualization**
  - crucial QP transform
  - new QP with **smaller dim**, **better conditioned** and **simpler constraints**

$$\min \tfrac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} - \mathbf{u}^T\mathbf{f} \text{ s.t. } \mathbf{B}_E\mathbf{u} = \mathbf{o} \text{ and } \mathbf{B}_I\mathbf{u} \le \mathbf{c}_I$$

$$\min \tfrac{1}{2}\boldsymbol{\lambda}^T\mathbf{A}\boldsymbol{\lambda} - \boldsymbol{\lambda}^T\mathbf{b} \text{ s.t. } \mathbf{C}\boldsymbol{\lambda} = \mathbf{d} \text{ and } \boldsymbol{\lambda} \ge \boldsymbol{l}$$

- **SMALXE**
  - **"pass-through"** solver taking care of **equality constraints**
    - moved to the Hessian matrix by means of penalty term
  - **aux. problem** with the rest of constraints solved by **inner solver**

$$\min \tfrac{1}{2}\boldsymbol{\lambda}^T\mathbf{A}\boldsymbol{\lambda} - \boldsymbol{\lambda}^T\mathbf{b} \text{ s.t. } \cancel{\mathbf{C}\boldsymbol{\lambda} = \mathbf{d}} \text{ and } \boldsymbol{\lambda} \ge \boldsymbol{l}$$

- **concrete solvers** for convex QP
  - **unconstrained** – PETSc KSP wrapper (QPSKSP)
  - **bound/box constraints**
  - MPGP, APGD, PBBf, SPG-QP, ...
  - PETSc TAO wrapper

$$\min \tfrac{1}{2}\boldsymbol{\lambda}^T\mathbf{A}\boldsymbol{\lambda} - \boldsymbol{\lambda}^T\mathbf{b} \text{ s.t. } \boldsymbol{\lambda} \ge \boldsymbol{l}$$

SMALXE for $\min_{\mathbf{Bx}=\mathbf{o}} \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{b}$ (SMALE)

- **Mainstream:** solve KKT system $\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{o} \end{bmatrix}$
  - using **direct/iterative linear system solver**
  - $\boldsymbol{\mu}$ is equality constraint Lagrange multiplier
  - inequality constrained → slacks, nonlin. eq. → **Interior Point Method**
- **SMALXE:** solve directly QP $\min \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{b}$ s.t. $\mathbf{Bx} = \mathbf{o}$
  - using **Augmented Lagrangians** and inner **iterative** solver
  - easier to extend for **inequality constraints**
  - equality and inequality constraints handled completely **separately**, **one after another**, in a specialized manner
  - **matrix structure agnostic** - both A and B may be even implicit

Lagrangian

$$L(\mathbf{x}, \boldsymbol{\mu}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} - \mathbf{x}^{\mathrm{T}}\mathbf{b} + \boldsymbol{\mu}^{\mathrm{T}}\mathbf{B}\mathbf{x}$$ (Lagrangian)

$$\mathbf{g}(\mathbf{x}, \boldsymbol{\mu}) = \mathbf{A}\mathbf{x} - \mathbf{b} + \mathbf{B}^{\mathrm{T}}\boldsymbol{\mu}$$ (Lagrangian gradient)

Augmented Lagrangian

$\tilde{L}(\mathbf{x},\boldsymbol{\mu},\rho) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{b} + \boldsymbol{\mu}^T\mathbf{B}\mathbf{x} + \frac{1}{2}\rho\|\mathbf{B}\mathbf{x}\|^2$ (augmented Lagrangian)

$\tilde{\mathbf{g}}(\mathbf{x},\boldsymbol{\mu},\rho) = \mathbf{A}\mathbf{x} - \mathbf{b} + \mathbf{B}^T\boldsymbol{\mu} + \rho\mathbf{B}^T\mathbf{B}\mathbf{x}$ (augmented Lagrangian gradient)

Note1: $\tilde{L}$ is Lagrangian of penalized problem

$$\min_{\mathbf{B}\mathbf{x}=\mathbf{o}} \frac{1}{2}\mathbf{x}^T(\mathbf{A} + \rho\mathbf{B}^T\mathbf{B})\mathbf{x} - \mathbf{x}^T\mathbf{b} \qquad \text{(QPP)}$$

$$\tilde{L}(\mathbf{x},\boldsymbol{\mu},\rho) = \frac{1}{2}\mathbf{x}^T(\mathbf{A} + \rho\mathbf{B}^T\mathbf{B})\mathbf{x} - \mathbf{x}^T(\mathbf{b} + \mathbf{B}\boldsymbol{\mu})$$

Note2: Solution $\hat{\mathbf{x}}$ of unconstrained penalized problem

$$\min \frac{1}{2}\mathbf{x}^T(\mathbf{A} + \rho\mathbf{B}^T\mathbf{B})\mathbf{x} - \mathbf{x}^T\mathbf{b} \qquad \text{(QPPU)}$$

approximates that of original QP: $\|\tilde{\mathbf{g}}\| \le \varepsilon\|\mathbf{b}\| \Longrightarrow \|\mathbf{B}\mathbf{x}\| \le \frac{1+\varepsilon}{\sqrt{\lambda_{\min}(\mathbf{A})\rho}}$

Exact aug. Lag. method for $\min_{\mathbf{Bx}=\mathbf{o}} \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Ax} - \mathbf{x}^{\mathrm{T}}\mathbf{b}$

(Method of multipliers)

initialize: $\mathbf{x}_0$ $[\mathbf{o}]$, $\rho_0 > 0$ $[2\|\mathbf{A}\|]$

$\boldsymbol{\mu}_0 = \mathbf{o}, \; k = 0$

while $\|\tilde{\mathbf{g}}(\mathbf{x}_k, \boldsymbol{\mu}_k, \rho_k)\| > \varepsilon\|\mathbf{b}\| \; \vee \; \|\mathbf{Bx}_k\| > \varepsilon\|\mathbf{b}\|$

$\quad \boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \rho_k \mathbf{Bx}_k$

$\quad \mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \tilde{L}(\mathbf{x}, \boldsymbol{\mu}_{k+1}, \rho_k)$ starting with $\mathbf{x}_k$

$\quad k = k + 1$

end

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^{\mathrm{T}}(\mathbf{A} + \rho_k \mathbf{B}^{\mathrm{T}}\mathbf{B})\mathbf{x} - \mathbf{x}^{\mathrm{T}}(\mathbf{b} + \mathbf{B}\boldsymbol{\mu}_{k+1})$$

- inner QP with penalized Hessian and updated RHS
- unconstrained QP ~ solving linear system
- solve with CG

Dostál, Z.:
Optimal Quadratic Programming Algorithms,
Springer Optimization and Its Applications,
2009, ISBN 978-0387848051

SMALXE for $\min_{\mathbf{Bx}=\mathbf{o}} \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} - \mathbf{x}^{\mathrm{T}}\mathbf{b}$ (SMALE)

initialize: $\mathbf{x}_0$ $[\mathbf{o}]$, $\beta > 0$ $[10]$, $M_0 > 0$ $[100\|\mathbf{A}\|]$, $\rho_0 > 0$ $[2\|\mathbf{A}\|]$, $\eta > 0$ $[0.1\|\mathbf{b}\|]$

$\boldsymbol{\mu}_0 = \mathbf{o}$, $k = 0$

while $\|\tilde{\mathbf{g}}(\mathbf{x}_k, \boldsymbol{\mu}_k, \rho_k)\| > \varepsilon\|\mathbf{b}\| \;\vee\; \|\mathbf{B}\mathbf{x}_k\| > \varepsilon\|\mathbf{b}\|$

$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \rho_k \mathbf{B}\mathbf{x}_k$

find $\mathbf{x}_{k+1}$ such that $\|\tilde{\mathbf{g}}(\mathbf{x}_{k+1}, \boldsymbol{\mu}_{k+1}, \rho_k)\| \leq \min(M_k\|\mathbf{B}\mathbf{x}_{k+1}\|, \eta)$

if $\tilde{L}(\mathbf{x}_{k+1}, \boldsymbol{\mu}_{k+1}, \rho_k) > \tilde{L}(\mathbf{x}_k, \boldsymbol{\mu}_k, \rho_{k-1}) + \frac{1}{2}\rho_k\|\mathbf{B}\mathbf{x}_{k+1}\|^2$

$\rho_{k+1} = \beta\rho_k$ or $M_{k+1} = M_k/\beta$

SMALE or SMALE-M

end

$k = k + 1$

end

again solving $\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^{\mathrm{T}}(\mathbf{A} + \rho_k\mathbf{B}^{\mathrm{T}}\mathbf{B})\mathbf{x} - \mathbf{x}^{\mathrm{T}}(\mathbf{b} + \mathbf{B}\boldsymbol{\mu}_{k+1})$

BUT with early termination

- inner QP with penalized Hessian and updated RHS
- unconstrained QP ~ solving linear system
- solve with e.g. CG

SMALXE for $\min_{\substack{\mathbf{Bx}=\mathbf{o} \\ \mathbf{x} \geq \boldsymbol{l}}} \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Ax} - \mathbf{x}^{\mathrm{T}}\mathbf{b}$ (SMALBE)

initialize: $\mathbf{x}_0$ $[\mathbf{o}]$, $\beta > 0$ $[10]$, $M_0 > 0$ $[100\|\mathbf{A}\|]$, $\rho_0 > 0$ $[2\|\mathbf{A}\|]$, $\eta > 0$ $[0.1\|\mathbf{b}\|]$

$\boldsymbol{\mu}_0 = \mathbf{o}$, $k = 0$

$\mathbf{g}^P$ ... projected gradient – see next slide

while $\|\tilde{\mathbf{g}}^P(\mathbf{x}_k, \boldsymbol{\mu}_k, \rho_k)\| > \varepsilon\|\mathbf{b}\| \vee \|\mathbf{Bx}_k\| > \varepsilon\|\mathbf{b}\|$

$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \rho_k \mathbf{Bx}_k$

find $\mathbf{x}_{k+1} \geq \boldsymbol{l}$ such that $\|\tilde{\mathbf{g}}^P(\mathbf{x}_{k+1}, \boldsymbol{\mu}_{k+1}, \rho_k)\| \leq \min(M_k\|\mathbf{Bx}_{k+1}\|, \eta)$

if $\tilde{L}(\mathbf{x}_{k+1}, \boldsymbol{\mu}_{k+1}, \rho_k) > \tilde{L}(\mathbf{x}_k, \boldsymbol{\mu}_k, \rho_{k-1}) + \frac{1}{2}\rho_k\|\mathbf{Bx}_{k+1}\|^2$

$\rho_{k+1} = \beta\rho_k$ or $M_{k+1} = M_k/\beta$

SMALBE or SMALBE-M

end

now solving $\min_{\mathbf{x} \geq \boldsymbol{l}} \frac{1}{2}\mathbf{x}^{\mathrm{T}}(\mathbf{A} + \rho_k\mathbf{B}^{\mathrm{T}}\mathbf{B})\mathbf{x} - \mathbf{x}^{\mathrm{T}}(\mathbf{b} + \mathbf{B}\boldsymbol{\mu}_{k+1})$

but using the same $\tilde{L}$ corresponding to $\min_{\mathbf{Bx}=\mathbf{o}} \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Ax} - \mathbf{x}^{\mathrm{T}}\mathbf{b}$

$k = k + 1$

end

IT4Innovations supercomputing for industry

Department of Applied Mathematics

Dostál, Z.: Optimal Quadratic Programming Algorithms, Springer Optimization and Its Applications, 2009, ISBN 978-0387848051

PERMON.it4i.cz

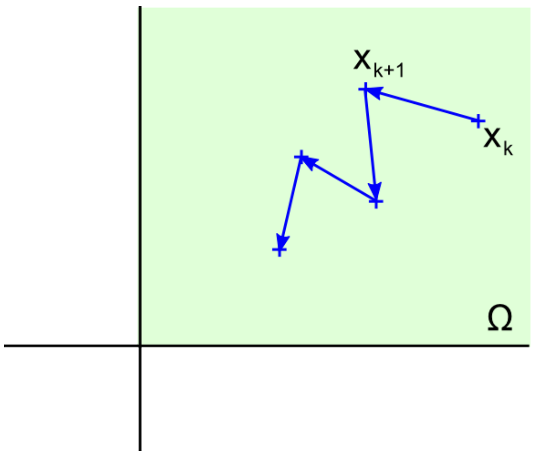
Projected gradient

- $\tilde{\mathbf{g}}^P = [g_i^P]$, $\tilde{\mathbf{g}} = [g_i]$, $\mathbf{x} = [x_i]$, $\boldsymbol{l} = [l_i]$
- M … set of indices of inequality constrained entries of $\mathbf{x}$

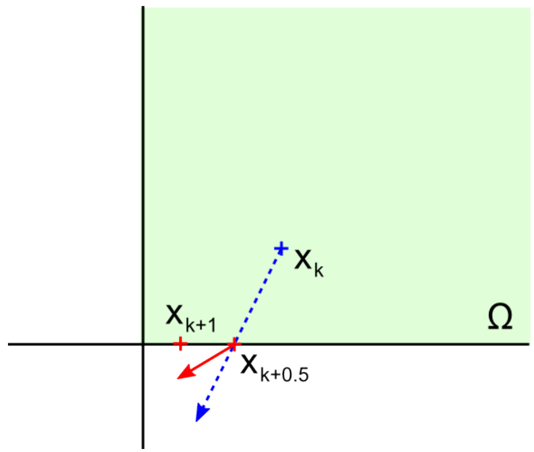
$$g_i^P = \begin{cases} g_i & \text{for} \quad x_i > l_i \ \vee \ i \notin M \\ \min(g_i, 0) & \text{for} \quad x_i = l_i \ \wedge \ i \in M \end{cases}$$

Dostál, Z.:
Optimal Quadratic Programming Algorithms,
Springer Optimization and Its Applications,
2009, ISBN 978-0387848051

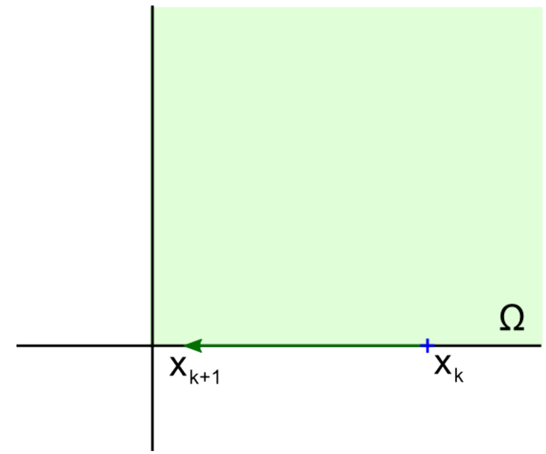
MPGP (Modified Proportioning with Gradient Projection)



CG-step



CG-halfstep
+ expansion step
- expand active set



optimal step on active set
(proportioning step)
- reduce active set

IT4Innovations
supercomputing
for#industry01

Department
of Applied
Mathematics

Dostál Z., Pospíšil L.: Minimization of the quadratic function with semidefinite Hessian subject to the bound constraints, Computers and Mathematics with Applications, (2015).

MPGP (Modified Proportioning with Gradient Projection)

Choose $x_0 \in \Omega$, $\overline{\alpha} \in (0, 2/\lambda_{\max})$,
for $k = 0, 1, 2, \ldots$ **do** (while $\|g^P(x_k)\|$ is not small)
if $\|\varphi(x_k)\| >> \|\beta(x_k)\|$ (proportioning condition)
CG-step or *CG-halfstep*
- do CG-step for the solution of the problem on free set
- if this means leaving Ω,
do only feasible step, expansion step and restart CG

else
Proportioning step
- do optimal step to solve problem on active set, restart CG

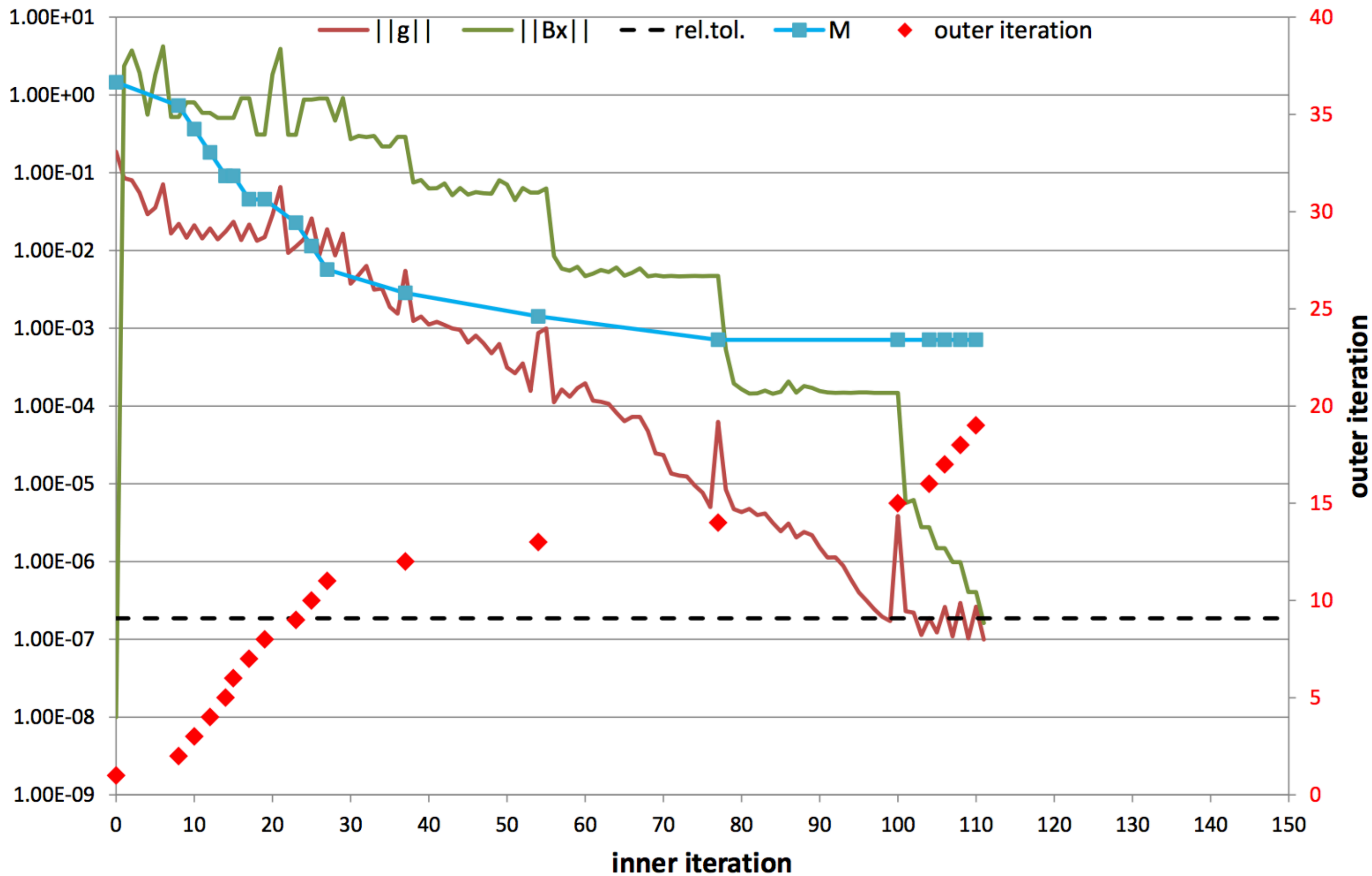
endif
$k = k + 1$
endfor
$\bar{x} \approx x_k$

Dostál Z., Pospíšil L.: Minimization of the quadratic function with semidefinite Hessian subject to the bound constraints, Computers and Mathematics with Applications, (2015).

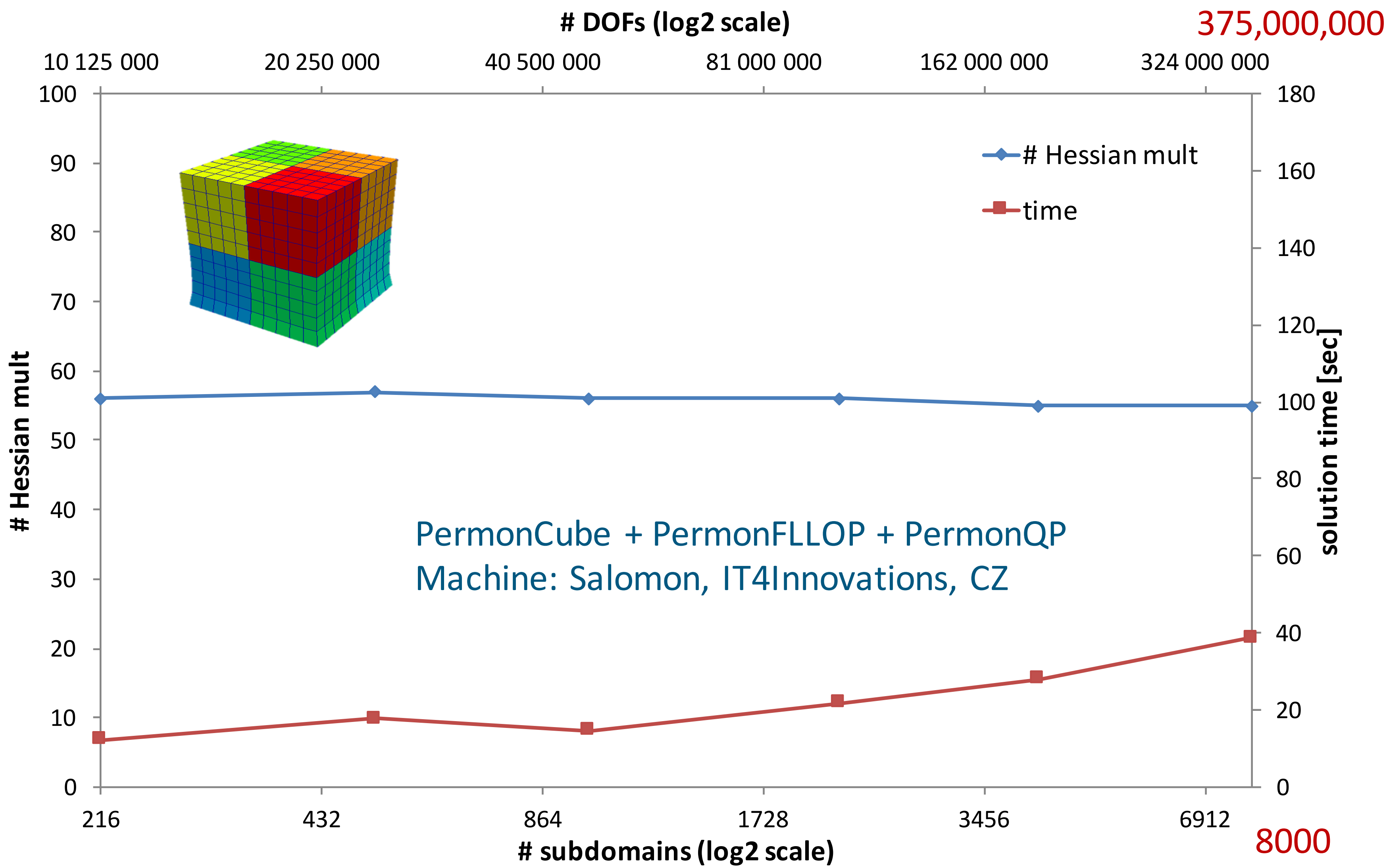
SMALXE+MPGP iteration progress

Inner stopping criterion: $\|\tilde{\mathbf{g}}^P\| \leq \min(M\|\mathbf{Bx}_k\|, \eta) \ \wedge \|\tilde{\mathbf{g}}^P\| < \varepsilon$

Start to increase penalty ρ by factor of 2 each outer iteration once $\|\tilde{\mathbf{g}}^P\| < \varepsilon$ is reached



Experiments – linear elasticity

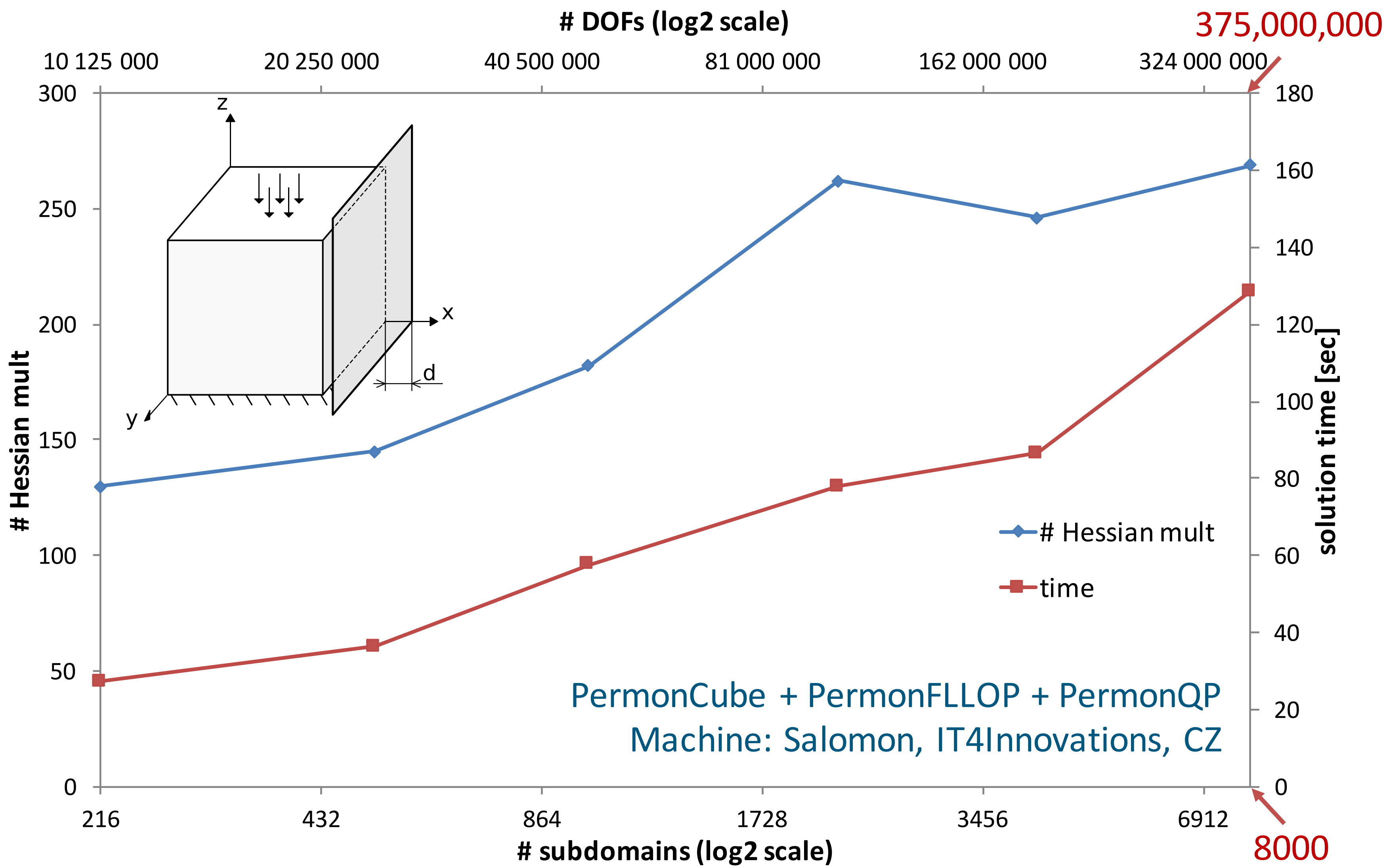


IT4Innovations supercomputing for#industry01

Department of Applied Mathematics

PERMON.it4i.cz

Experiments – contact elasticity

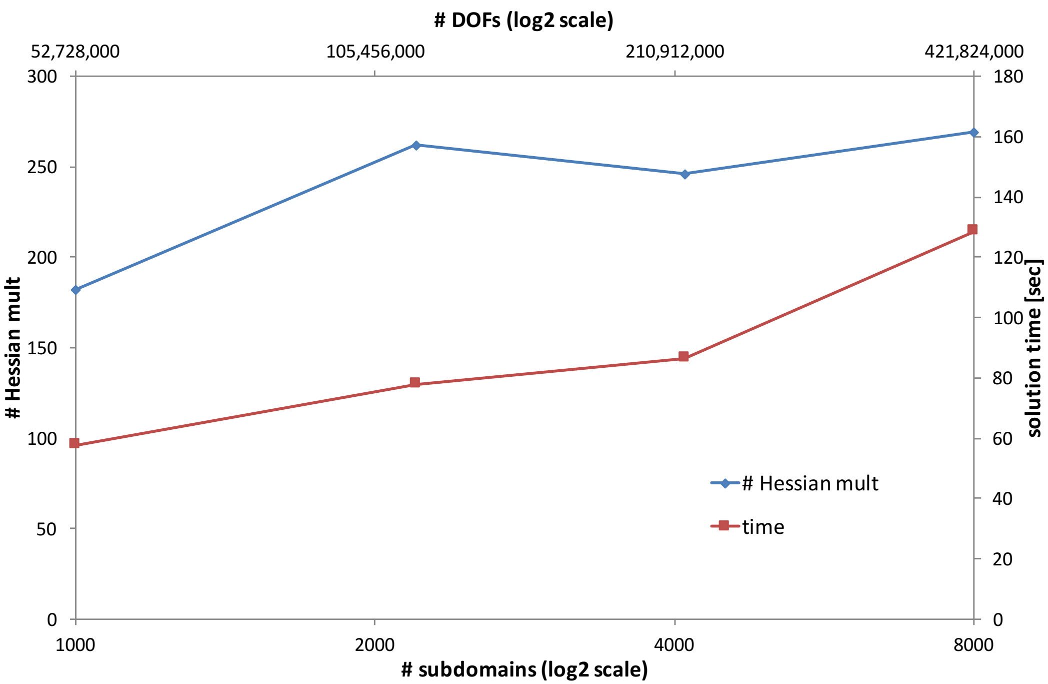


IT4Innovations supercomputing for industry

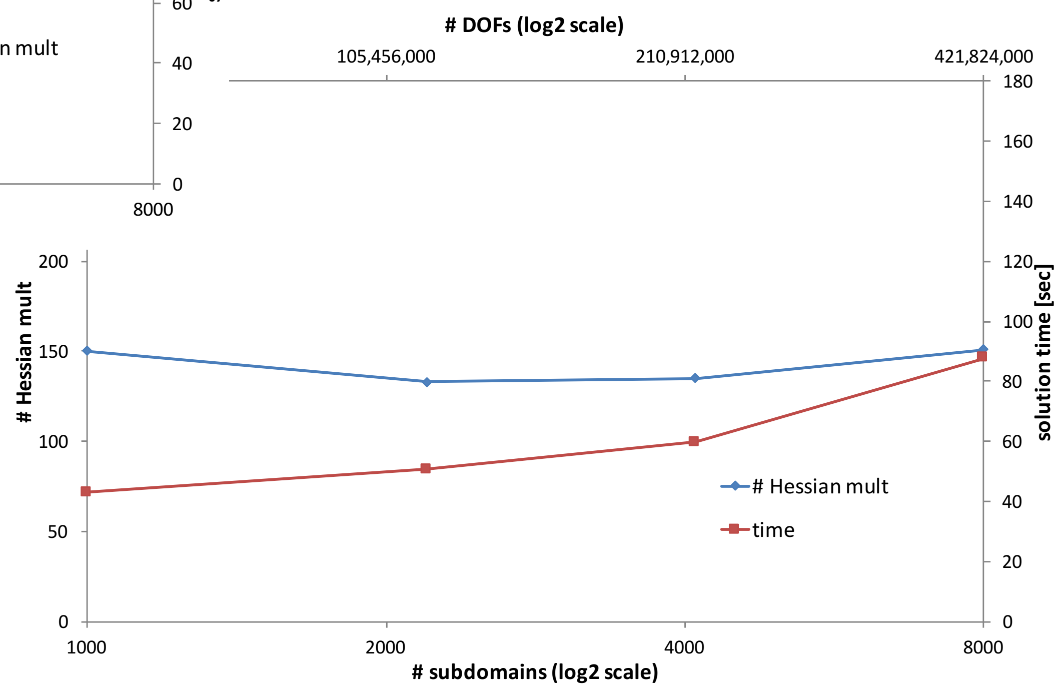
Department of Applied Mathematics

PERMON.it4i.cz

Very recent improvement



adaptive length
of expansion step
(computed yesterday)



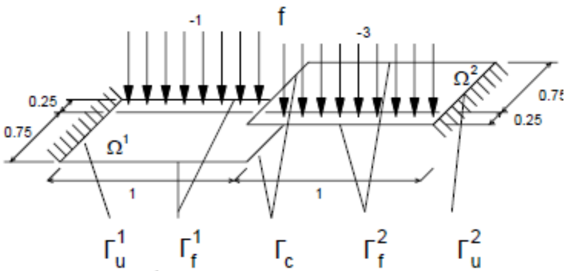
IT4Innovations
supercomputing
for#industry01

Department
of Applied
Mathematics

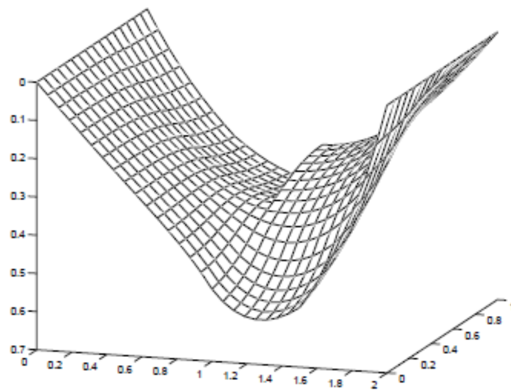
Coercive and semicoercive variational inequalities

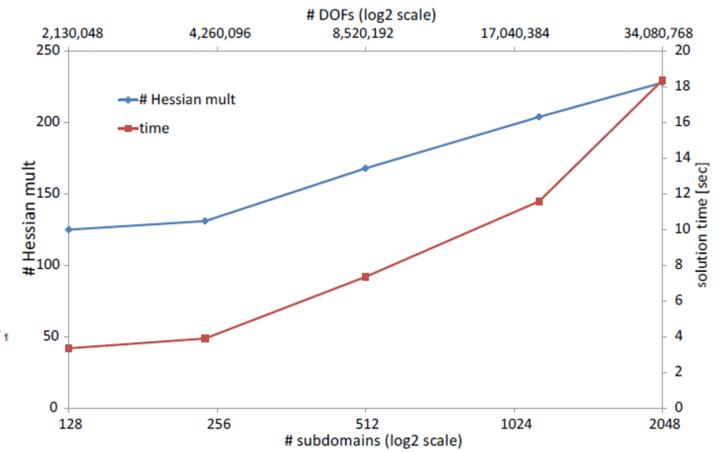
(older PERMON version)

Coercive membrane problem

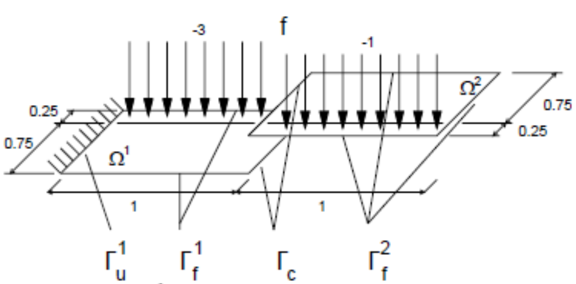


$$f(x,y) = \begin{cases} -1 & \text{for } (x,y)\ (0,1)\times[0.75,1) \\ 0 & \text{for } (x,y)\ (0,1)\times[0,0.75) \\ & \text{and } (x,y)\ (1,2)\times[0.25,1) \\ -3 & \text{for } (x,y)\ (1,2)\times[0,0.25) \end{cases}$$

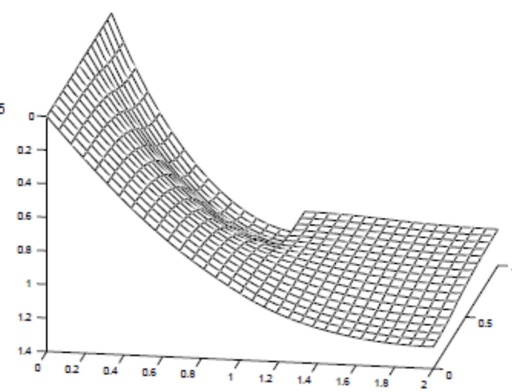


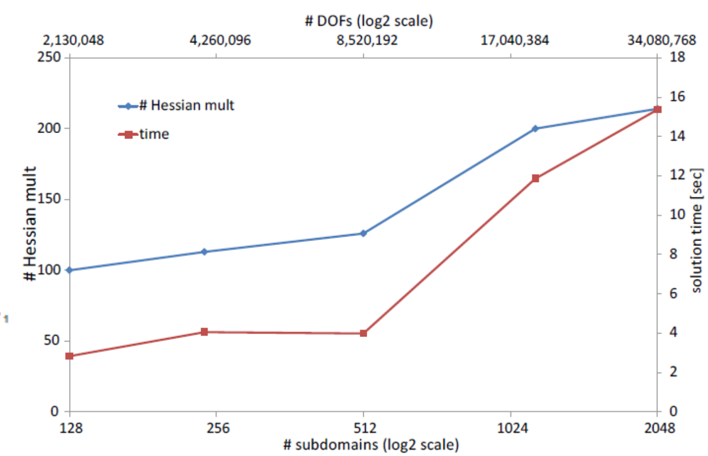


Semicoercive membrane problem



$$f(x,y) = \begin{cases} -3 & \text{for } (x,y)\ (0,1)\times[0.75,1) \\ 0 & \text{for } (x,y)\ (0,1)\times[0,0.75) \\ & \text{and } (x,y)\ (1,2)\times[0.25,1) \\ -1 & \text{for } (x,y)\ (1,2)\times[0,0.25) \end{cases}$$





IT4Innovations supercomputing for#industry01

Department of Applied Mathematics

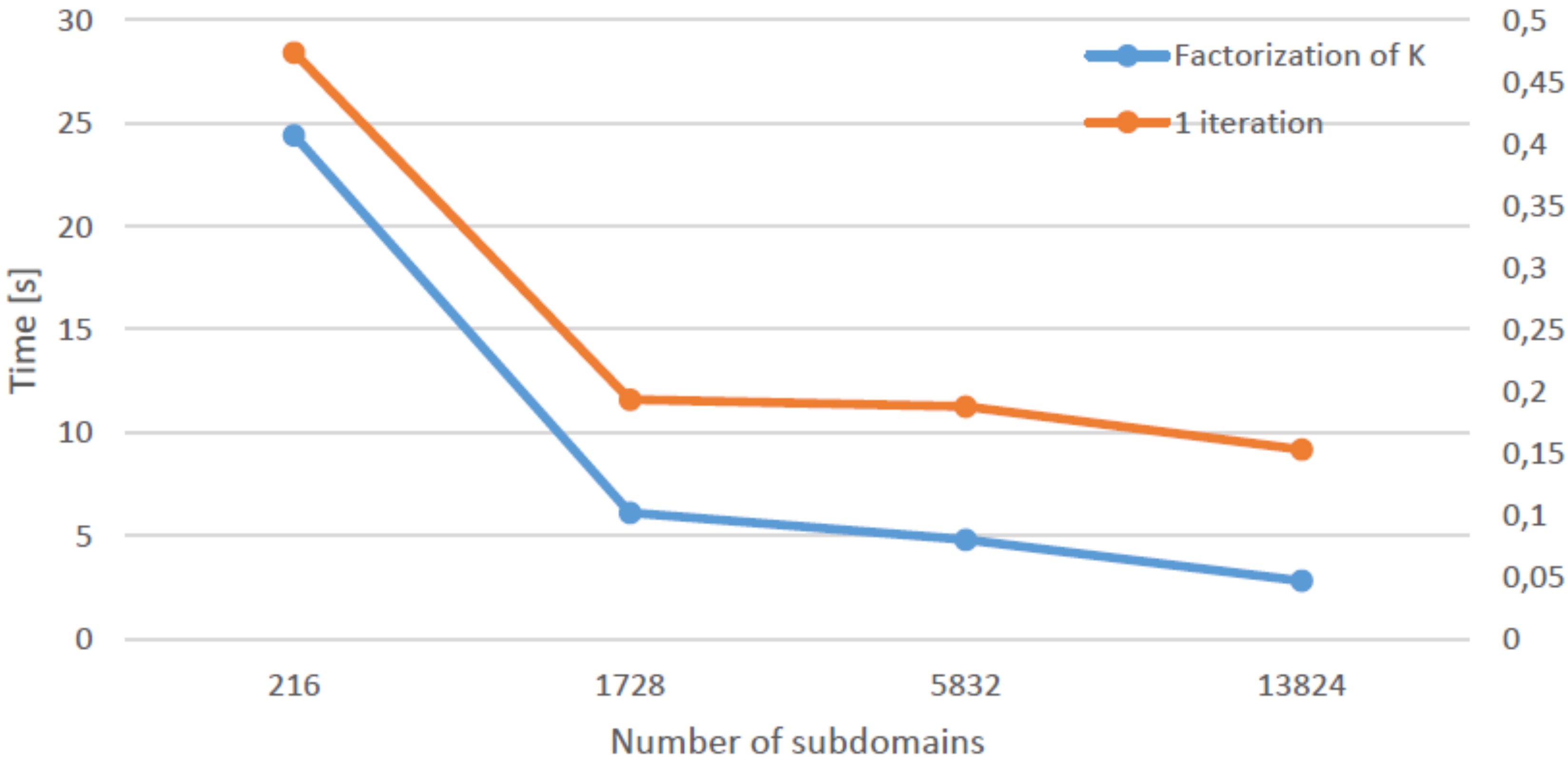
PERMON.it4i.cz

Experiments show

- **FETI DDM numerical scalability**
  - condition number is O(n) for constant local problems size n
- **parallel scalability** of PermonFLLOP's FETI implementation
- **SMALXE numerical scalability**
  - number of iterations given by Hessian spectrum
- **PermonQP** infrastructure does **not slow** it **down**

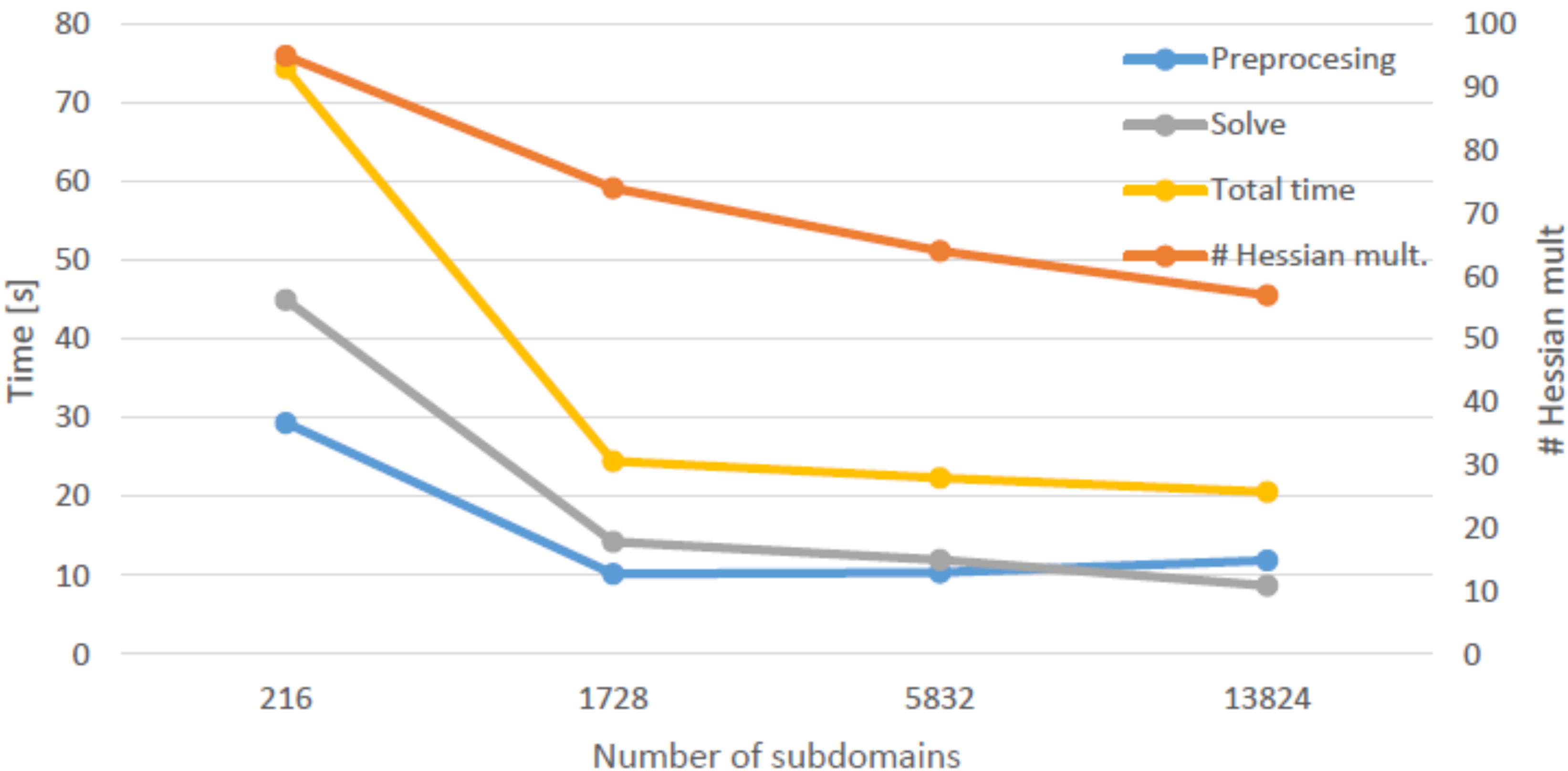
New feature: >1 subdomains per rank

- 216 MPI processes/cores
- cost of factorization and one iteration
- MUMPS



New feature: >1 subdomains per rank

- 216 MPI processes/cores
- time of setup phase and solve phase and total time
- effect of improving condition number



Bleeding edge

- optimized assembly and storage of mortar **B** matrices
  - support for non-conforming submeshes – hanging nodes at interfaces
- stiffness matrix kernel **R** black-box stochastic generation
- other applications of QP – *do you use some*?
  - support vector machines, data fitting, SQP, Proximal Bundle Method,…
- condensation of local problems
  - improve arithmetical density, accelerators
- real-world problems using existing FEM software
  - libMesh, FEniCS, Sifel, Elmer, ViennaMesh/Grid tools
- other QP algorithms – IP, semismooth Newton
- gitpublish under FreeBSD (2-clause BSD) license this year
  - merge separable convex constraints (QPC)
  - finish splitting into PermonQP and PermonFLLOP
  - basic documentation

Acknowledgements

READEX
Runtime Exploitation of Application Dynamism for Energy-efficient eXascale computing

The EXA2CT European Project (EXascale Algorithms and Advanced Computational Techniques)

EXA2CT

GAČR
GRANTOVÁ AGENTURA ČESKÉ REPUBLIKY

Účinné metody odhadu životnosti pro obecné víceosé namáhání
(Efficient methods for life prediction for general multiaxial fatigue)
GA15-18274S

PRACE

MŠMT
MINISTRY OF EDUCATION, YOUTH AND SPORTS

National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science" LQ1602

intel
IPCC-IT4I

IT4Innovations supercomputing for#industry01

Department of Applied Mathematics

VŠB TECHNICKÁ UNIVERZITA OSTRAVA 1849

PERMON.it4i.cz

Additional acknowledgement

"Coffee-driven research"

1^{st} parallel machine @ IT4I *

* It is able to prepare 2 espressos in parallel.

IT4Innovations
supercomputing
for#industry01

Department
of Applied
Mathematics

PERMON.it4i.cz

PERMON team @ IT4I
is your PETSc partner in Czechia ☺

Thank you for your attention!

Do you have any questions?

See you also at our poster.

IT4Innovations
supercomputing
for#industry01

Department
of Applied
Mathematics

PERMON.it4i.cz