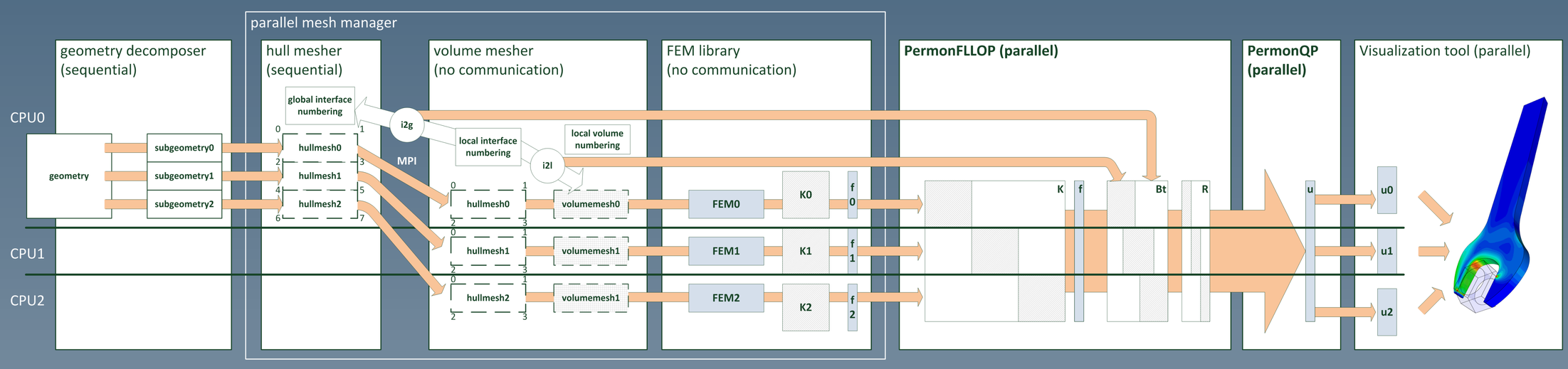
# PERMON

Martin Čermák, Václav Hapla*, David Horák, Jakub Kružík,
Alexandros Markopoulos, Marek Pecha, Lukáš Pospíšil, Radim Sojka,
Alena Vašatová

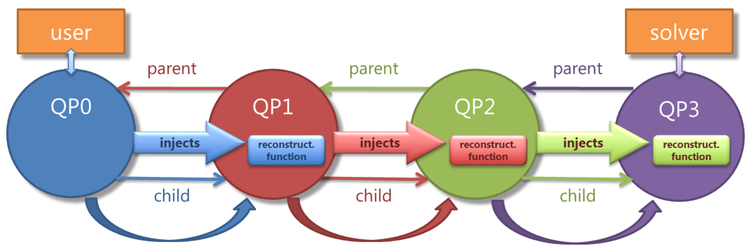
**Parallel, Efficient, Robust, Modular, Object-oriented, Numerical simulations**

- discretization methods
- domain decomposition methods
- linear system solvers
- quadratic programming solvers
- application-specific modules
- PETSc extension




## PermonQP

- framework and concrete solvers quadratic programming (QP)
- QP problems, transforms, solvers
- easy-to-use / HPC-oriented
- workflow
  1. QP problem specification
  2. QP transforms
  3. automatic/manual choice of an appropriate solver
  4. the solver is called




```
QP
QP parent, child; /* QP chain objects */
Vec x; /* solution */

Mat A; /* Hessian matrix */
Vec b; /* right-hand side */

Mat BE; Vec cE; /* linear equality constraints */
Vec lambdaE; /* equality Lagrange multipliers */

QPC constraint; /* inequality constraints */
```

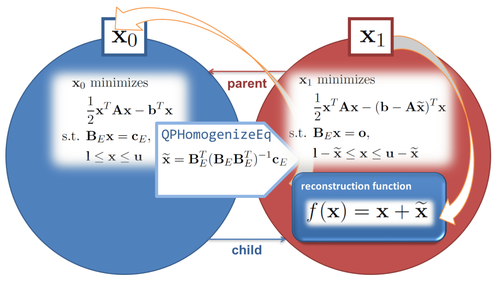
```
QPS
QP topQP, solQP; /* QP chain objects */
PetscReal rtol, atol, divtol; /* alg. tolerances */

PetscReal rnorm; PetscInt it; /* current state */
QPSMonitor monitor;
```

```
QPT
QPTDualize()
QPTEnforceEqByProjector()
QPTEnforceEqByPenalty()
QPTOrthonormalizeEq()
QPTHomogenizeEq()
QPTFetiPrepare()
```

$$\min_{\substack{l_b \le x \le u_b \\ B_{eq}x = c_{eq} \\ B_{ineq}x \le c_{ineq}}} \frac{1}{2}x^T Ax - x^T b$$

### Homogenization of equality constraints

- an example of QP transformation for problem




## PermonFLLOP

- extends PermonQP with domain decomposition methods of the FETI type
- problems without / with contacts
- scalability up to tens of thousands of CPU cores, billions of unknowns
- assembly of FETI-specific objects

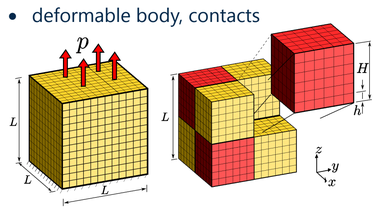
```
Mat Ks, BIs; Vec fs, cI, coords;
IS l2g, dbcis; MPI_Comm comm;
FLLOP fllop;

/*  generate the data */

/*  create FLLOP living in communicator */
FllopCreate(comm, &fllop);

/*  set the subdomain stiffness matrix and load vector */
FllopSetStiffnessMatrix(fllop, Ks);
FllopSetLoadVector(fllop, fs);

/*  set the local-to-global mapping for gluing */
FllopSetLocalToGlobalMapping(fllop, l2g);

/*  specify the Dirichlet conditions in the local numbering and tell FLLOP to enforce them by means of the B matrix */
FllopAddDirichlet(fllop, dbcis, FETI_LOCAL, FETI_DBC_B);

/*  set vertex coordinates for rigid body modes */
FllopSetCoordinates(fllop, coords);

/*  set the non-penetration inequality constraints */
FllopSetIneq(fllop, BIs, cI);

FllopSolve(fllop);
```

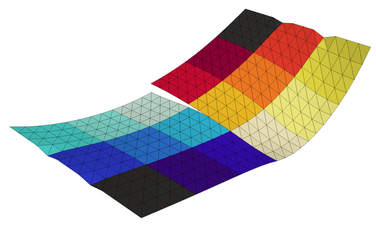
```
/* FllopSolve() function */

/*  subdomain data */
Mat Ks, BIs, Bgs, Bds, Rs;  Vec fs;
/* global data */
Mat K, BI, Bg, Bd, R;  Vec f, cI, cd;
/*  QP problem, QP solvers */
QP qp; QPS qps;

/*  create a QP data structure */
QPCreate(comm,&qp);

/*  Globalise the data. */
MatCreateBlockDiag(Ks,&K);
MatCreateBlockDiag(Rs,&R);
MatMerge(Bgs,&Bg); MatMerge(Bds,&Bd);
MatMerge(BIs,&BI); VecMerge(fs,&f);

/*  Insert the data into the QP. */
QPSetOperator(qp, K);
QPSetOperatorNullspace(qp, R);
QPSetRHS(qp, f);
QPAddEq(qp, Bg, NULL);
QPAddEq(qp, Bd, cd);
QPSetIneq(qp, BI, cI);

/*  Basic sequence of QP transforms
    giving (T)FETI method.
    QP chain is created in backend. */
QPTDualize(qp);
QPTHomogenizeEq(qp);
QPTEnforceEqByProjector(qp);

/*  Create a PermonQP solver. */
QPSCreate(comm, &qps);

/* Set the QP to be solved. */
QPSSetQP(qps,qp);

/*  Solve, i.e. hand over to PermonQP.
    The last QP in the chain is solved. */
QPSSolve(qps);
```

## PermonCube

## PermonMembrane

- benchmark generation - mesh with billion of unknowns in parallel over the cubical or membrane domain
- FEM assembly
- deformable body, contacts

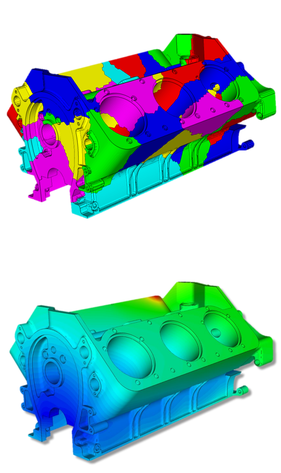


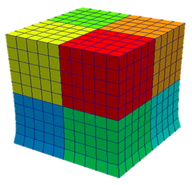



## Numerical experiments

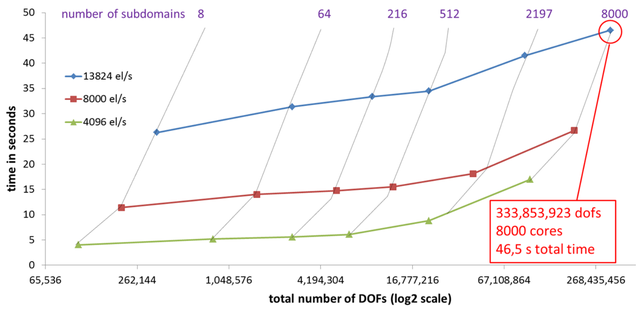
- realistic linear elasticity problem
- displacement computed
- boundary conditions prescribed in one cylinder
- quadratic tetrahedral elements with midnodes
- METIS mesh decomposition
- benchmark prepared by Tomáš Brzobohatý (IT4Innovations)

| | | | |
|---|---|---|---|
| **# DOF** | 98,214,558 | | |
| **# constraints (dual dim.)** | 13,395,882 | | |
| **# subdomains = # cpus** | 5,012 | | |
| **# CG iter.** | 181 | | |
| **computer** | HECToR | CURIE | Hermit |
| **total setup time** | 28 s | 64 s | 30 s |
| **total solution time** | 233 s | 283 s | 283 s |

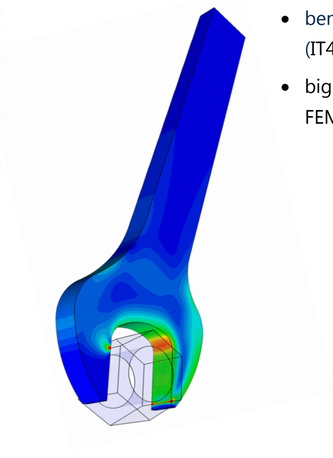




- linear elasticity problem
- problem generated using PermonCube
- computed on HECToR (EPCC)

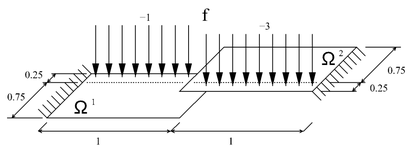


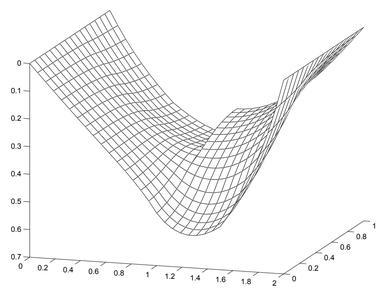

- realistic elasticity problem with contact
- displacements computed
- gluing and non-penetration constraints
- METIS mesh decomposition
- benchmark prepared by Tomáš Brzobohatý (IT4Innovations)
- bigger problems currently limited by the FEM implementation, work in progress

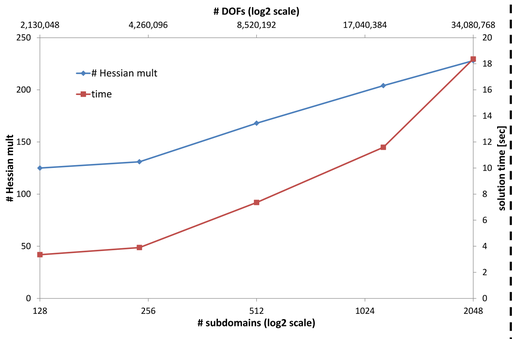


| | |
|---|---|
| **# DOF** | 193,227 |
| **# constraints (dual dim.)** | 24380 |
| **# subdomains = # cpus** | 39 |
| **# outer SMALSE iter.** | 12 |
| **# inner MPRGP iter.** | 588 |
| **computer** | Salomon |
| **total setup time** | 0.5 s |
| **total solution time** | 24.4 s |

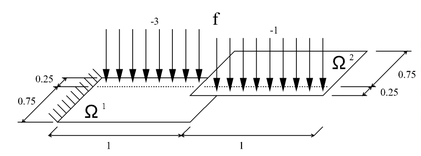
### Coercive membrane problem

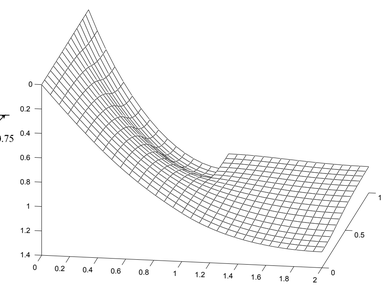


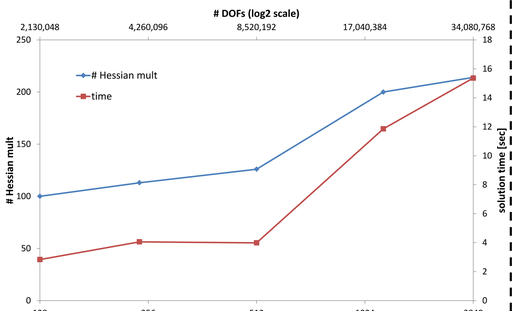





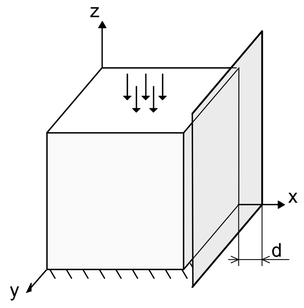

### Semicoercive membrane problem

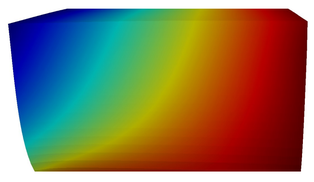


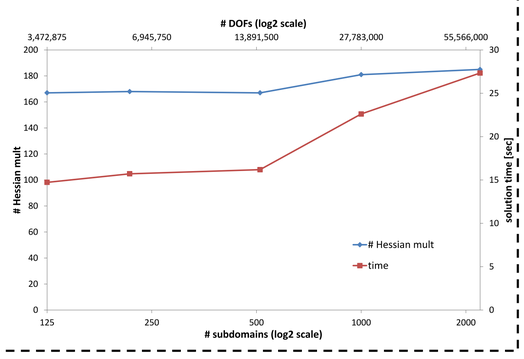



### Elastic cube in potential contact with obstacle



computed on Archer (EPCC)




IT4Innovations national supercomputing center

IT4Innovations supercomputing for industry

Department of Applied Mathematics

SGS SP2016/178

GAČR 15-18274S

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

The Implementation Phase of PRACE receives funding from the EU's Seventh Framework Programme (FP7/2007-2013) under grant agreement RI-312763 and from the EU's Horizon 2020 Research and Innovation Programme (2014-2020) under grant agreement No. 653838

This work was supported by the READEX project - the European Union's Horizon 2020 research and innovation programme under grant agreement No. 671657 and by the EXA2CT project funded from the

Čermák M., Hapla V., Horák D., Merta M., Markopoulos A.: *Total-FETI domain decomposition method for solution of elasto-plastic problems*, DOI: 10.1016/j.advengsoft.2014.12.011, 2015.

Merta M., Vašatová A., Hapla V., Horák D.: *Parallel Implementation of Total-FETI DDM with Application to Medical Image Registration*, DOI: 10.1007/978-3-319-05789-7_89, 2014.

Hapla V., Čermák M., Markopoulos A., Horák D.: *FLLOP: A Massively Parallel Solver Combining FETI Domain Decomposition Method and Quadratic Programming*, DOI: 10.1109/HPCC.2014.56, 2014.

Markopoulos A., Hapla V., Čermák M., Fusek M.: *Massively parallel solution of elastoplasticity problems with tens of millions of unknowns using PermonCube and FLLOP packages*, DOI: 10.1016/j.amc.2014.12.097, 2015.

Dostál Z., Pospíšil L.: *Optimal iterative QP and QPQC algorithms*, DOI: 10.1007/s10479-013-1479-0, 2013.

Bouchala J., Dostál Z., Kozubek T., Pospíšil L., Vodstrčil P.: *On the solution of convex QPQC problems with elliptic and other separable constraints*, DOI: 10.1016/j.amc.2014.09.044, 2014.

Pospíšil L.: *An optimal algorithm with Barzilai-Borwein steplength and superrelaxation for QPQC problem*, WOS:000317994100024, 2012.