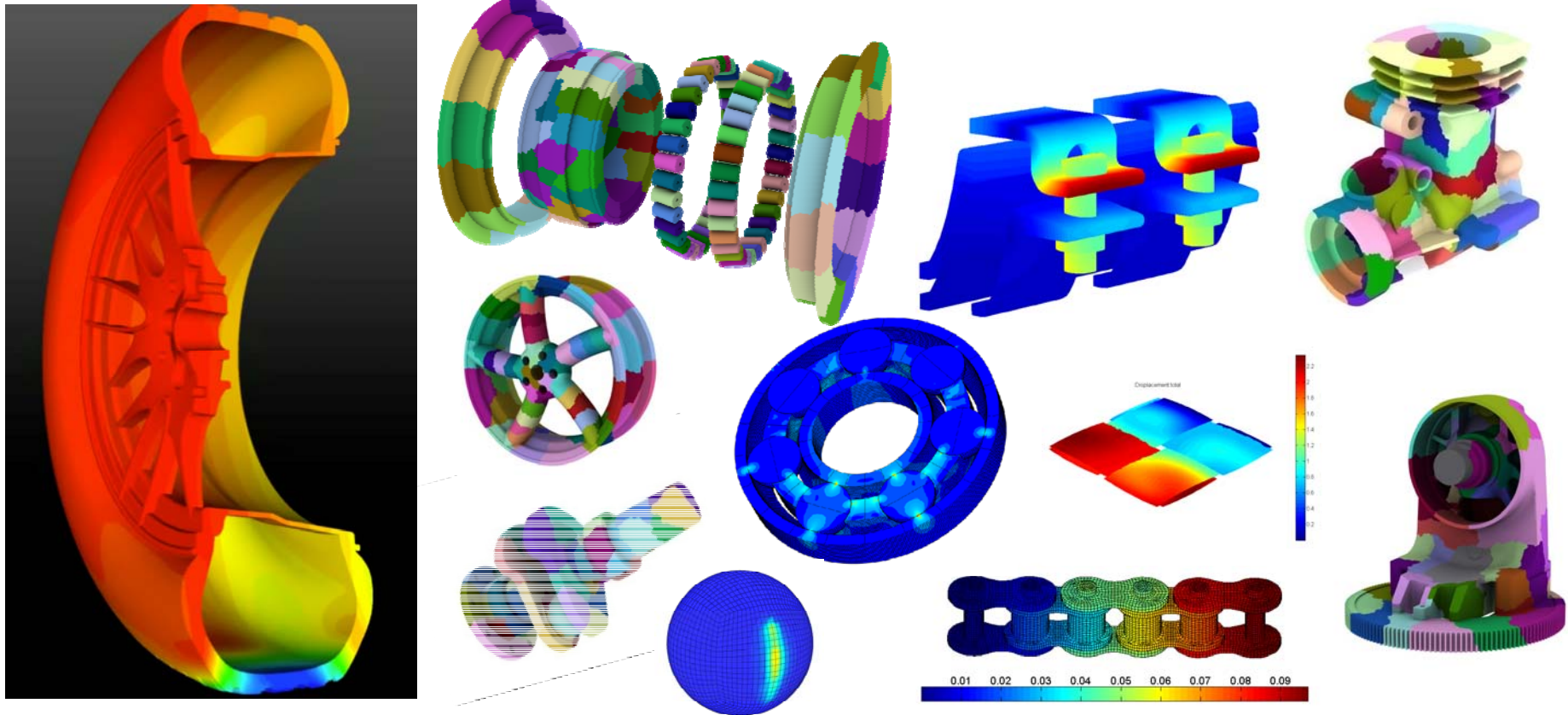


FETI based solvers for exascale computations in mechanics

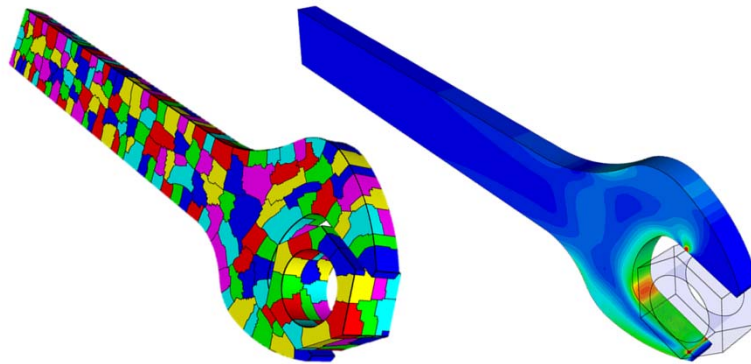
T. Kozubek, T. Brzobohatý, L. Říha, and A. Markopoulos
IT4Innovations, VŠB-Technical University of Ostrava
Ostrava, Czech Republic



Scalable algorithms for contact problems



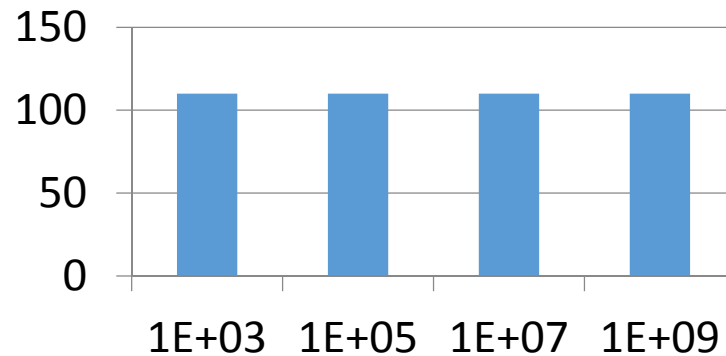
Efficient Solvers in Mechanics



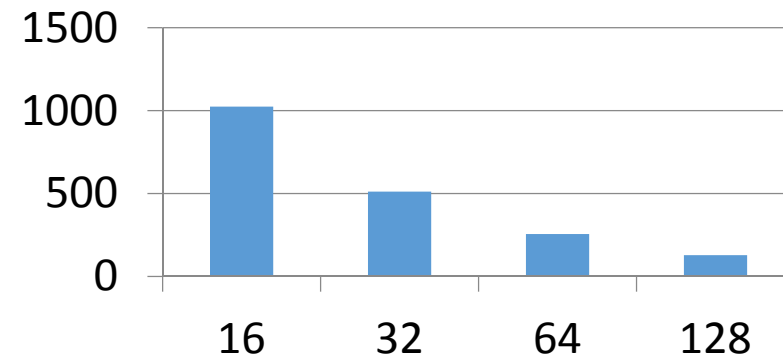
$$Ax = b$$

$$\begin{aligned} \min_{lb \leq x \leq ub} \quad & \frac{1}{2} x^T A x - x^T b \\ \text{subject to} \quad & B_{eq} x = c_{eq} \\ & B_{ineq} x \leq c_{ineq} \end{aligned}$$

Numerical scalability
iterations / problem size

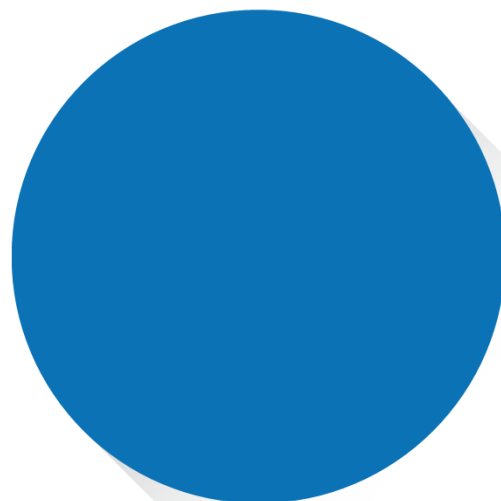


Strong parallel scalability
solution time / cores



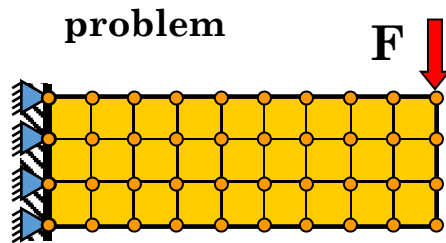
Weak parallel scalability
solution time / problem size

IT4Innovations
national0\$%&@
supercomputing
center\$@00&10%



HTFETI

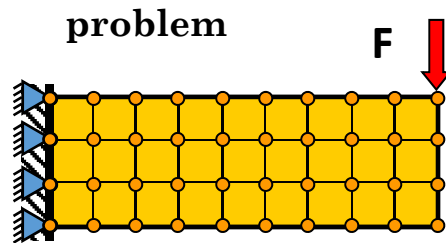
FETI approaches



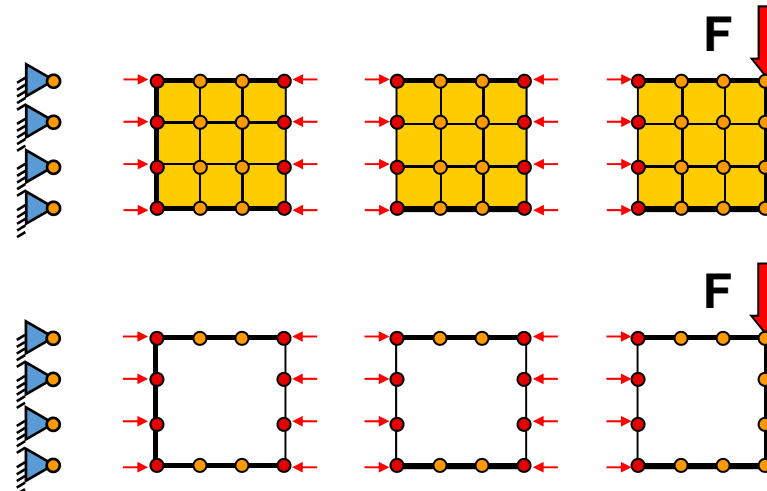
1. FETI		subdomains are fixed or free but with different defects
2. FETI-DP		FETI-DP (partial splitting, nonsingular)
3. TFETI		all subdomains are free with the same defect

BETI approach

IT4Innovations
national
supercomputing
center

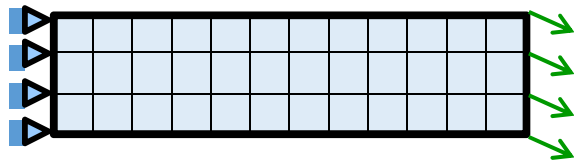


TFETI/TBETI

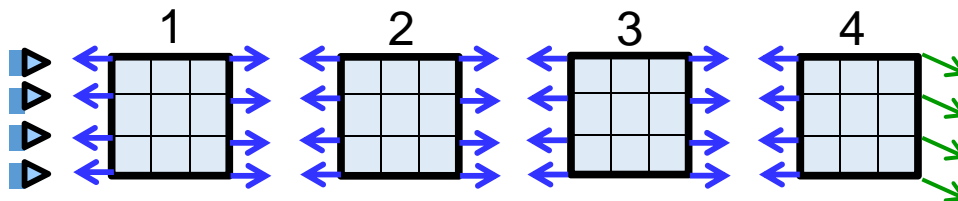


Hybrid Total FETI Method

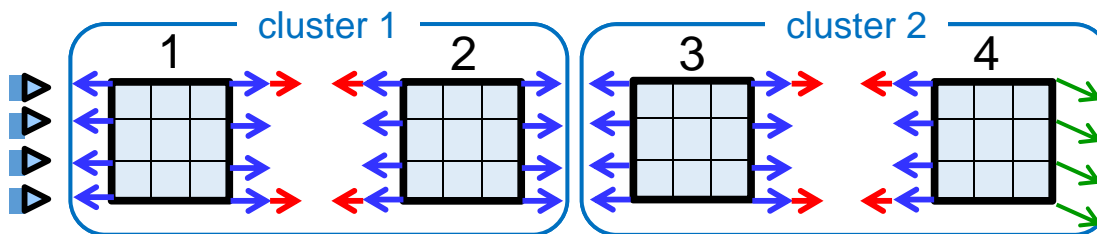
IT4Innovations
national
supercomputing
center



FEM discretization

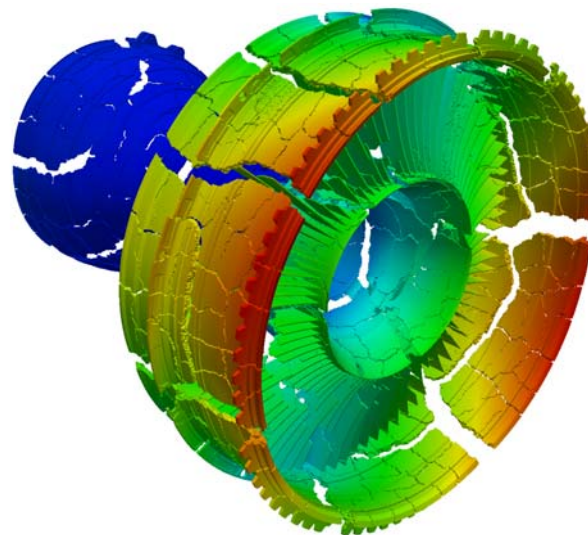
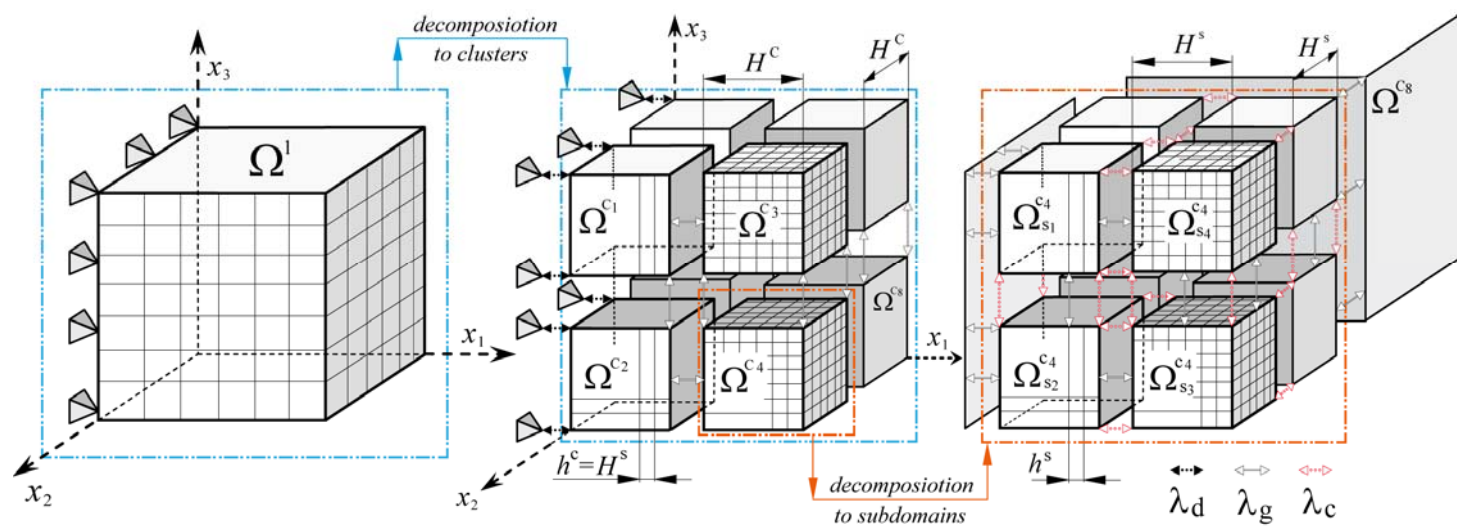


FETI method



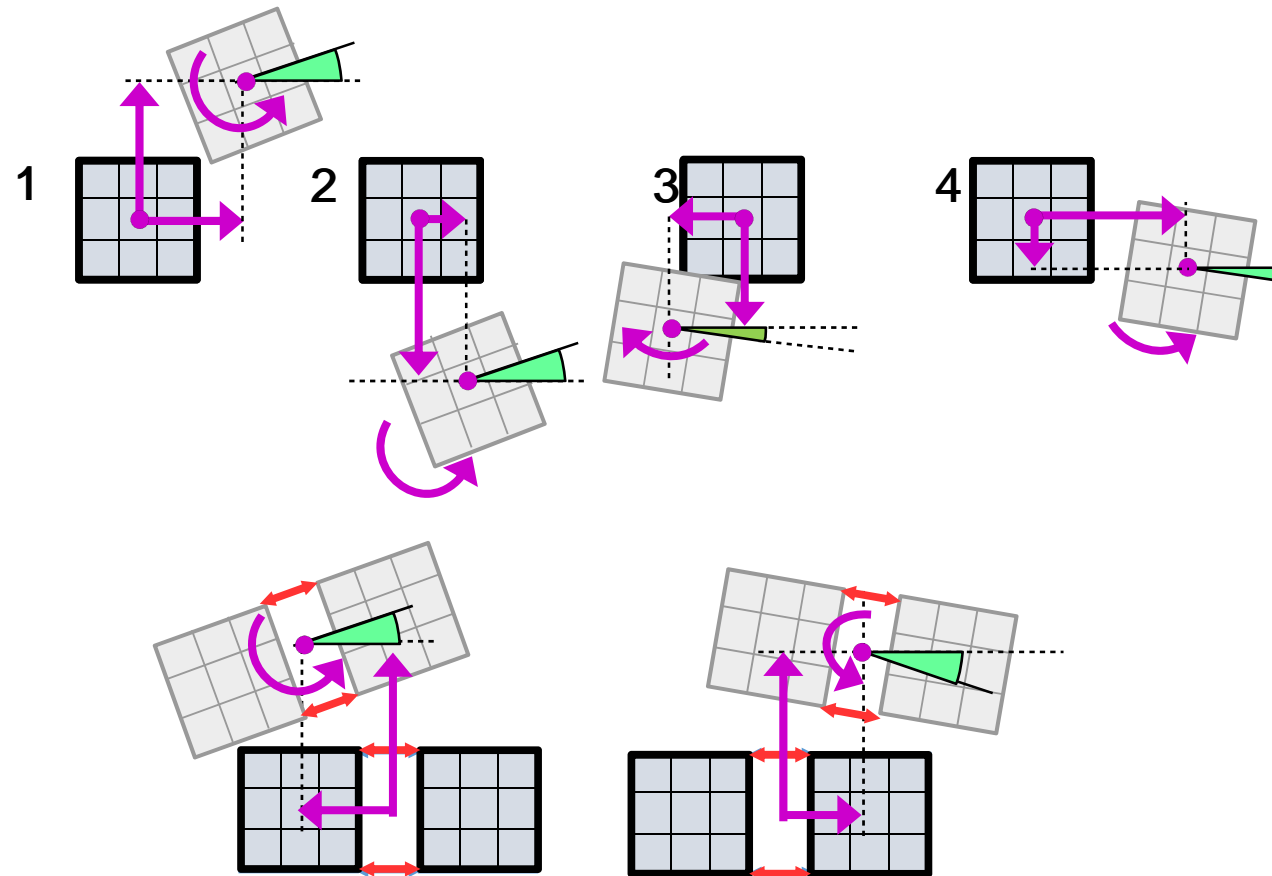
Hybrid FETI method

Hybrid Total FETI Method



Hybrid Total FETI Method

IT4Innovations
national
supercomputing
center



In Hybrid Total FETI method the dimension of GG^T is smaller (dim=6) compared to Total FETI (dim=12) due to additional constraints which remove local rigid body modes.

Hybrid Total FETI Method

Implementation

$$\mathbf{B}_c = \begin{pmatrix} \mathbf{B}_{c,1} & \mathbf{B}_{c,2} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{B}_{c,3} & \mathbf{B}_{c,4} \end{pmatrix}, \quad \mathbf{B} = (\mathbf{B}_1 \ \mathbf{B}_2 \ \mathbf{B}_3 \ \mathbf{B}_4)$$

additional constraints:
*duplication of 'corners' Lagrange
multipliers*

$$\left(\begin{array}{cccc|cc|c} \mathbf{K}_1 & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{B}_{c,1}^\top & \mathbf{O} & \mathbf{B}_1^\top \\ \mathbf{O} & \mathbf{K}_2 & \mathbf{O} & \mathbf{O} & \mathbf{B}_{c,2}^\top & \mathbf{O} & \mathbf{B}_2^\top \\ \mathbf{O} & \mathbf{O} & \mathbf{K}_3 & \mathbf{O} & \mathbf{O} & \mathbf{B}_{c,3}^\top & \mathbf{B}_3^\top \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{K}_4 & \mathbf{O} & \mathbf{B}_{c,4}^\top & \mathbf{B}_4^\top \\ \hline \mathbf{B}_{c,1} & \mathbf{B}_{c,2} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{B}_{c,3} & \mathbf{B}_{c,4} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \hline \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 & \mathbf{B}_4 & \mathbf{O} & \mathbf{O} & \mathbf{O} \end{array} \right) \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \mathbf{u}_4 \\ \hline \lambda_{c,1} \\ \lambda_{c,2} \\ \hline \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \\ \mathbf{f}_4 \\ \hline \mathbf{o} \\ \mathbf{o} \\ \hline \mathbf{c} \end{pmatrix}$$

augmented KKT system
by the matrix \mathbf{B}_c and λ_c

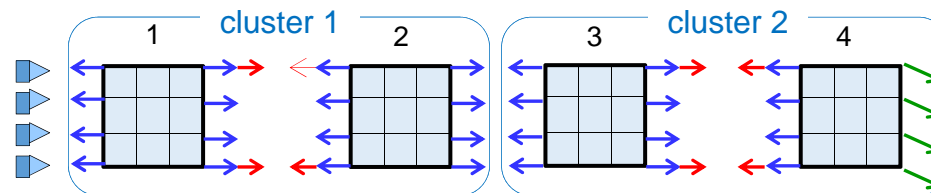
$$\left(\begin{array}{ccc|ccc|c} \mathbf{K}_1 & \mathbf{O} & \mathbf{B}_{c,1}^\top & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{B}_1^\top \\ \mathbf{O} & \mathbf{K}_2 & \mathbf{B}_{c,2}^\top & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{B}_2^\top \\ \mathbf{B}_{c,1} & \mathbf{B}_{c,2} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \hline \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{K}_3 & \mathbf{O} & \mathbf{B}_{c,3}^\top & \mathbf{B}_3^\top \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{K}_4 & \mathbf{B}_{c,4}^\top & \mathbf{B}_4^\top \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{B}_{c,3} & \mathbf{B}_{c,4} & \mathbf{O} & \mathbf{O} \\ \hline \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{O} & \mathbf{B}_3 & \mathbf{B}_4 & \mathbf{O} & \mathbf{O} \end{array} \right) \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \lambda_{c,1} \\ \hline \mathbf{u}_3 \\ \mathbf{u}_4 \\ \lambda_{c,2} \\ \hline \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{o} \\ \hline \mathbf{f}_3 \\ \mathbf{f}_4 \\ \mathbf{o} \\ \hline \mathbf{c} \end{pmatrix}$$

reordering according to clusters

Hybrid Total FETI Method

Implementation

$$\mathbf{B}_c = \begin{pmatrix} \mathbf{B}_{c,1} & \mathbf{B}_{c,2} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{B}_{c,3} & \mathbf{B}_{c,4} \end{pmatrix}, \quad \mathbf{B} = (\mathbf{B}_1 \ \mathbf{B}_2 \ \mathbf{B}_3 \ \mathbf{B}_4) \quad \text{additional constraints (red arrows)}$$



The matrix \mathbf{B}_c is a copy of specific rows from the matrix \mathbf{B} corresponding to components of λ acting on the corners between subdomains 1,2, and 3,4, respectively (red arrows).

Hybrid Total FETI Method

Modified KKT system is solvable by the same family of algorithms like FETI (Preconditioned Conjugate Projected Gradient method, ...).

Size of $\tilde{\mathbf{G}}\tilde{\mathbf{G}}^T$ (or defect of global stiffness matrix) can be **significantly smaller** than in original FETI approach ...

$$\begin{pmatrix} \tilde{\mathbf{F}} & \tilde{\mathbf{G}} \\ \tilde{\mathbf{G}}^T & \mathbf{O} \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{\lambda}} \\ \tilde{\boldsymbol{\alpha}} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{d}} \\ \tilde{\mathbf{e}} \end{pmatrix}$$

$$\tilde{\mathbf{K}} = \text{diag}(\tilde{\mathbf{K}}_1, \tilde{\mathbf{K}}_2)$$

$$\tilde{\mathbf{R}}^T = (\tilde{\mathbf{R}}_1^T, \tilde{\mathbf{R}}_2^T)$$

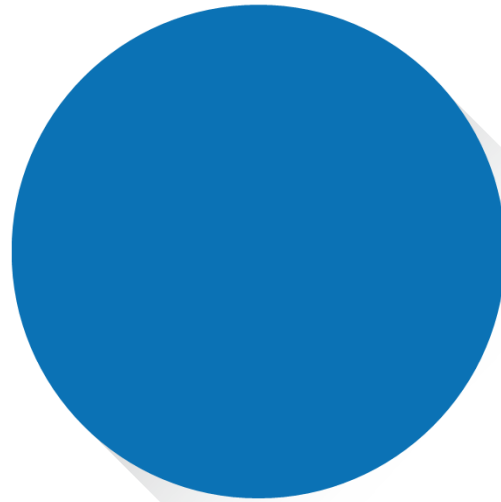
$$\tilde{\mathbf{B}} = (\tilde{\mathbf{B}}_1, \tilde{\mathbf{B}}_2)$$

$$\tilde{\mathbf{F}} = \tilde{\mathbf{B}}\tilde{\mathbf{K}}^+ \tilde{\mathbf{B}}^T$$

$$\tilde{\mathbf{d}} = \tilde{\mathbf{B}}\tilde{\mathbf{K}}^+ \tilde{\mathbf{f}} - \tilde{\mathbf{c}}$$

$$\tilde{\mathbf{G}} = -\tilde{\mathbf{B}}\tilde{\mathbf{R}}$$

$$\tilde{\mathbf{e}} = -\tilde{\mathbf{R}}^T \tilde{\mathbf{f}}^T$$



Pipelining

Bi-Conjugate Gradients Stabilized (BiCGStab)

IT4Innovations
national
supercomputing
center

Algorithm 4 Standard BiCGStab

```
1: function BICGSTAB( $A, b, x_0$ )
2:    $r_0 := b - Ax_0$ ;  $p_0 := r_0$ 
3:   for  $i = 0, \dots$  do
4:      $s_i := Ap_i$ 
5:     compute  $(r_0, s_i)$ 
6:      $\alpha_i := (r_0, r_i) / (r_0, s_i)$ 
7:      $q_i := r_i - \alpha_i s_i$ 
8:      $y_i := Aq_i$ 
9:     compute  $(q_i, y_i)$  ;  $(y_i, y_i)$ 
10:     $\omega_i := (q_i, y_i) / (y_i, y_i)$ 
11:     $x_{i+1} := x_i + \alpha_i p_i + \omega_i q_i$ 
12:     $r_{i+1} := q_i - \omega_i y_i$ 
13:    compute  $(r_0, r_{i+1})$ 
14:     $\beta_i := (\alpha_i / \omega_i) (r_0, r_{i+1}) / (r_0, r_i)$ 
15:     $p_{i+1} := r_{i+1} + \beta_i (p_i - \omega_i s_i)$ 
16:  end for
17: end function
```

dot-prod
SpMV
axpy

Traditional BiCGStab:

(non-preconditioned)

Global communication

- ▶ 3 global reduction phases

Semi-local communication

- ▶ 2 non-overlapping SpMVs

Local communication

- ▶ 4 axpy(-like) operations

General two-step framework for deriving pipelined Krylov methods:

*Step 1. **Avoiding communication:** merge global reductions*

*Step 2. **Hiding communication:** overlap SpMVs & global reductions*

Pipelined BiCGStab (p-BiCGStab) by Wim Vanroose – EXA2CT project

Algorithm 6 Pipelined BiCGStab

```

1: function PIPE-BICGSTAB( $A, b, x_0$ )
2:    $r_0 := b - Ax_0$ ;  $w_0 := Ar_0$ ;  $t_0 := Aw_0$ ;
3:   for  $i = 0, \dots$  do
4:      $p_i := r_i + \beta_{i-1} (p_{i-1} - \omega_{i-1} s_{i-1})$ 
5:      $s_i := w_i + \beta_{i-1} (s_{i-1} - \omega_{i-1} z_{i-1})$ 
6:      $z_i := t_i + \beta_{i-1} (z_{i-1} - \omega_{i-1} v_{i-1})$ 
7:      $q_i := r_i - \alpha_i s_i$ 
8:      $y_i := w_i - \alpha_i z_i$ 
9:     compute  $(q_i, y_i)$  ;  $(y_i, y_i)$ 
10:     $\omega_i := (q_i, y_i) / (y_i, y_i)$ 
11:    overlap  $v_i := Az_i$ 
12:     $x_{i+1} := x_i + \alpha_i p_i + \omega_i q_i$ 
13:     $r_{i+1} := q_i - \omega_i y_i$ 
14:     $w_{i+1} := y_i - \omega_i (t_i - \alpha_i v_i)$ 
15:    compute  $(r_0, r_{i+1})$  ;  $(r_0, w_{i+1})$  ;  $(r_0, s_i)$  ;  $(r_0, z_i)$ 
16:     $\beta_i := (\alpha_i / \omega_i) (r_0, r_{i+1}) / (r_0, r_i)$ 
17:     $\alpha_{i+1} := (r_0, r_{i+1}) / ((r_0, w_{i+1}) + \beta_i (r_0, s_i) - \beta_i \omega_i (r_0, z_i))$ 
18:    overlap  $t_{i+1} := Aw_{i+1}$ 
19:  end for
20: end function

```

dot-prod
SpMV
axpy

p-BiCGStab:

(non-preconditioned)

Global communication

- 2 global red. phases (vs. 3)

Semi-local communication

- 2 overlapping SpMVs

Local communication

- 8 axpy(-like) operations (vs. 4)

Status after Step 2: both global comm. phases are overlapped with SpMV computations ('hidden'), at the cost of 4 additional axpys

Preconditioned pipelined BiCGStab

Algorithm 8 Preconditioned Pipelined BiCGStab

```

1: function P-PIPE-BICGSTAB( $A, M^{-1}, b, x_0$ )
2:    $r_0 := b - Ax_0$ ;  $\hat{r}_0 := M^{-1}r_0$ ;  $w_0 := A\hat{r}_0$ ;  $\hat{w}_0 := M^{-1}w_0$ 
3:    $t_0 := A\hat{w}_0$ ;  $\alpha_0 := (r_0, r_0) / (r_0, w_0)$ ;  $\beta_{-1} := 0$ 
4:   for  $i = 0, \dots$  do
5:      $\hat{p}_i := \hat{r}_i + \beta_{i-1} (\hat{p}_{i-1} - \omega_{i-1} \hat{s}_{i-1})$ 
6:      $s_i := w_i + \beta_{i-1} (s_{i-1} - \omega_{i-1} z_{i-1})$ 
7:      $\hat{s}_i := \hat{w}_i + \beta_{i-1} (\hat{s}_{i-1} - \omega_{i-1} \hat{z}_{i-1})$ 
8:      $z_i := t_i + \beta_{i-1} (z_{i-1} - \omega_{i-1} v_{i-1})$ 
9:      $q_i := r_i - \alpha_i s_i$ 
10:     $\hat{q}_i := \hat{r}_i - \alpha_i \hat{s}_i$ 
11:     $y_i := w_i - \alpha_i z_i$ 
12:    compute  $(q_i, y_i)$  ;  $(y_i, y_i)$ 
13:     $\omega_i := (q_i, y_i) / (y_i, y_i)$ 
14:    overlap  $\hat{z}_i := M^{-1}z_i$ 
15:    overlap  $v_i := A\hat{z}_i$ 
16:     $x_{i+1} := x_i + \alpha_i \hat{p}_i + \omega_i \hat{q}_i$ 
17:     $r_{i+1} := q_i - \omega_i y_i$ 
18:     $\hat{r}_{i+1} := \hat{q}_i - \omega_i (\hat{w}_i - \alpha_i \hat{z}_i)$ 
19:     $w_{i+1} := y_i - \omega_i (t_i - \alpha_i v_i)$ 
20:    compute  $(r_0, r_{i+1})$  ;  $(r_0, w_{i+1})$  ;  $(r_0, s_i)$  ;  $(r_0, z_i)$ 
21:     $\beta_i := (\alpha_i / \omega_i) (r_0, r_{i+1}) / (r_0, r_i)$ 
22:     $\alpha_{i+1} := (r_0, r_{i+1}) / ((r_0, w_{i+1}) + \beta_i (r_0, s_i) - \beta_i \omega_i (r_0, z_i))$ 
23:    overlap  $\hat{w}_{i+1} := M^{-1}w_{i+1}$ 
24:    overlap  $t_{i+1} := A\hat{w}_{i+1}$ 
25:  end for
26: end function

```

dot-prod
SpMV
axpy

*Like for any pipelined method,
including a preconditioner is easy.*

p-BiCGStab:
(preconditioned)

Global communication

- 2 global red. phases
(vs. 3)

Semi-local communication

- 2 overlapping Prec +
SpMV

Local communication

- 11 axpy(-like) operations
(vs. 4)

Pipelined BiCGStab vs. Improved BiCGStab

 L.T. Yang and R.P. Brent. The improved BiCGStab method for large and sparse unsymmetric linear systems on parallel distributed memory architectures. In Proceedings of the Fifth International Conference on Algorithms and Architectures for Parallel Processing, pp. 324–328, IEEE, 2002.

	GLRED	SPMV	Flops (AXPY + DOT-PROD)	Time (GLRED + SPMV)	Memory
BiCGStab	3	2	20	3 GLRED + 2 SPMV	7
IBiCGStab	1	2	30	1 GLRED + 2 SPMV	10
p-BiCGstab	2	2*	38	2 max(GLRED, SPMV)	11

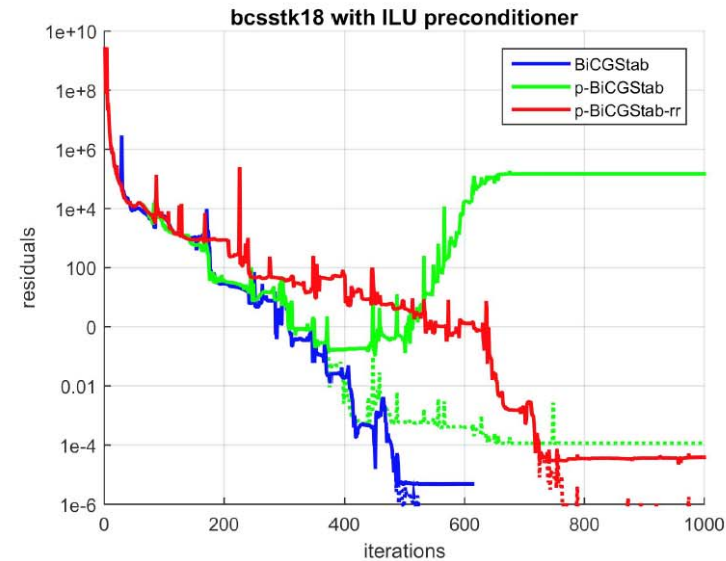
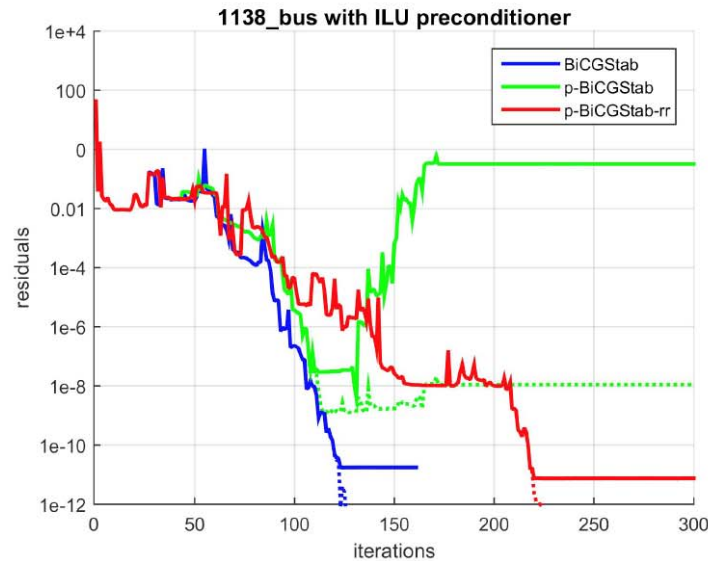
If $\text{time}(\text{GLRED}) \approx \text{time}(\text{SPMV})$:

- ▶ speed-up factor $p\text{-BiCGStab}/\text{BiCGStab} = 2.5$
- ▶ speed-up factor $\text{IBiCGStab}/\text{BiCGStab} = 1.66$

If $\text{time}(\text{GLRED}) \gg \text{time}(\text{SPMV})$:

- ▶ speed-up factor $p\text{-BiCGStab}/\text{BiCGStab} = 2.5$
- ▶ speed-up factor $\text{IBiCGStab}/\text{BiCGStab} = 3.0$

Robustness and attainable accuracy: p-BiCGStab-rr



Residual replacement every rr -th iteration

(non-automated, i.e. rr is a parameter of the method, but chosen large s.t. no. res. repl. is small)

$$r_i := b - Ax_i,$$

$$\hat{r}_i := M^{-1}r_i,$$

$$w_i := A\hat{r}_i,$$

$$s_i := A\hat{p}_i,$$

$$\hat{s}_i := M^{-1}s_i,$$

$$z_i := A\hat{s}_i.$$

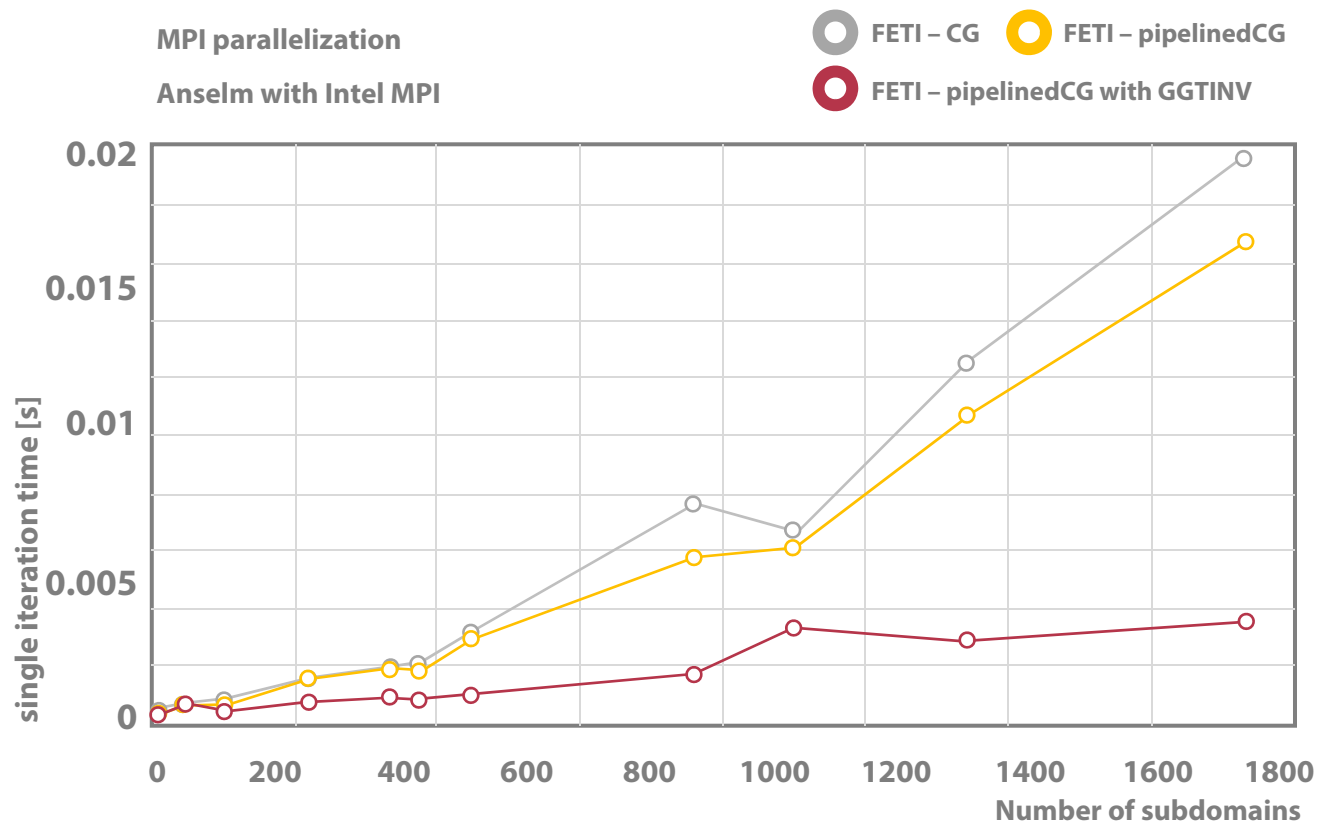
- ⊕ increased **maximal attainable accuracy**: comparable to BiCGStab level
- ⊕ increased **robustness**: negates instable true residual behaviour
- ⊖ increased **number of iterations** possible



ESPRESSO FETI

IT4Innovations
national
supercomputing
center

Optimization of global communication for FETI Solvers



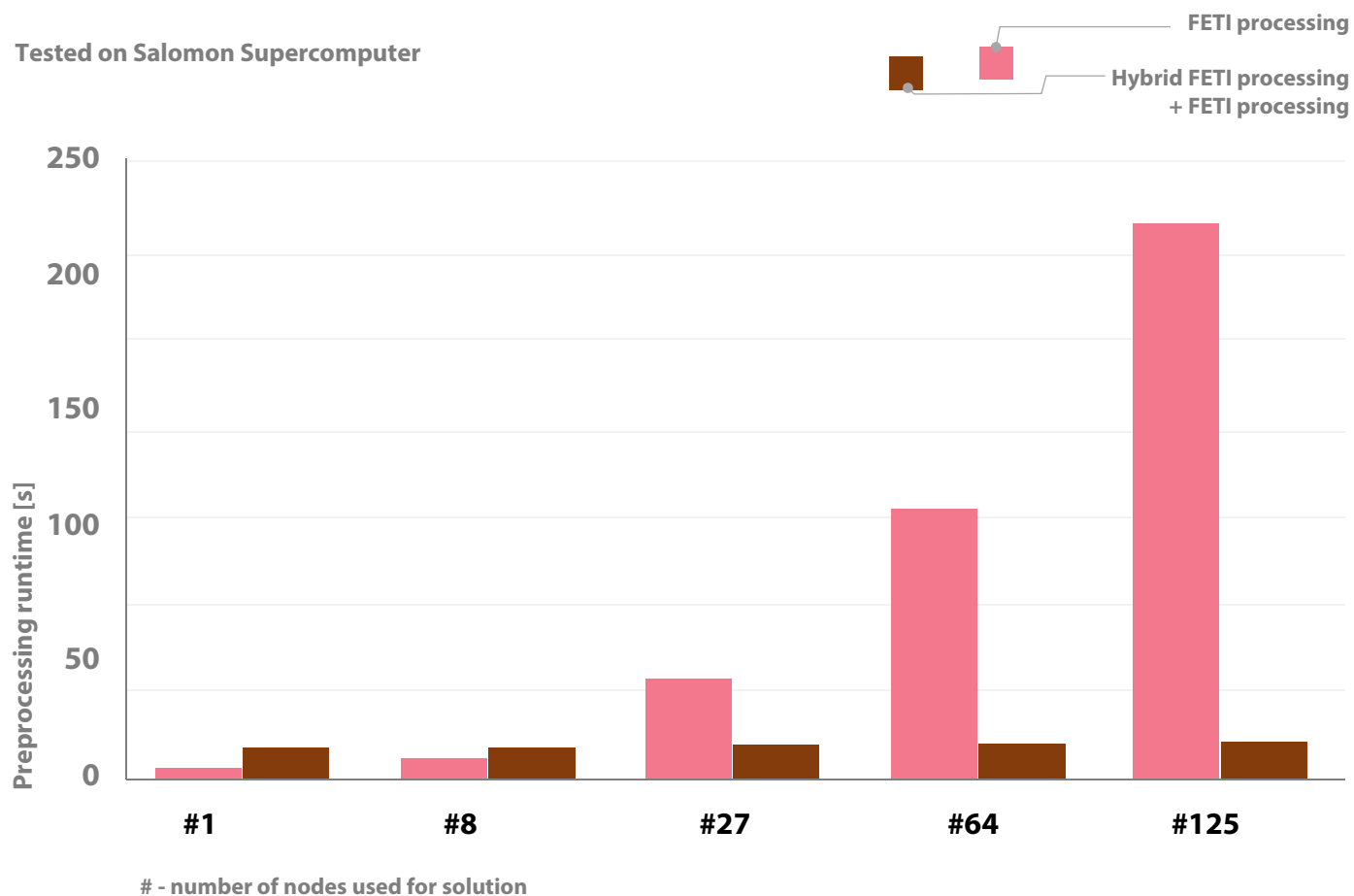


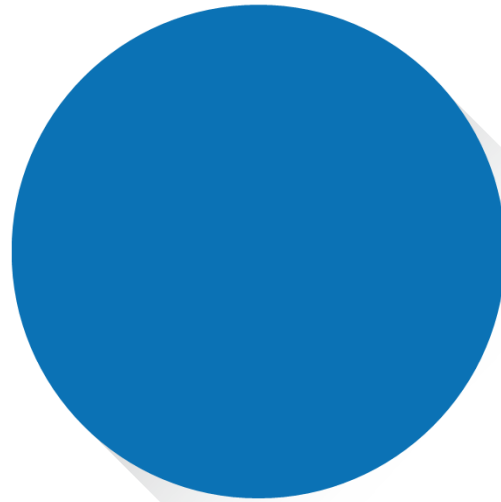
ESPRESSO FETI

IT4Innovations
national
supercomputing
center

Hybrid FETI vs. FETI preprocessing

Tested on Salomon Supercomputer



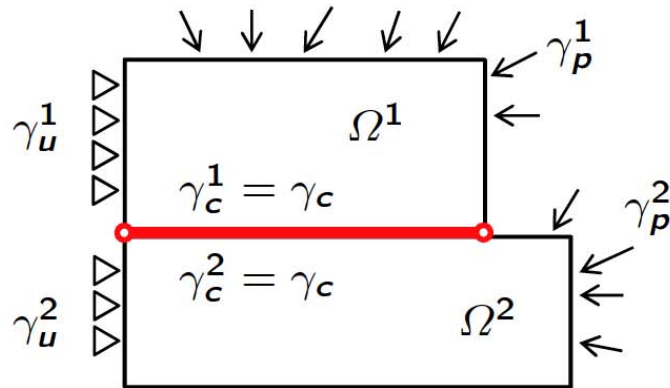


Contact problems

Contact problems of linear elasticity with Tresca friction

- Elastic bodies $\Omega^1, \Omega^2 \subset \mathbb{R}^3$:

$$\partial\Omega^k = \bar{\gamma}_u^k \cup \bar{\gamma}_p^k \cup \bar{\gamma}_c^k, \quad k = 1, 2$$



- Lame PDEs + b.c.:

$$\begin{aligned} -\operatorname{div} \sigma^k &= \mathbf{f}^k && \text{in } \Omega^k \\ \sigma^k &= \mu^k \operatorname{tr}(\varepsilon^k) \mathbf{I} + 2\lambda^k \varepsilon^k && \text{in } \Omega^k \\ \varepsilon^k &= \frac{1}{2}(\nabla \mathbf{u}^k + \nabla^\top \mathbf{u}^k) && \text{in } \Omega^k \\ \mathbf{u}^k &= \mathbf{0} && \text{on } \gamma_u^k \\ \sigma^k \mathbf{n}^k &= \mathbf{p}^k && \text{on } \gamma_p^k \end{aligned}$$

for $k = 1, 2$

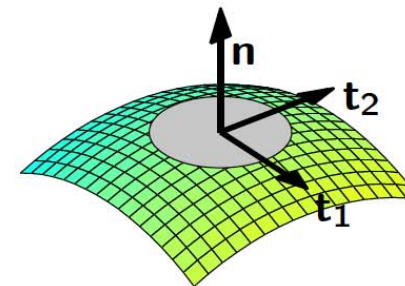
- Unilateral contact on γ_c :

$$u_n - d \leq 0, \quad \sigma_n \leq 0, \quad \sigma_n(u_n - d) = 0$$

- Tresca friction on γ_c : slip bound $g > 0$

$$\begin{aligned} \|\sigma_t\| &\leq g \\ \|\sigma_t\| &< g &\Rightarrow \mathbf{u}_t = \mathbf{0} \\ \|\sigma_t\| &= g &\Rightarrow \exists c \geq 0 : \mathbf{u}_t = -c \sigma_t, \end{aligned}$$

$$\mathbf{u}_t = (u_{t1}, u_{t2}), \quad \sigma_t = (\sigma_{t1}, \sigma_{t2})$$



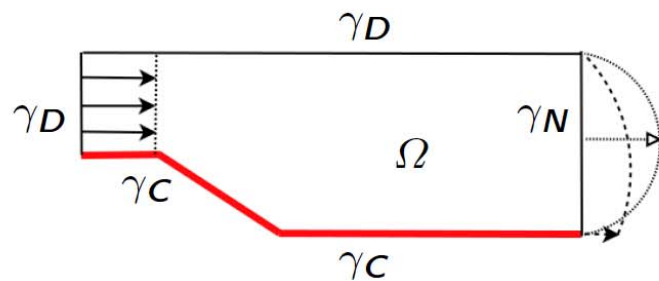
$\{\mathbf{n}, \mathbf{t}_1, \mathbf{t}_2\}$ local basis in points of γ_c

- starting point for other friction models (Coulomb, ...)

Flow problems with the stick-slip b.c.

- Flow in channel $\Omega \subset \mathbb{R}^2$:

$$\partial\Omega = \bar{\gamma}_D \cup \bar{\gamma}_N \cup \bar{\gamma}_C$$



- Stokes problem + b.c.:

$$-\nu \Delta \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega$$

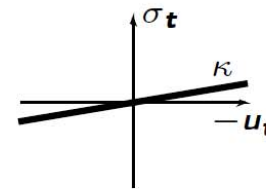
$$\mathbf{u} = \mathbf{u}_D \quad \text{on } \gamma_D$$

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_N \quad \text{on } \gamma_N$$

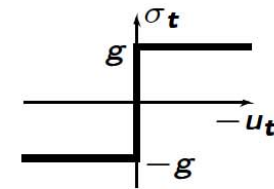
$$u_n = 0 \quad \text{on } \gamma_C$$

$$\begin{aligned} u_t = 0 &\Rightarrow |\sigma_t| \leq g && \text{on } \gamma_C \\ \sigma_t u_t + g|u_t| + \kappa u_t^2 &= 0 && \text{on } \gamma_C \end{aligned}$$

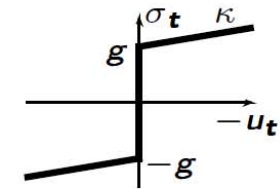
- Stick-slip b.c.: slip bound $g \geq 0$
adhesion $\kappa \geq 0$



Navier
(1826)
 $g = 0$



Tresca
(1994)
 $\kappa = 0$



Navier+Tresca
(2016)
 $g > 0, \kappa > 0$

- Dual problem for $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_t, \boldsymbol{\lambda}_n, \mathbf{p}) \in \mathbb{R}^{2m+n_p}$:

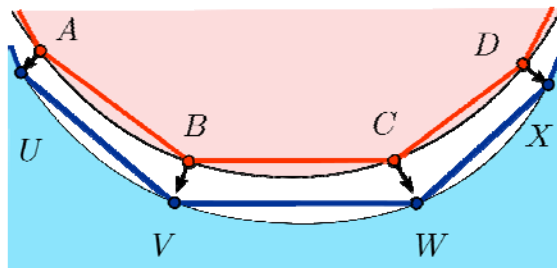
$$\boldsymbol{\lambda}^* = \arg \min_{\boldsymbol{\lambda} \in \Lambda} q(\boldsymbol{\lambda})$$

with $q(\boldsymbol{\lambda}) = \frac{1}{2} \boldsymbol{\lambda}^\top \mathbf{A}_\kappa \boldsymbol{\lambda} - \boldsymbol{\lambda}^\top \mathbf{b}$ on feasible set

$$\Lambda = \{\boldsymbol{\lambda}_t \in \mathbb{R}^m : |\lambda_{t,i}| \leq g_i, i \in \mathcal{M}\} \times \mathbb{R}^{m+n_p}.$$

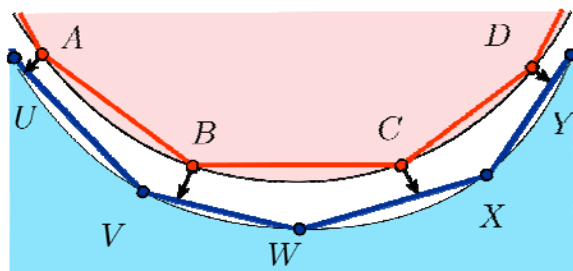
Contact discretization types

- node x node



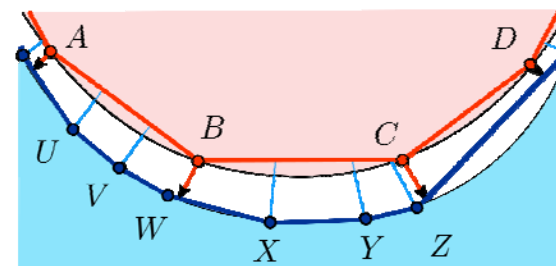
$$\begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 1 & \dots & -1 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \mathbf{u} = \begin{bmatrix} \vdots \\ 0 \\ \vdots \end{bmatrix}$$

- node x element



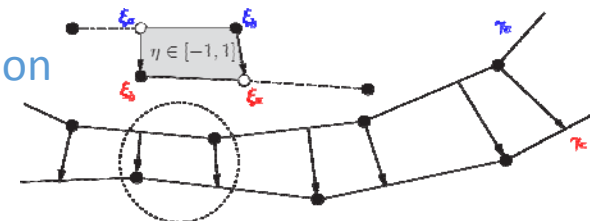
$$\begin{bmatrix} \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & 1 & \dots & -\alpha & \dots & -1 + \alpha \\ \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \mathbf{u} = \begin{bmatrix} \vdots \\ 0 \\ \vdots \end{bmatrix}$$

- mortars

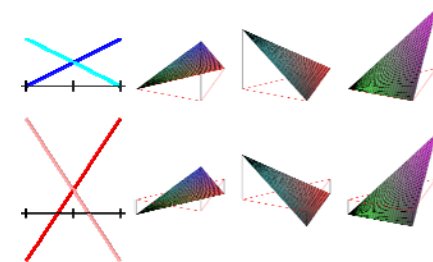


$$\begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & \int_{\gamma_e} \psi_i d\gamma & \dots & \int_{\gamma_e} \psi_i \phi_{j_1} d\gamma & \dots & \int_{\gamma_e} \psi_i \phi_{j_n} d\gamma & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \mathbf{u} = \begin{bmatrix} \vdots \\ 0 \\ \vdots \end{bmatrix}$$

Segmentation

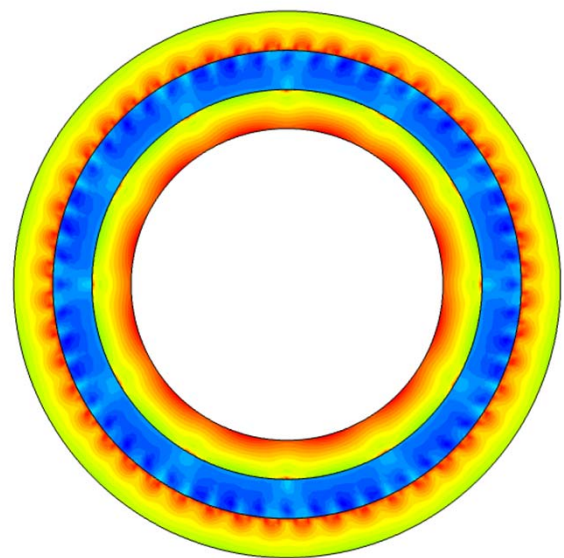


Primal x dual
Basis functions



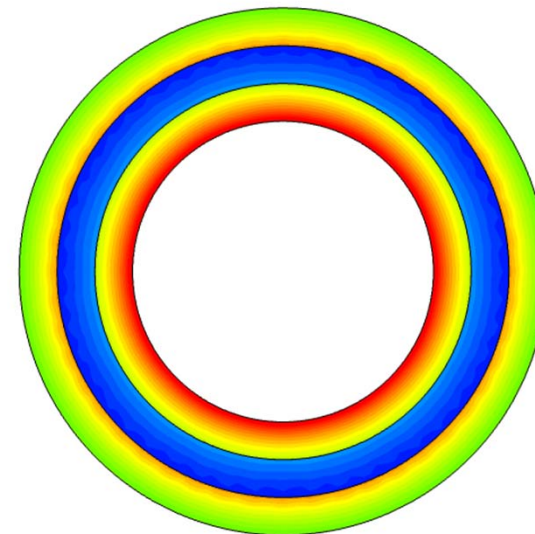
Contact discretization

IT4Innovations
national
supercomputing
center



Stress HMH
22.64 74.03 125.43 176.83 228.22

55.04 14.03 152.43 110.83 55.55
HMH stress



Stress HMH
45.78 90.35 134.91 179.48 224.04

42.18 80.32 134.91 110.48 55.04
HMH stress

Dual formulation

$$L(\mathbf{u}, \boldsymbol{\lambda}_N, \boldsymbol{\lambda}_E, \boldsymbol{\tau}) = f(\mathbf{u}) + \boldsymbol{\tau}^T \mathbf{T} \mathbf{u} + \boldsymbol{\lambda}_N^T (\mathbf{B}_N \mathbf{u} - \mathbf{c}_N) + \boldsymbol{\lambda}_E^T (\mathbf{B}_E \mathbf{u} - \mathbf{c}_E)$$

$$\min_{\mathbf{u}} \sup_{\substack{\boldsymbol{\lambda}_E \in \mathbb{R}^{m_E}, \boldsymbol{\lambda}_N \geq \mathbf{0} \\ \|\boldsymbol{\tau}_i\| \leq \Psi_i, i=1, \dots, m_C}} L(\mathbf{u}, \boldsymbol{\lambda}_N, \boldsymbol{\lambda}_E, \boldsymbol{\tau}) = \max_{\substack{\boldsymbol{\lambda}_E \in \mathbb{R}^{m_E}, \boldsymbol{\lambda}_N \geq \mathbf{0} \\ \|\boldsymbol{\tau}_i\| \leq \Psi_i, i=1, \dots, m_C}} \min_{\mathbf{u}} L(\mathbf{u}, \boldsymbol{\lambda}_N, \boldsymbol{\lambda}_E, \boldsymbol{\tau})$$

$$\Lambda(\Psi) = \{(\boldsymbol{\lambda}_E^T, \boldsymbol{\lambda}_N^T, \boldsymbol{\tau}^T)^T \in \mathbb{R}^{m_E + m_C + 2m_C} : \boldsymbol{\lambda}_N \geq \mathbf{0}, \|\boldsymbol{\tau}_i\| \leq \Psi_i, i = 1, \dots, m_C\}$$

$$\min \frac{1}{2} \boldsymbol{\lambda}^T \tilde{\mathbf{F}} \boldsymbol{\lambda} - \boldsymbol{\lambda}^T \tilde{\mathbf{d}} \quad \text{s.t.} \quad \boldsymbol{\lambda} \in \Lambda(\Psi) \quad \text{and} \quad \mathbf{G} \boldsymbol{\lambda} = \mathbf{e}$$

$$\begin{aligned} \tilde{\mathbf{F}} &= \mathbf{B} \mathbf{K}^+ \mathbf{B}^T, \\ \tilde{\mathbf{G}} &= \mathbf{R}^T \mathbf{B}^T, \\ \tilde{\mathbf{d}} &= \mathbf{B} \mathbf{K}^+ \mathbf{f} - \mathbf{c}, \end{aligned} \quad \boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_E \\ \boldsymbol{\lambda}_N \\ \boldsymbol{\tau} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_E \\ \mathbf{B}_N \\ \mathbf{T} \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} \mathbf{c}_E \\ \mathbf{c}_N \\ \mathbf{0} \end{bmatrix}.$$

$$\mathbf{K} \mathbf{u} - \mathbf{f} + \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{0}$$

Relation between primal
and dual variables

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

Final step

$$\min \frac{1}{2} \lambda^T (\mathbf{PFP} + \rho \mathbf{Q}) \lambda - \lambda^T \mathbf{Pd} \quad \text{s.t.} \quad \mathbf{G}\lambda = \mathbf{o} \quad \text{and} \quad \lambda \in \tilde{\Lambda}(\Psi),$$

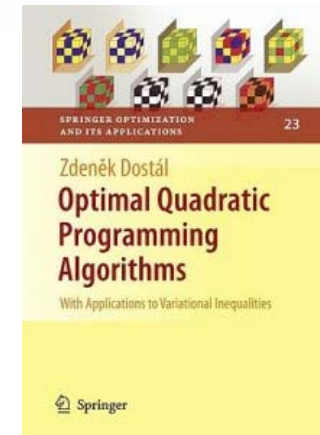
where ρ is an arbitrary positive constant and

$$\mathbf{Q} = \mathbf{G}^T \mathbf{G} \quad \text{and} \quad \mathbf{P} = \mathbf{I} - \mathbf{Q}$$

denote the orthogonal projectors onto the image space of \mathbf{G}^T and onto the kernel of \mathbf{G} , respectively.

The solutions of the discretized model problem with $H/h \leq C$ and a given relative precision may be obtained by SMALSE/MPGP at $O(1)$ matrix / vector multiplications.

Z. Dostal, T. Kozubek, An optimal algorithm and superrelaxation for minimization of a quadratic function subject to separable convex constraints with applications. **Mathematical Programming**, 2012.



REMARK. The Tresca friction is a simple friction law which violates some natural physical principles, but it can be used to define a mapping whose fixed point is a solution to the problem with the Coulomb friction.

Algorithm SMALSE-M

$$\beta < 1, \rho > 0, M_0 > 0, \eta > 0, \mu^0$$

{**Approximate solution** of separably constrained problems }

Step 1 Find \mathbf{x}^k such that $\|\mathbf{g}^P(\mathbf{x}^k, \mu^k, \rho)\| \leq \min \{M_k \|\mathbf{B}\mathbf{x}^k\|, \eta\}$

{**Update Lagrange multipliers**}

Step 2 $\mu^{k+1} = \mu^k + \rho \mathbf{B}\mathbf{x}^k$

{**Update M_k** }

Step 3 **if** $L(\mathbf{x}^k, \mu^k, \rho) \leq L(\mathbf{x}^{k-1}, \mu^{k-1}, \rho) + \frac{\rho}{2} \|\mathbf{B}\mathbf{x}^k\|$

then $M_{k+1} = \beta M_k$

else $M_{k+1} = M_k$

Step 4 $k = k + 1$ and return to *Step 1*

MPRGP algorithm

- quadratic programming algorithm
- active set strategy
- adaptive precision control
- expansion of active set by fixed reduced gradient projection
- conjugate gradients for well conditioned auxiliary linear problems
- release by the optimal chopped gradient step

Important features (Z.D., Schoeberl 2005, Z.D. 2009):

R-linear rate of convergence of gradient/projected gradient in bounds on the spectrum

Algebraic EQ+INEQ system

- Force equilibrium

$$\mathbf{K}\mathbf{u} + \mathbf{N}^T \boldsymbol{\lambda}_n + \mathbf{T}_{1T}^T \boldsymbol{\lambda}_{1T} + \mathbf{T}_{2T}^T \boldsymbol{\lambda}_{2T} + \mathbf{f} = \mathbf{0}$$

- Non-penetration

$$\mathbf{N}\mathbf{u} - \mathbf{d} \leq \mathbf{0}$$

$$\boldsymbol{\lambda}_n \geq \mathbf{0}$$

$$\boldsymbol{\lambda}_n^T (\mathbf{N}\mathbf{u} - \mathbf{d}) = 0$$

- Friction (Coulomb)

$$\mathbf{T}_1 \mathbf{u} + \text{diag}(\boldsymbol{\beta}) \boldsymbol{\lambda}_{1T} = \mathbf{T}_2 \mathbf{u} + \text{diag}(\boldsymbol{\beta}) \boldsymbol{\lambda}_{1T} = \mathbf{0}$$

$$\Psi_i := \|(\lambda_{1T}^i, \lambda_{2T}^i)\|_2 - \mathcal{F} |\lambda_n^i| \leq 0$$

$$\boldsymbol{\beta} \geq \mathbf{0}$$

$$\boldsymbol{\Psi}^T \boldsymbol{\beta} = 0$$

- Approaches to EQ+INEQ system

- Penalty method

+ simple

- inaccurate

- unstable

- Constrained QP minimization

+ suitable

- geometrical

for HPC

nonlinearities only

- Primal-dual active set strategy (PDASS)

+ for general

- nonsymmetrical

nonlin.

system matrix

- Interior point methods

+ for general

- nonsymmetrical

nonlin.

system matrix

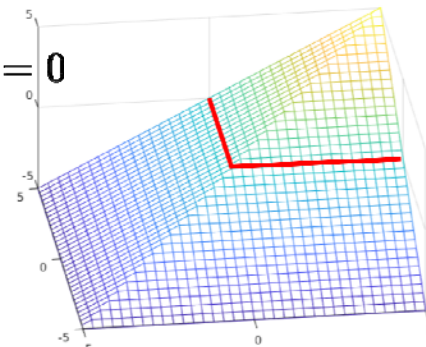
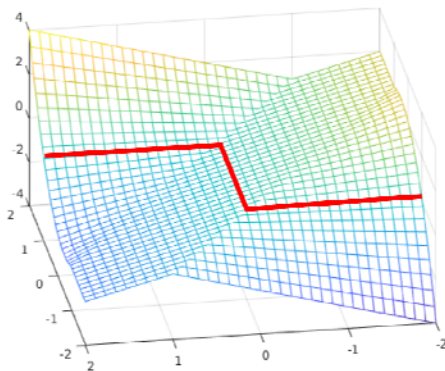
PDASS: SemiSmooth Newton M.

- Write inequality constraints as level 0 of nonsmooth function

$$a \geq 0, b \geq 0, ab = 0 \quad \text{nonpenetration}$$

\Updownarrow

$$C(a, b) := a - \max\{0, a - b\} = 0$$



Tresca friction

$$g - |a| \geq 0, b - \beta a = 0, \beta \geq 0, \beta(g - |a|) = 0$$

\Updownarrow

$$D(a, b) := \max\{g, |a + \alpha b|\} a - g(a + \alpha b) = 0$$

- Solve equality system by SSNM

$$\begin{bmatrix} \mathbf{K}_{NN} & \mathbf{K}_{NM} & \mathbf{K}_{NS} & \mathbf{K}_{Nd} & 0 & 0 \\ \mathbf{K}_{MN} & \mathbf{K}_{MM} & \mathbf{K}_{MS} & \mathbf{K}_{Md} & -\mathbf{M}_S^T & -\mathbf{M}_d^T \\ \mathbf{K}_{SN} & \mathbf{K}_{SM} & \mathbf{K}_{SS} & \mathbf{K}_{Sd} & \mathbf{D}_S & 0 \\ \mathbf{K}_{dN} & \mathbf{K}_{dM} & \mathbf{K}_{dS} & \mathbf{K}_{dd} & 0 & \mathbf{D}_d \\ 0 & 0 & 0 & 0 & \mathbf{I}_S & 0 \\ 0 & \tilde{\mathbf{M}}_d & \tilde{\mathbf{S}}_{dS} & \tilde{\mathbf{S}}_{dd} & 0 & 0 \\ 0 & 0 & \tilde{\mathbf{F}}_{dS} & \tilde{\mathbf{F}}_{dd} & 0 & \mathbf{T}_d \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_N \\ \Delta \mathbf{u}_M \\ \Delta \mathbf{u}_S \\ \Delta \mathbf{u}_d \\ \lambda_S \\ \lambda_d \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_N \\ \mathbf{r}_M \\ \mathbf{r}_S \\ \mathbf{r}_d \\ 0 \\ \tilde{\mathbf{g}}_d \\ 0 \end{bmatrix}$$

(nonpenetration only)

Initialize $(\mathbf{u}^0, \boldsymbol{\lambda}^0)$ and $\mathcal{A}^0, \mathcal{J}^0$

Find $(\Delta \mathbf{u}^k, \boldsymbol{\lambda}^{k+1})$ solving

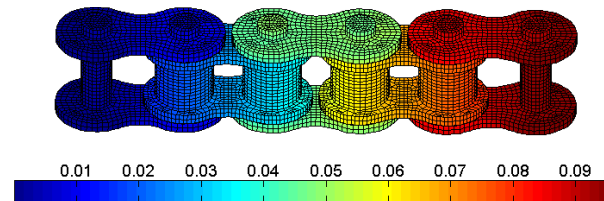
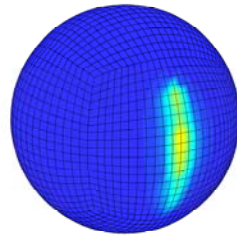
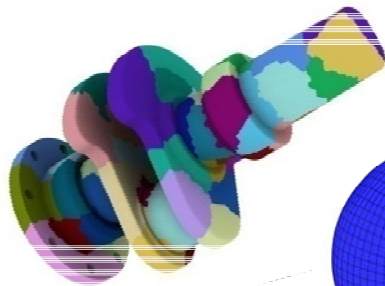
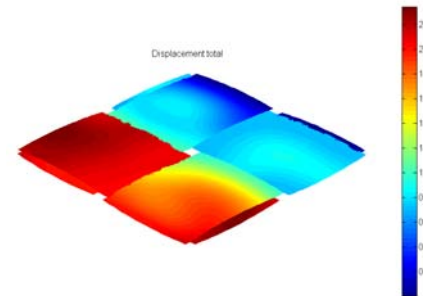
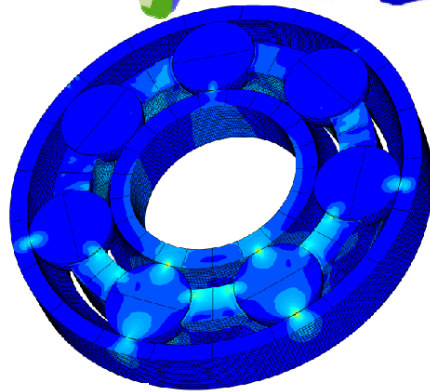
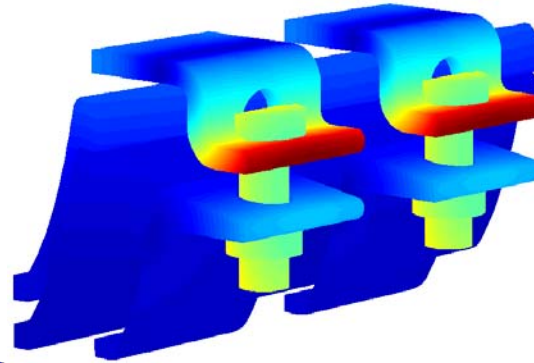
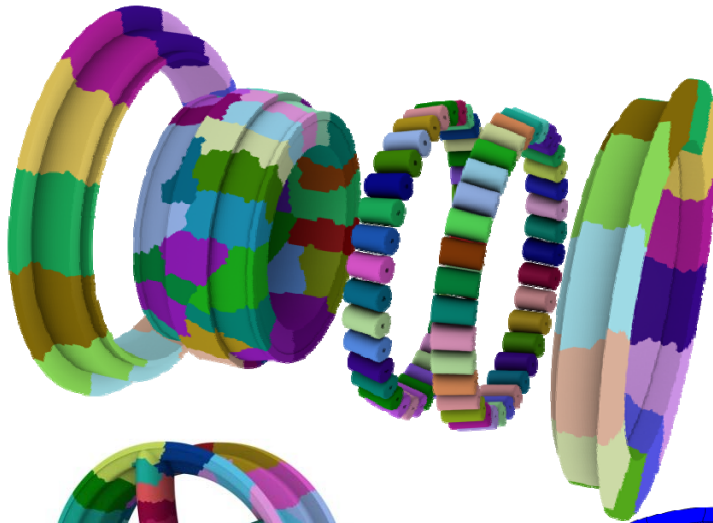
Update $\mathbf{u}^{k+1} = \mathbf{u}^k + \Delta \mathbf{u}^k$ and $\mathcal{A}^{k+1}, \mathcal{J}^{k+1}$

Repeat until

$$\mathcal{A}^{k+1} = \mathcal{A}^k, \mathcal{J}^{k+1} = \mathcal{J}^k, \|\mathbf{r}^{k+1}\| \leq \varepsilon$$

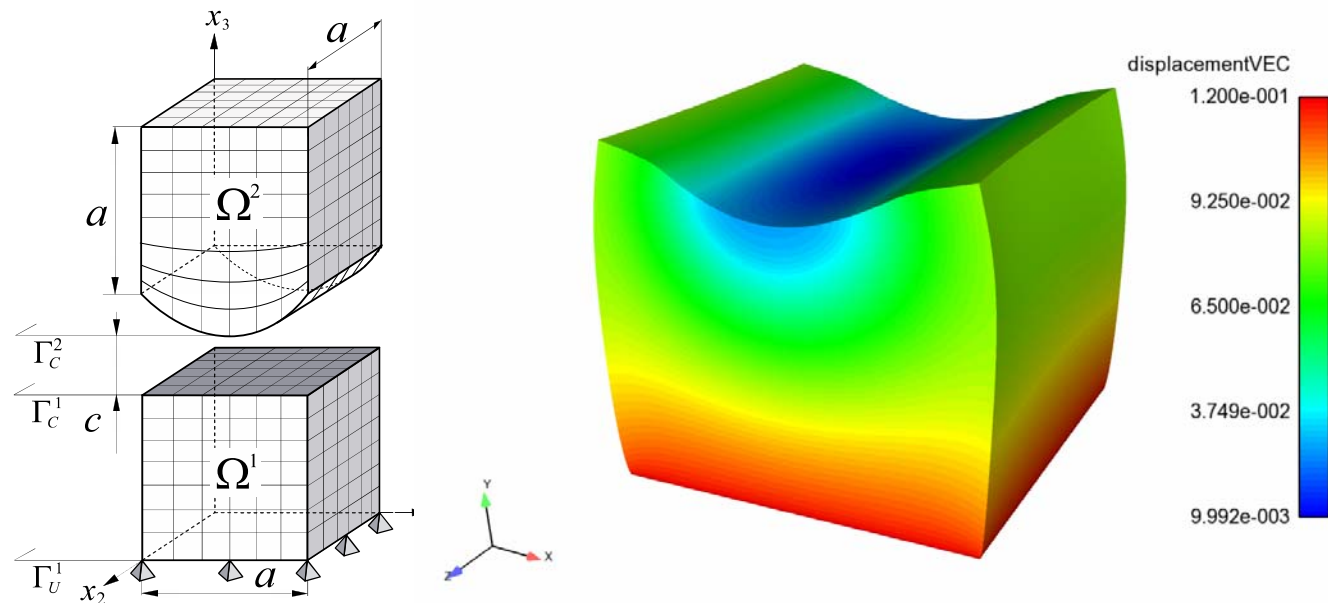
Benchmarks

IT4Innovations
national
supercomputing
center



Contact problems of mechanics

IT4Innovations
national
supercomputing
center

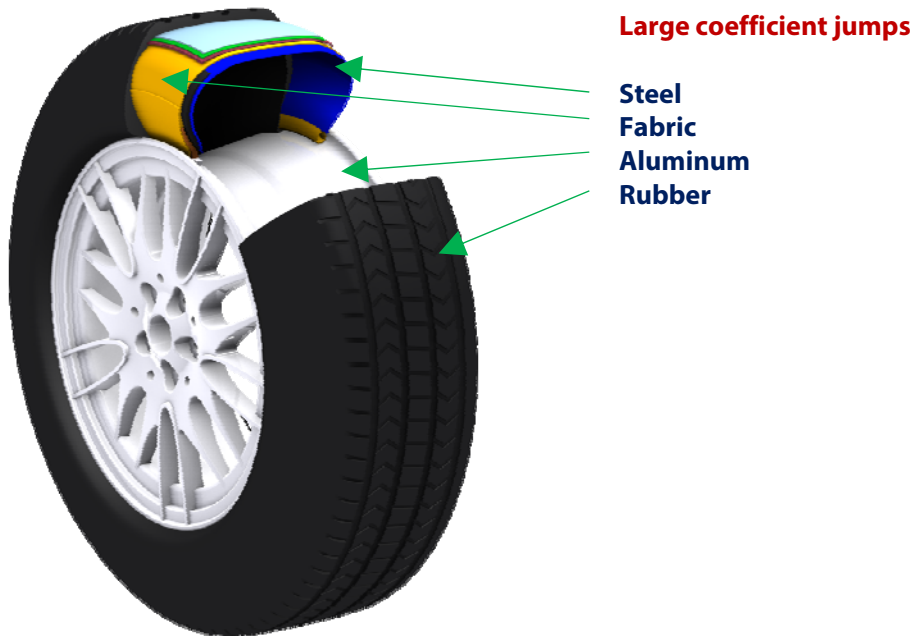


Subdomains	64	1 728	8 000	13 824	21 952	64 000
Number of DOFs	5 719 872	154 436 544	714 984 000	1 235 492 352	1 961 916 096	5 719 872 000
Number of Hessians	73	83	117	108	142	149
CG steps	41	35	33	28	16	17
Propor. steps	1	1	1	1	1	1
Expans. steps	15	23	41	39	62	65
Solution time [s]	219	249	351	324	426	447

● ESPRESO Contact

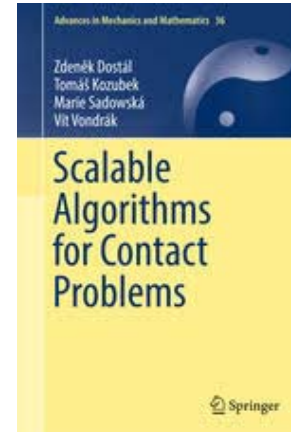
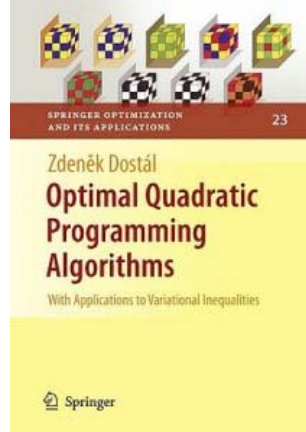
Tire Rim assembly
Geometrically nonlinear problem - contact with rigid roadway

IT4Innovations – SALOMON
Supercomputer



References

1. <http://espresso.it4i.cz>
2. Riha, Lubomir; Brzobohaty, Tomas; Markopoulos, Alexandros; Meca, Ondrej; Kozubek, Tomas: Massively Parallel Hybrid Total FETI (HTFETI) Solver, Platform for Advanced Scientific Computing Conference, PASC, ACM, 2016, ISBN: 978-1-4503-4126-4/16/06.
3. Riha, Lubomir; Brzobohaty, Tomas; Markopoulos, Alexandros; Meca, Ondrej; Kozubek, Tomas; Schenk, Olaf; Vanroose, Wim: Efficient Implementation of Total FETI Solver for Graphic Processing Units Using Schur Complement, HPCSE 2015, LNCS 9611 2016.
4. Riha, Lubomir; Brzobohaty, Tomas; Markopoulos, Alexandros: Hybrid parallelization of the Total FETI solver, Advances in Engineering Software, pp. -, 2016, ISSN: 0965-9978.



Thank you

ESPRESSO Library

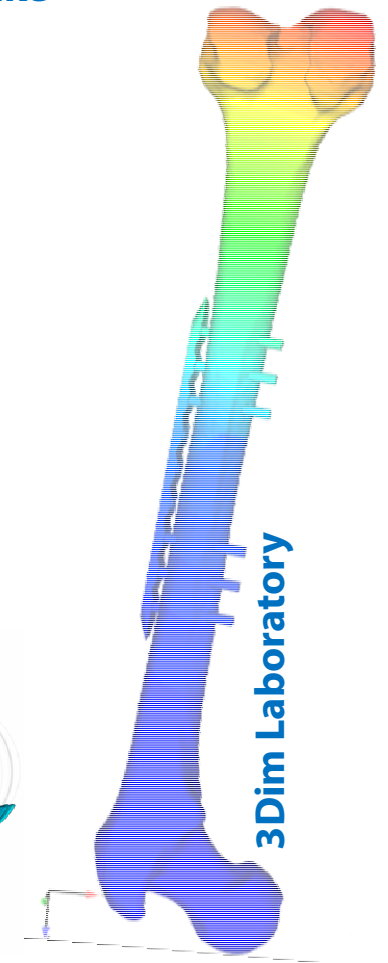
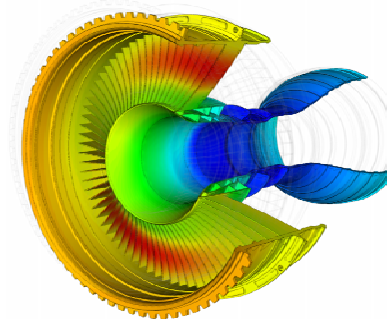
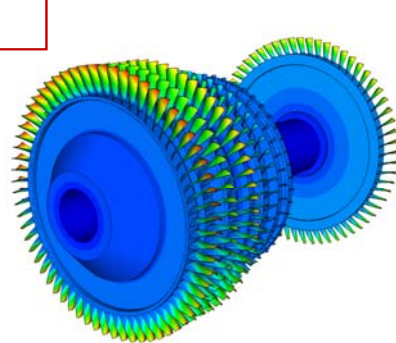
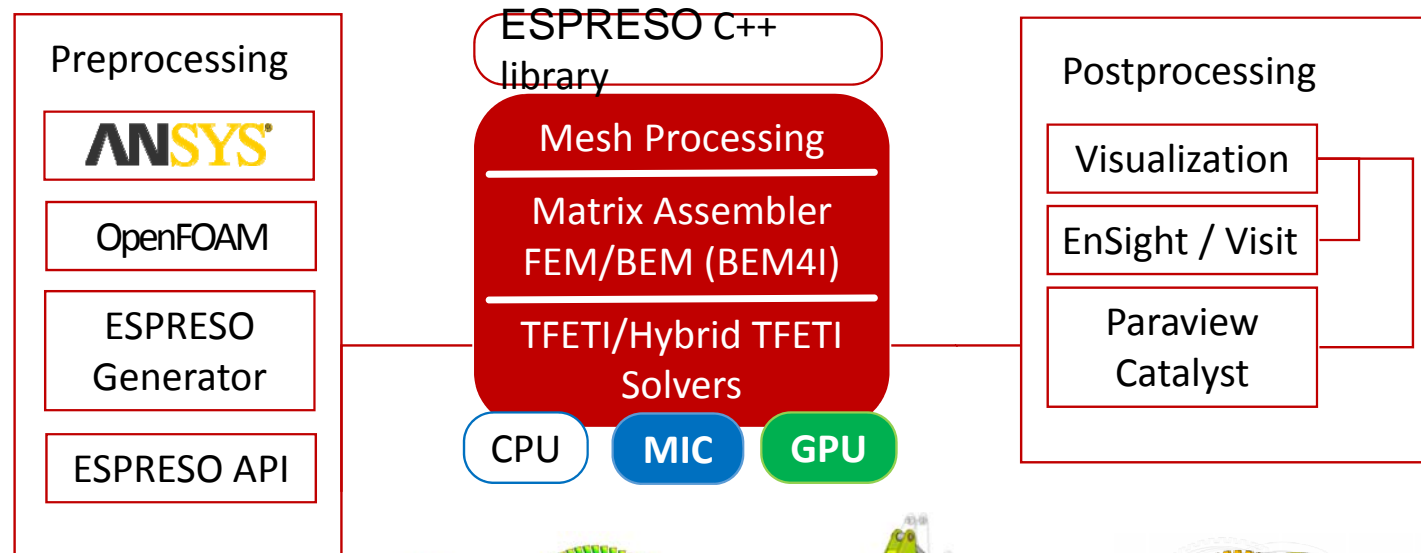
T. Kozubek, T. Brzobohatý, L. Říha, and A. Markopoulos
IT4Innovations, VŠB-Technical University of Ostrava
Ostrava, Czech Republic



ESPRESSO Library

IT4Innovations
national
supercomputing
center

Library based on FEM/BEM with Massively parallel sparse linear solver designed to take full advantage of today's most powerful peta-scale supercomputers



● ESPRESSO Solver

IT4Innovations
national
supercomputing
center

symmetric, non-symmetric, semidefinite, indefinite and positive definite systems

- FETI with core oversubscription
- FETI with hybrid parallelization
- Hybrid FETI
 - clusterization by classical corners
 - clusterization by local kernels
- Preconditioning
 - orthogonal projector
 - conjugate projector for dynamic analysis
 - LUMPED
 - DIRICHLET
 - LIGHT DIRICHLET
 - scaling for coefficient jumps
 - GENE0
- Iterative solvers
 - PCG, Pipelined PCG
 - GMRES, BiCGStab
 - Full Orthogonal PCG
 - SMALSE - **semi-monotonic augmented Lagrangian method with separable convex constraints and general equality constraints**
 - **HYPRE interface (AMG)**
- Automatic kernel detection
- Averaging
 - Transformation of basis (equivalent to local kernel clusterization, but different in performance)

ESPRESSO FEM library

IT4Innovations
national 0\$%&&@
supercomputing
center \$@00&1@%

Finite/boundary element library for engineering applications

- Mesh Interface
 - internal mesh processing for self testing and benchmarking
 - ANSYS input database file
 - OpenFOAM mesh format
 - domain decomposition using third party tools – METIS and ParMETIS
- PreProcessing/solver setting
 - simple configuration file
 - C based API (proof of concept – CSC ELMER)
- FEM/BEM discretization
 - Heat transfer, Elasticity, advection diffusion problems
 - Transient, implicit solvers, stabilizations of advection diffusion problems
 - wide range of engineering boundary conditions types
 - Isotropic, orthotropic and anisotropic materials,...
- PostProcessing
 - Asynchronous parallel I/O (CzeBaCa)
 - paraView vtk file format
 - Enight case file format

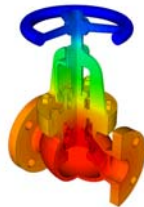
ESPRESSO FEM Highly parallel finite element package for engineering simulations

Heat Transfer Module Capability List:

Load steps definition for combination of multiple steady-state and time dependent analyses

Transient solvers

- Generalized trapezoidal rule
- Automatic time stepping based on response frequency approach



Nonlinear solvers

- Newton Raphson – full and symmetric
- Newton Raphson with constant tangent matrices
- Line search damping
- Sub-steps definition
- Adaptive precision control for iterative solvers

Linear and quadratic finite element discretization

Gluing nonmatching grids by mortar discretization techniques



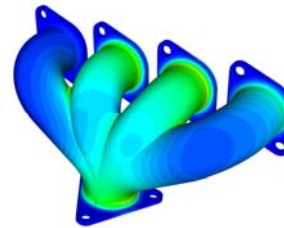
Full-fledged material models

- nonlinear materials
- isotropic, orthotropic and anisotropic material models
- materials for phase change

Element coordinate system definition – cartesian, polar and spherical

Temperature and time dependent boundary conditions

- linear convection
- nonlinear convection
- heat flow
- heat flux
- diffuse radiation
- heat source
- translation motion



Consistent SUPG and CAU stabilization for Translation Motion (advection), Inconsistent stabilization

Phase Change based on apparent heat capacity method

Boundary element discretization for selected physical applications

Highly parallel multilevel FETI domain decomposition based solver for billions of unknowns for symmetric and non-symmetric systems with accelerators support and combination of MPI and OpenMP techniques

Asynchronous parallel I/O

Input mesh format from popular open source and commercial packages like OpenFOAM, ELMER or ANSYS

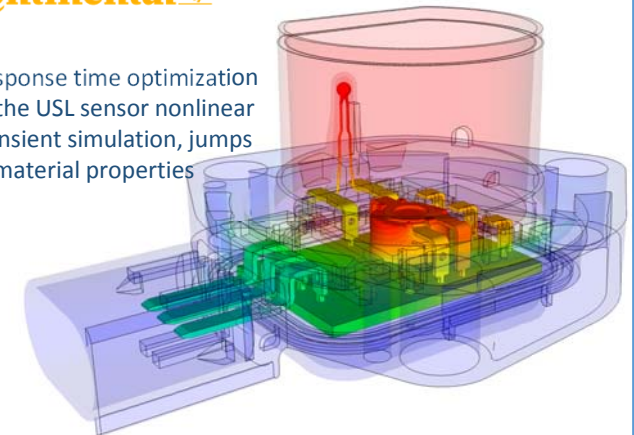
Output to commonly used post-processing formats, VTK and EnSight

Monitoring results on selected regions for statistic and optimization toolchain

Simple text Espresso Configuration File (ecf) for setting all ESPRESSO FEM solver parameters without GUI. Control each parameter in ecf file from command line



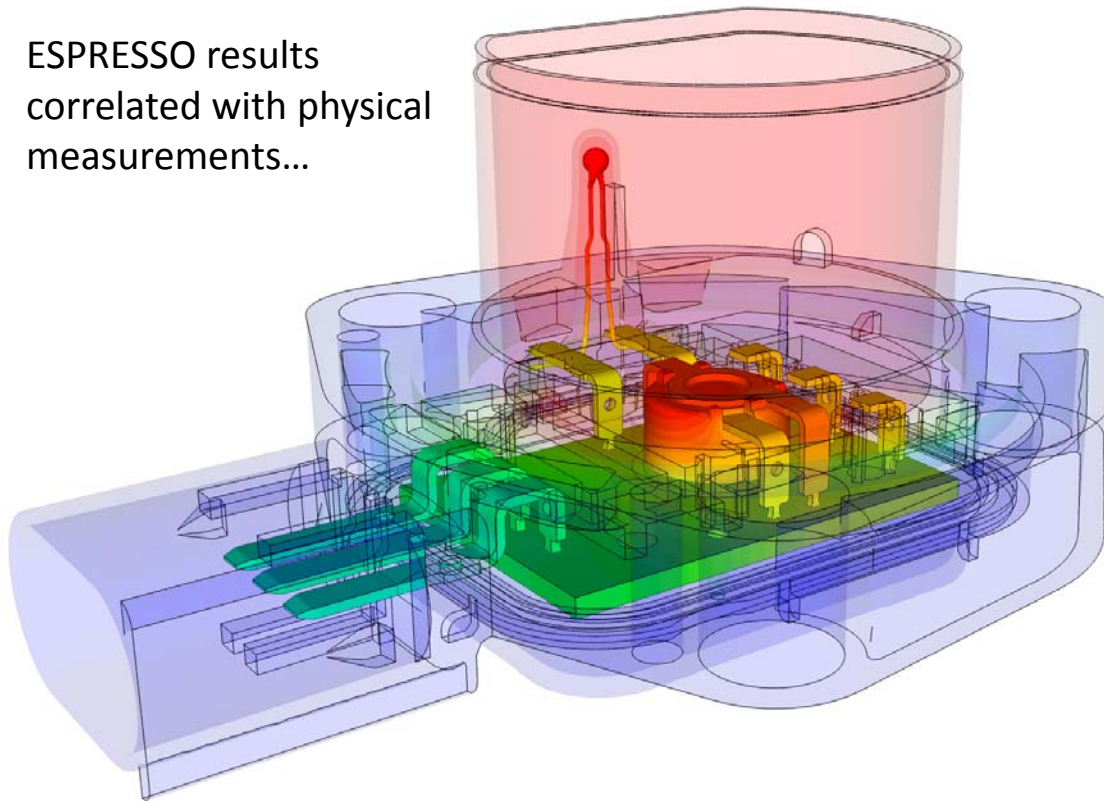
Response time optimization of the USL sensor nonlinear transient simulation, jumps in material properties



ULTRASONIC OIL SENSOR

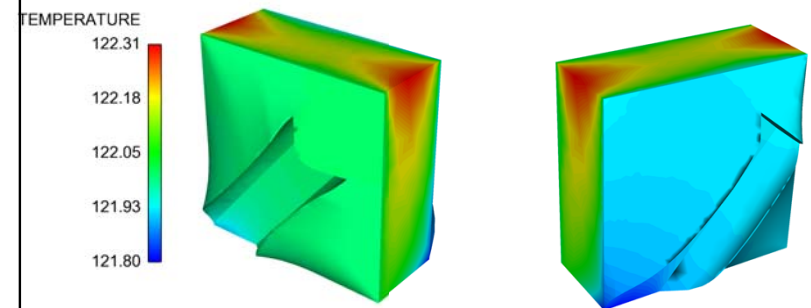


ESPRESSO results
correlated with physical
measurements...

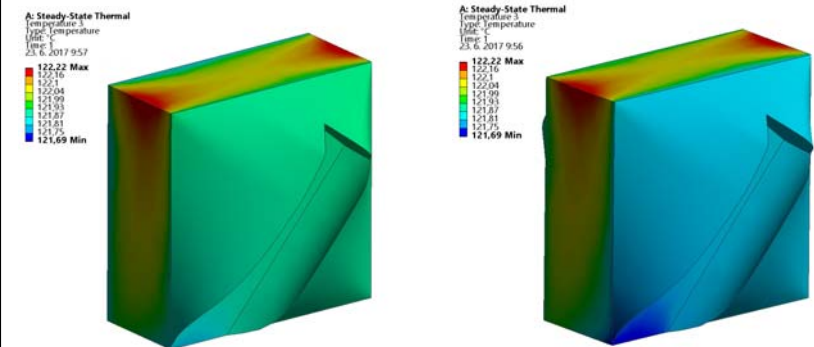


... and we obtain the same results as in Commercial code

ESPRESSO



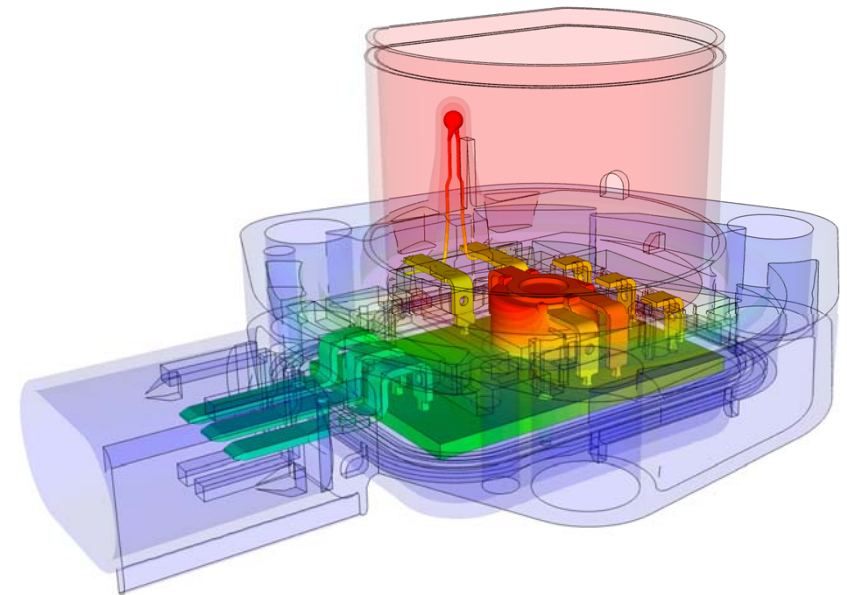
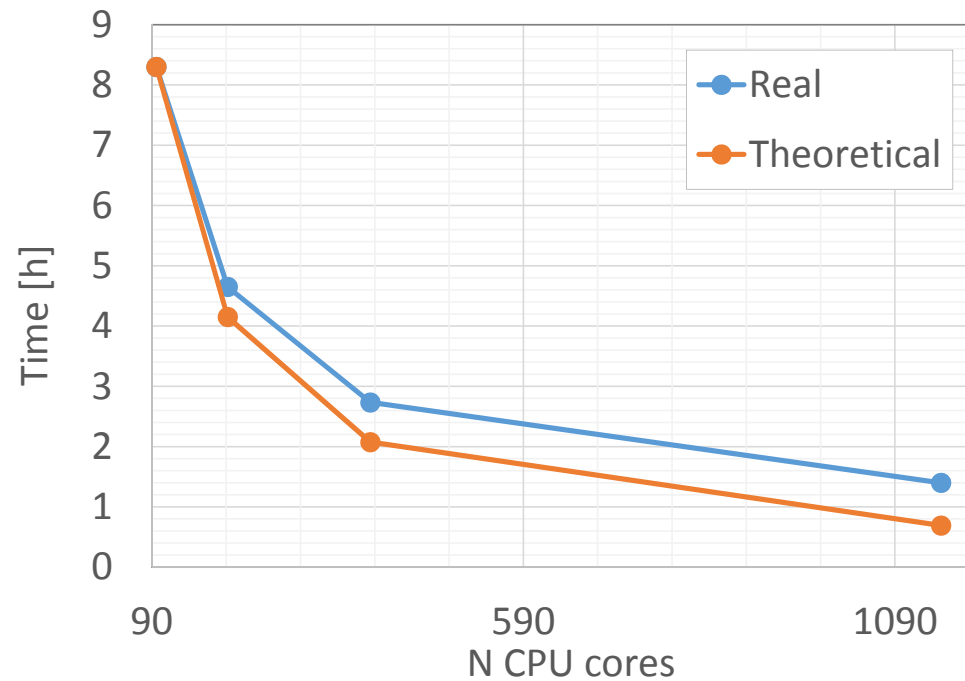
Commercial Code



ULTRASONIC OIL SENSOR



Parallel scalability of nonlinear transient solver - FETI – 1500 time steps
10 M unknowns - 33 h on 24 cores, 1.4 h on 1152



● ESPRESSO Implementation Details

IT4Innovations
national
supercomputing
center

Implementation in C++

- start – 2014 ~ 65 000 lines
- CPU/GPU/MIC version
- int32 (default) and int64 bit version
 - for problems larger than ~2 billion DOF
- Intel compiler preferred
- GCC also supported

Dependencies

- Math library - Intel MKL – required for sparse BLAS
- PARDISO sparse direct solver (both the original version and the MKL versions are supported)

Parallelization tools and strategies

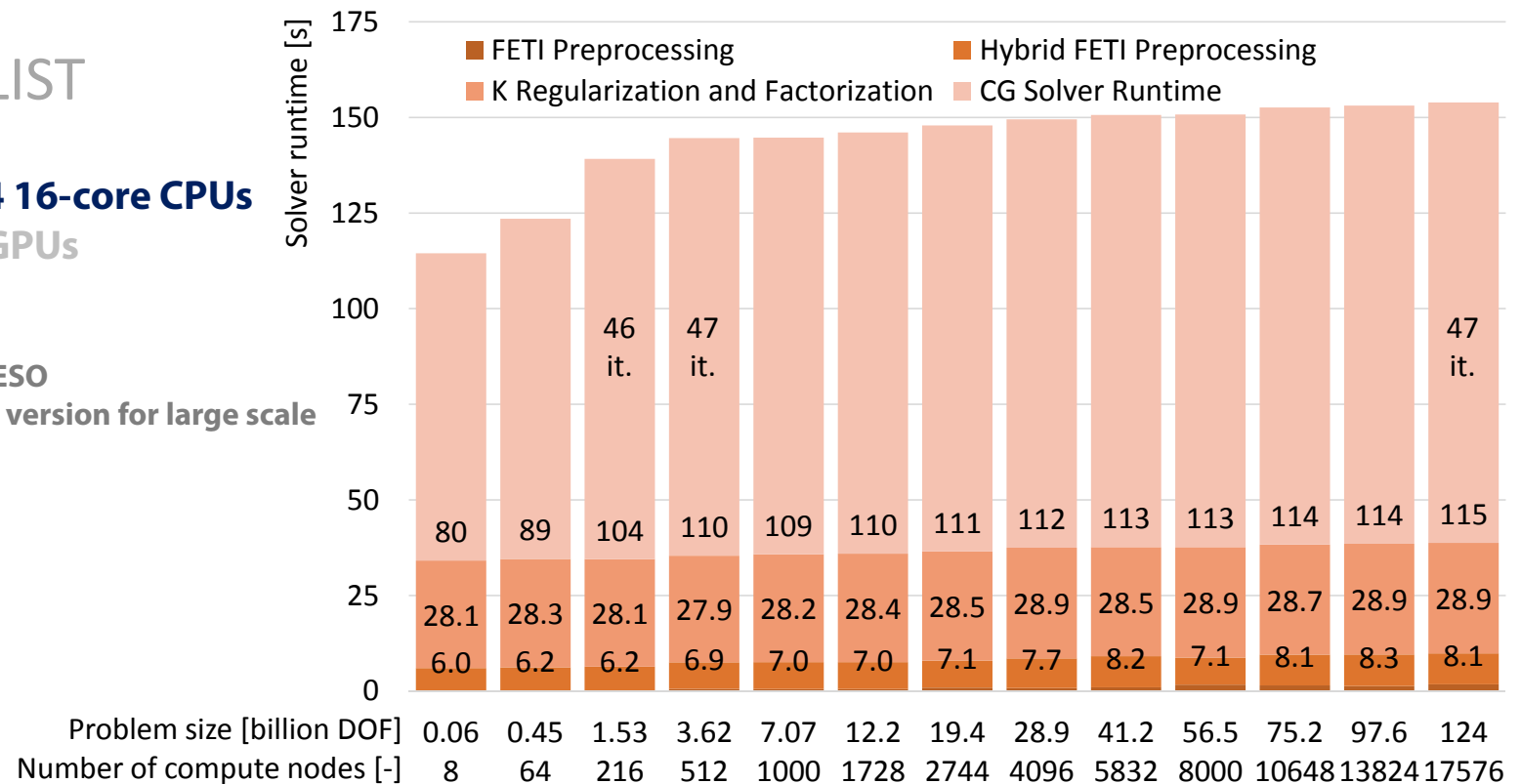
- hybrid parallelization for multi-socket, multicore compute nodes
- distributed memory parallelization – MPI
- shared memory parallelization – using OpenMP
- vectorization using Intel MKL and compiler



Weak Scalability Test

18,688 AMD Opteron 6274 16-core CPUs
18,688 Nvidia Tesla K20X GPUs

- **scalability optimization of ESPRESO**
- **optimization of GPU accelerated version for large scale problems**



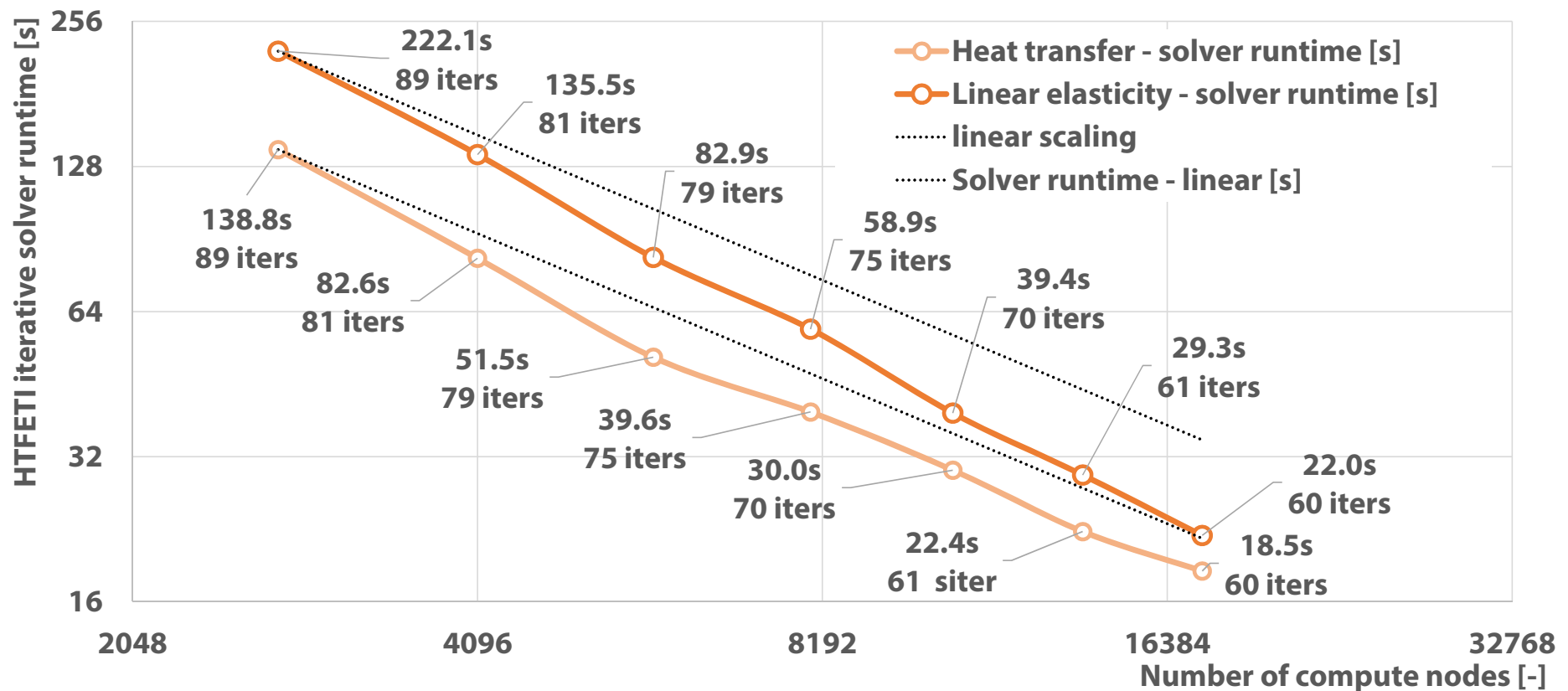
Strong Scalability Test

IT4Innovations
national
supercomputing
center

Heat transfer 20 billion DOF on up to 17 576 Compute Nodes (281 216 cores)

Linear elasticity 11 billion DOF

ORNL Titan 3rd in TOP500 LIST



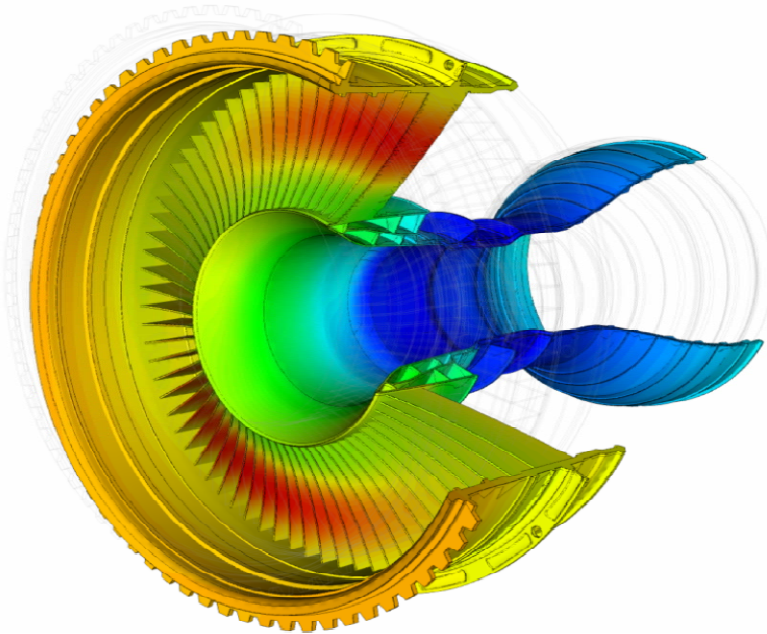
Scalability for Real World Problems

IT4Innovations
national
supercomputing
center

300 million unknown - ANSYS Workbench real world problem

Linear elasticity – Hybrid FETI with Dirichlet preconditioner

IT4Innovations – SALOMON Supercomputer

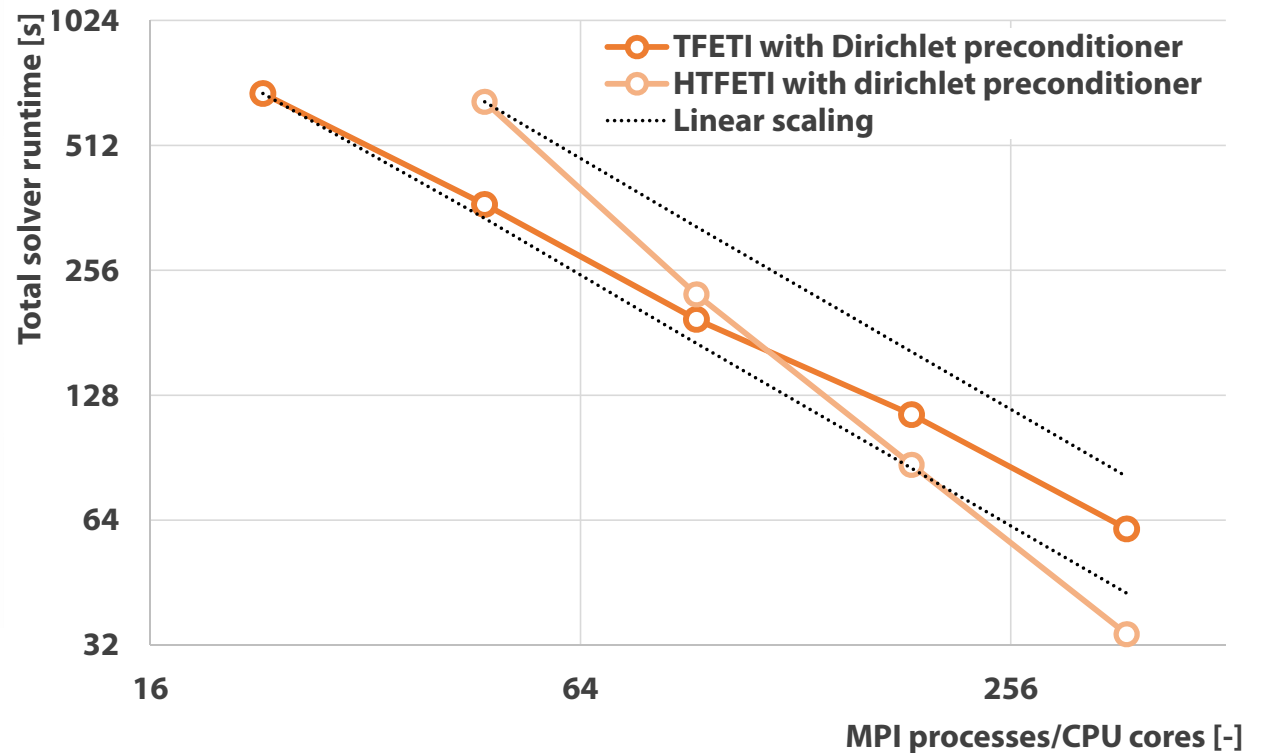


Experiment setup:

Dirichlet preconditioner

regular CG solver

HTFETI method with clusterization by corners





ESPRESSO CG in FETI

Projected Conjugate Gradient in FETI

```

1:  $r_0 := b - Ax_0$ ;  $u_0 := M^{-1}r_0$ ;  $p_0 := u_0$ 
2: for  $i = 0, \dots, m - 1$  do
3:    $s := Ap_i$ 
4:    $\alpha := \langle r_i, u_i \rangle / \langle s, p_i \rangle$ 
5:    $x_{i+1} := x_i + \alpha p_i$ 
6:    $r_{i+1} := r_i - \alpha s$ 
7:    $u_{i+1} := M^{-1}r_{i+1}$ 
8:    $\beta := \langle r_{i+1}, u_{i+1} \rangle / \langle r_i, u_i \rangle$ 
9:    $p_{i+1} := u_{i+1} + \beta p_i$ 
10: end for
  
```

**Pre-processing - $S_c = B_1 K^{-1} B_1^T \rightarrow$
GPU/MIC**

```

1.)  $\lambda \rightarrow$  GPU/MIC - PCIe transfer from CPU
2.)  $\lambda = S_c \cdot \lambda$  - DGEMV, DSYMV on GPU/MIC
3.)  $\lambda \leftarrow$  GPU/MIC - PCIe transfer to CPU
4.) stencil data exchange in  $\lambda$ 
  
```

Pre-processing – K factorization

```

1.)  $x = B_1^T \cdot \lambda$  - SpMV
2.)  $y = K^{-1} \cdot x$  - solve
3.)  $\lambda = B_1 \cdot y$  - SpMV
4.) stencil data exchange in  $\lambda$ 
   - MPI – Send and Recv
   - OpenMP – shared mem. Vec
  
```

Pre-processing - $S_c = B_1 K^{-1} B_1^T$

```

1.) - nop
2.)  $\lambda = S_c \cdot \lambda$  - DGEMV, DSYMV
3.) - nop
4.) stencil data exchange in  $\lambda$ 
   - MPI – Send and Recv
   - OpenMP – shared mem. Vec
  
```

90 – 95% of runtime spent in Ap_i

Large Scale Tests on Salomon with Xeon Phi

IT4Innovations
national
supercomputing
center

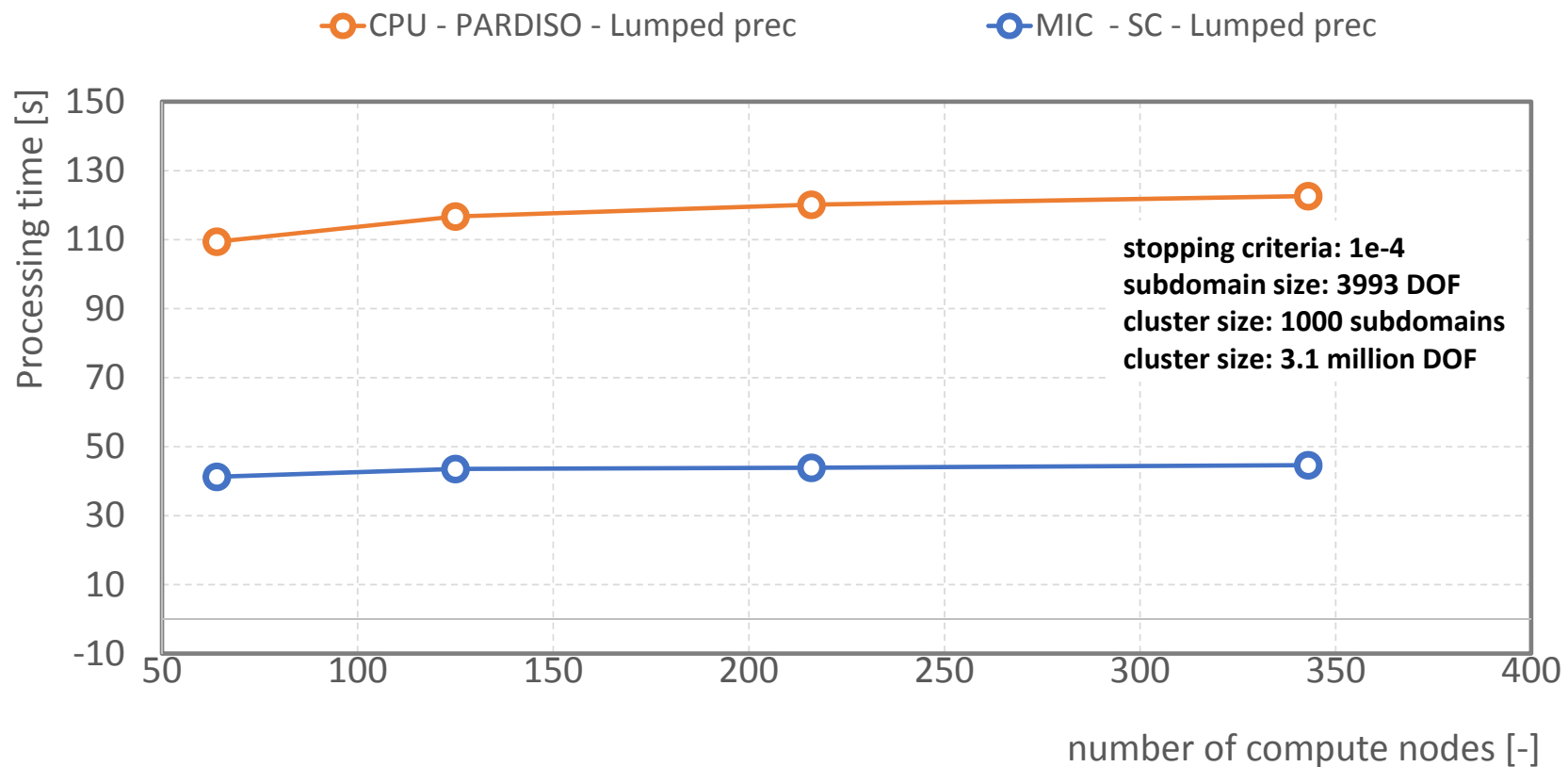
192 - 1033 million DOF Hybrid FETI CG Solver Runtime

Linear elasticity – CG Solver Runtime w. Lumper Prec.

IT4Innovations Salomon Supercomputer

speedU

2.4



GPU acceleration of the ESPRESO Solver

IT4Innovations
national
supercomputing
center

0.3 - 300 million DOF Hybrid FETI CG Solver Runtime

Linear elasticity

ORNL Titan 5rd in TOP500 LIST

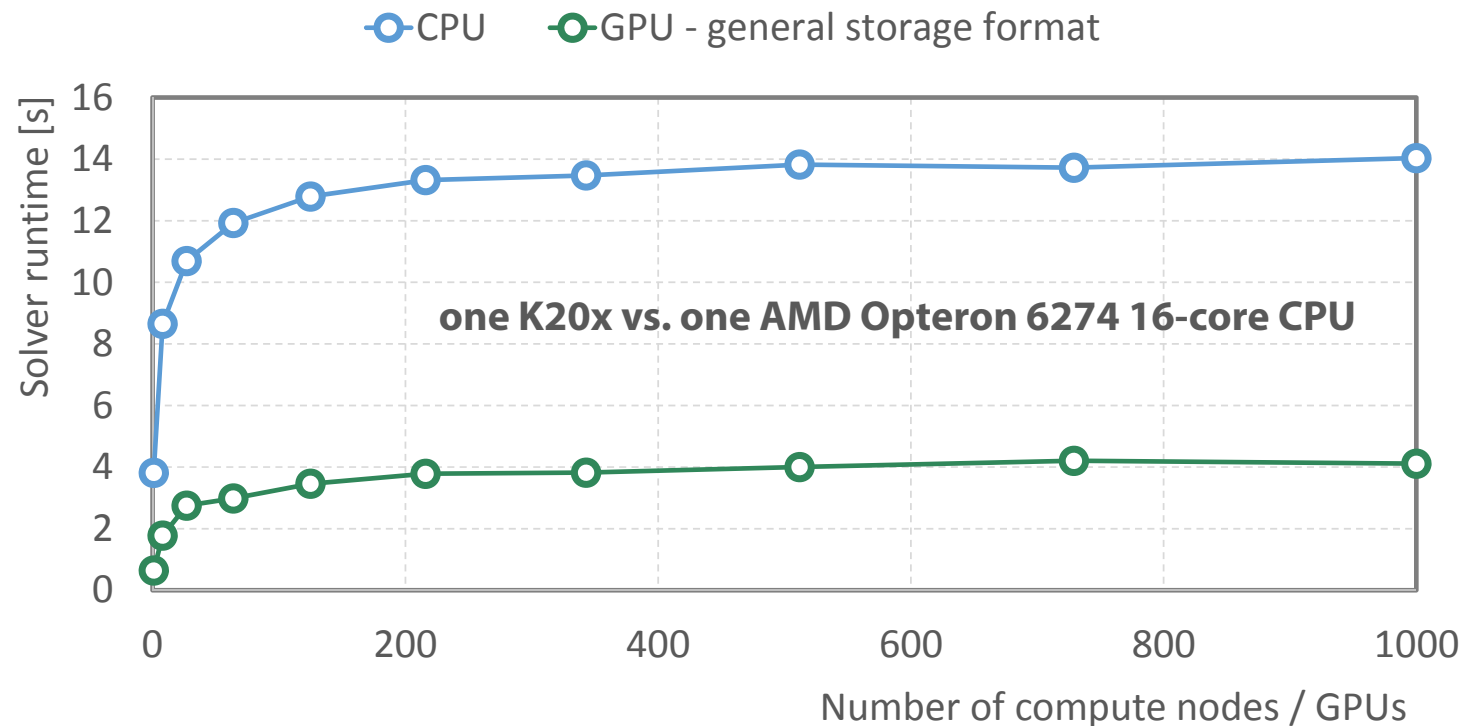
speedU
3.4

Current work:

- support for cuDense and cuSparse solvers from CUDA toolkit
- memory optimization for symmetric local schur complements
- implementation of memory efficient methods

Future work:

- Support for Power8/9 platforms with Pascal/Volta GPUs with NVLink





ESPRESSO Contact Problems

IT4Innovations
national
supercomputing
center

TFETI Weak Scaling – 5.7 Billions Unknowns

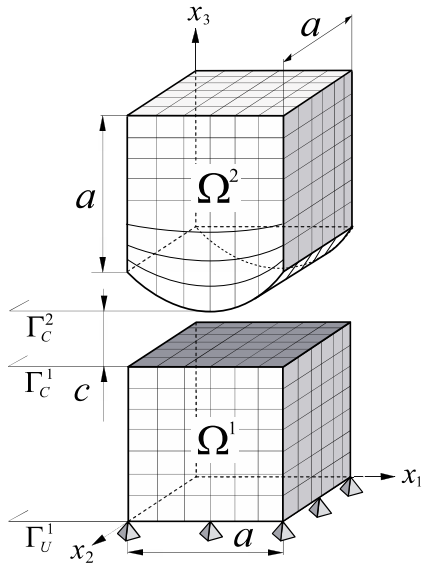
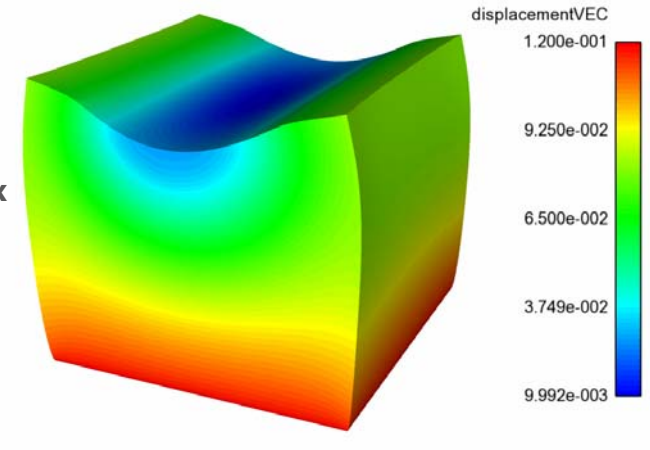
Geometrically nonlinear problem

IT4Innovations – SALOMON

Supercomputer

SMALSE-M - semi-monotonic augmented Lagrangian method with separable convex constraints and general equality constraints

MPGP - modified proportioning with the reduced gradient projection



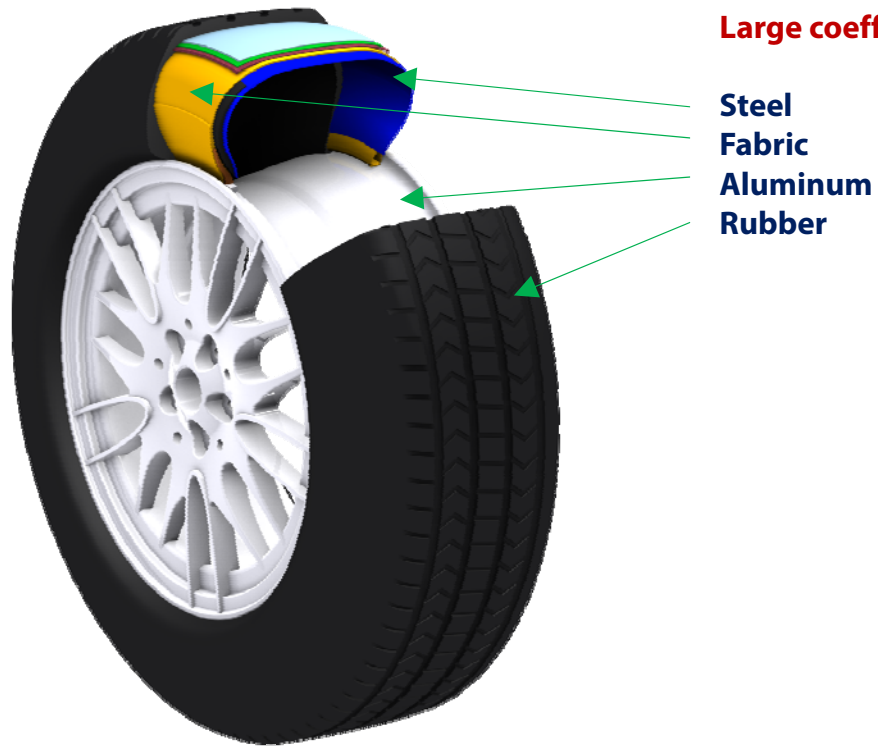
Subdomains	64	1 728	8 000	13 824	21 952	64 000 1 000 MPI ranks 512 nodes
Problem size [billion of DOFs]	0.0057	0.15	0.72	1.24	1.96	5.72
Number of Hessians	73	83	117	108	142	149
CG steps [-]	41	35	33	28	16	17
Proportioning steps [-]	1	1	1	1	1	1
Expansion steps [-]	15	23	41	39	62	65
Solution time [s]	219	249	351	324	426	447

● ESPRESSO Contact Problems

TFETI with QPCE – Real world problem – Tire Rim assembly

Geometrically nonlinear problem - contact with rigid roadway

IT4Innovations – SALOMON
Supercomputer





ESPRESSO Outlook

IT4Innovations
national
supercomputing
center

FEM library

- **Extend structure mechanics module (nonlinear material modules)**
- **Parallel Radial Basis function mesh Morphing (RBF) solver for shape optimization methods**
 - **solver for large FULL Matrices**
- **Topological optimization**
- **Extension of mesh interface – CGNS, Nastran, LS-DYNA**
- **New post-processing features – inSitu real time visualization**
- **Multiphysics simulations (thermo-elasticity)**

Solver library

- **HFETI for transient problems**
- **Support for new architectures**
- **Semi Smooth newton, interior point for contact problems**

And of course EXPERTISE consortium requirements are welcome!!!!

ESPRESSO Future Targets

IT4Innovations
national
supercomputing
center

NEXT-GENERATION U.S. DEPARTMENT OF ENERGY SUPERCOMPUTERS

US Department of Energy CORAL Supercomputers

	Aurora	Theta	Summit
CPU Architecture	Intel Xeon	Intel Xeon	IBM POWER9
Accelerator Architecture	Intel Xeon Phi	Intel Xeon Phi	NVIDIA Volta
Performance (RPEAK)	180 PFLOPS	8.5 PFLOPS	130-300 PFLOPS
Nodes	50,000	N/A	3,400
Laboratory	Argonne	Argonne	Oak Ridge

Next generation of the ESPRESSO solver

- Support fat nodes with multiple accelerators (Power9 and GPU with NVLink)
 - multiple GPUs per node
 - fast CPU/GPU interconnect
 - Oak Ridge Nat. Lab - Summit
- Support extremely large number of thin nodes with KNL/KNH nodes
 - Multi-core machine with multilevel memory hierarchy and NUMA
 - new sparse matrix formats
 - Argonne National Lab - Aurora
- Heterogeneous machines
 - combination of Xeon CPU only islands and Xeon Phi only islands in one machine
 - offload over fabric

Thank you