FE2TI: Computational Scale Bridging for Dual-Phase Steels

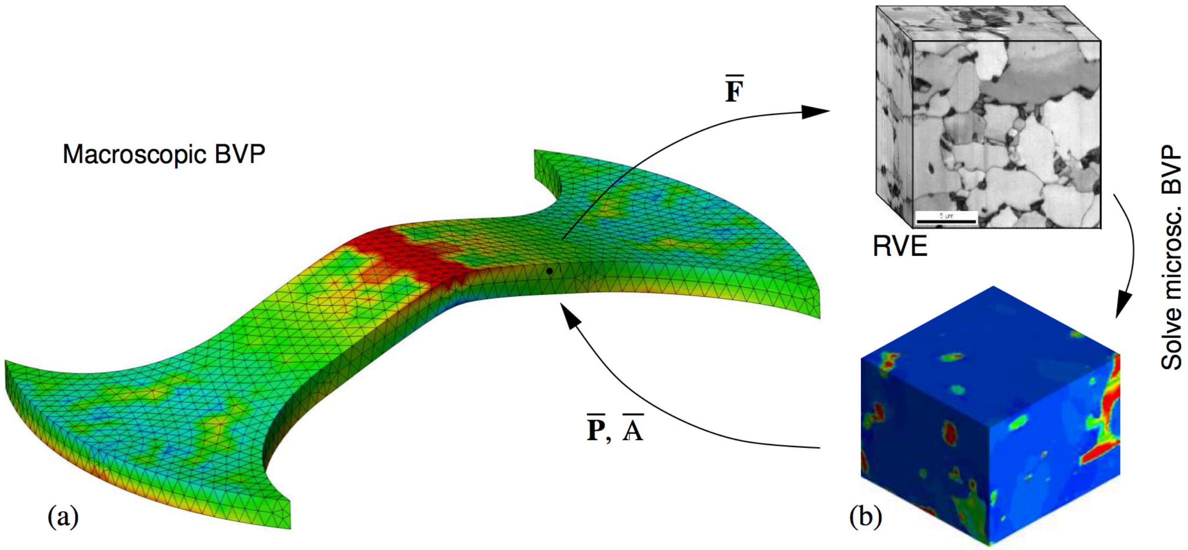
Martin Lanser
Mathematical Institute, University of Cologne

Based on joint work with
Axel Klawonn (University of Cologne)
Oliver Rheinbach (TU Bergakademie Freiberg)

ParCo 2015
Edinburgh
09/01/2015 - 09/04/2015

DFG-Priority Program SPP 1648 - Software for Exascale Computing

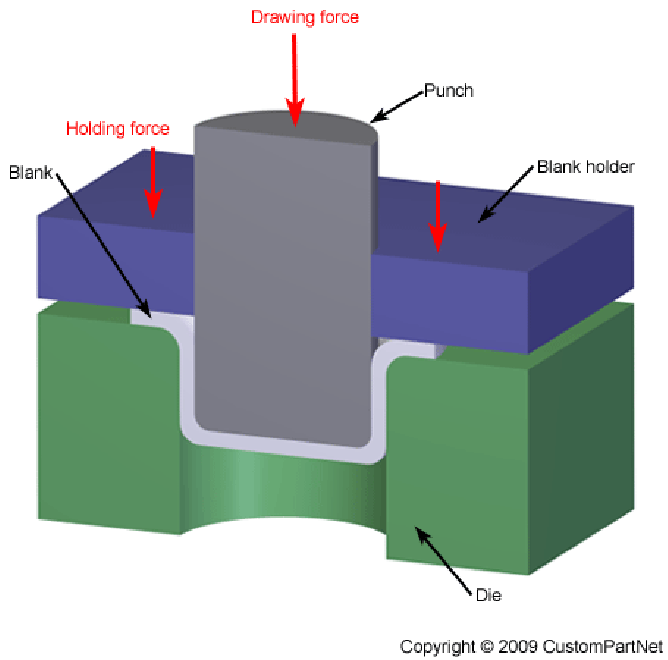
EXASTEEL - Bridging Scales for Multiphase Steels



Principal Investigators

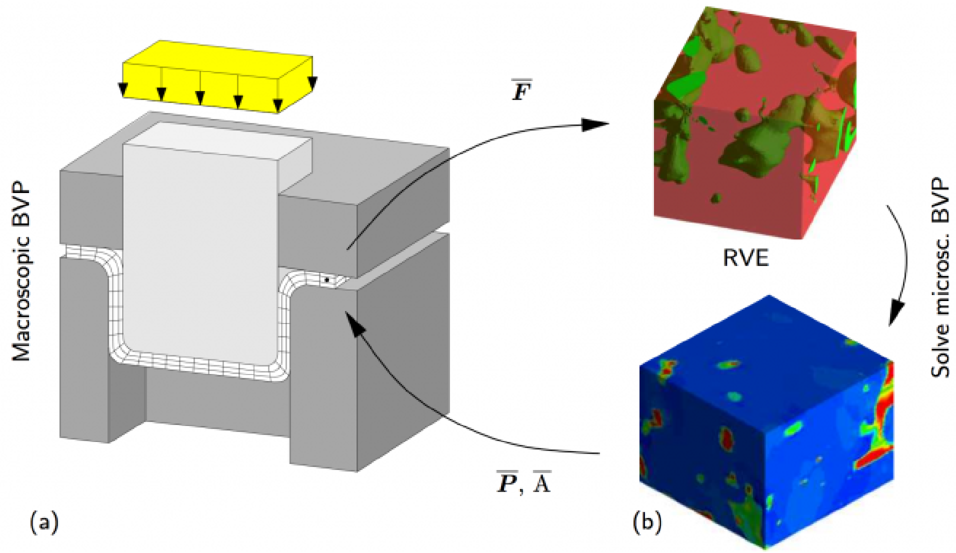
- A. Klawonn, U Cologne
- O. Rheinbach, TU Freiberg
- J. Schröder, U Duisburg-Essen
- D. Balzani, TU Dresden
- G. Wellein, U Nuremberg-Erlangen

- Challenging 3D multiscale problem from nonlinear structural mechanics with plasticity.
- Highly concurrent computational scale bridging in continuum mechanics (FE2)
- Hybrid domain decomposition/multigrid implicit solvers for nonlinear problems
- Software FE2TI is based on PETSc and BoomerAMG



FE^2 - Computational Scale Bridging

- Characteristic lengths (macro/micro): $L/d \approx 10^4$ - 10^6.
- Brute force FE discretization not feasible.
- Scale-bridging procedure is essential.
- FE^2: FE-discretization of both scales, reduces problem size by factor a of 10^3–10^6.



(a) Macroscopic boundary value problem (BVP).
(b) Microscopic BVP (one in each Gauss point).

The averaged results on the microscale replace a macroscopic phenomenological constitutive law:

$$\overline{P} = \frac{1}{V}\int_{\mathcal{B}} P\,dV, \qquad \overline{A} = \frac{\partial \overline{P}}{\partial \overline{F}} = \frac{\partial}{\partial \overline{F}}\left(\frac{1}{V}\int_{\mathcal{B}} P\,dV\right)$$

Miehe, Schröder, Schotte 1999; Schröder 2000; Feyel 1999; Smit, Brekelmans, Meijer 1998; Kouznetsova, Brekelmans, Baaijens 2001

Remaining orders of magnitude are resolved by highly parallel solver algorithms and performance engineering.

FE^2 - Computational Scale Bridging

- The macroscopic problem is **comparable small**, since the microstructure is neglected on this level
- Usage of a **Representative Volume Element (RVE)** in each Gauß integration point, which is able to describe the microstructure of the material
- Only sufficiently large RVEs can resolve the microstructure $\Rightarrow$ Need for fast and **efficient parallel solvers on the RVEs**
- Communication between different RVEs only through (small) macroscopic problem
- Independent solution of nonlinear problems on the RVEs

Two Levels of Parallelism

Level 1:

- FE^2 method decomposes macroscopic problem into many **nonlinear and independent** RVEs
- Each RVE is assigned to its own communicator (Split MPI_COMM_WORLD into subcommunicators)

Level 2:

- Parallel and highly scalable solver on each RVE (on each subcommunicator)
- **FE2TI** software uses domain decomposition methods of the **FETI-DP** type

FE^{2}TI - Algorithmic Overview

We denote by FE^{2}TI the combination of FE^2 scale bridging method and FETI-DP domain decomposition methods used for the RVE solves

Repeat until convergence:

Microscopic calculations:

1. Apply boundary conditions $x = \overline{F}X$ on $\partial\mathcal{B}$. (In case of Dirichlet conditions)
2. Solve microscopic nonlinear boundary value problem **using FETI-DP** or related methods.
3. Compute macroscopic stresses $\overline{P} = \frac{1}{V}\int_{\mathcal{B}} P\, dV$.
4. Compute macroscopic tangent moduli $\overline{A} = \frac{\partial}{\partial \overline{F}}(\frac{1}{V}\int_{\mathcal{B}} P\, dV)$.

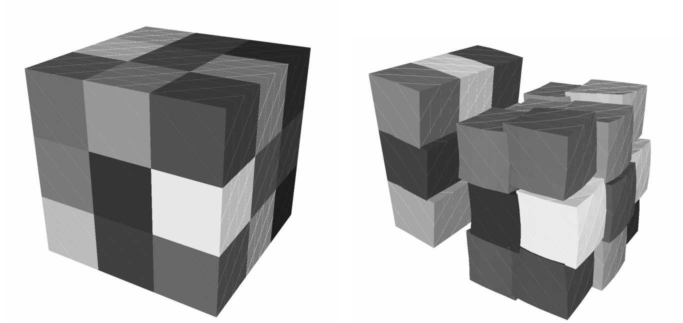
Macroscopic calculations:

5. Set up tangent matrix and rhs of linearized macroscopic BVP using $\overline{P}$ and $\overline{A}$.
6. Solve linearized macroscopic boundary value problem.
7. Update macroscopic deformation gradient $\overline{F}$.

Efficient Parallel RVE Solver: FETI-DP

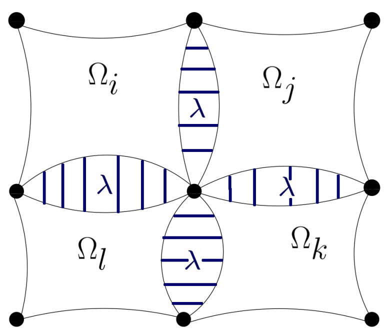
Finite Element Tearing and Interconnecting - Dual-Primal

Divide and Conquer Algorithm: Decompose the RVE into N nonoverlapping subdomains.
FETI-DP coarse space: Strong coupling in some degrees of freedom.



$$\begin{bmatrix} K_{BB}^{(1)} & & & \widetilde{K}_{\Pi B}^{(1)T} \\ & \ddots & & \vdots \\ & & K_{BB}^{(N)} & \widetilde{K}_{\Pi B}^{(N)T} \\ \widetilde{K}_{\Pi B}^{(1)} & \cdots & \widetilde{K}_{\Pi B}^{(N)} & \widetilde{K}_{\Pi\Pi} \end{bmatrix} =: \begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^{T} \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} \end{bmatrix}.$$

Introduce Lagrange multipliers and enforce zero jump between subdomains: $B_B u_B = 0$



$$\begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^{T} & B_B^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} & O \\ B_B & O & O \end{bmatrix} \begin{bmatrix} u_B \\ \tilde{u}_\Pi \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} f_B \\ \tilde{f}_\Pi \\ 0 \end{bmatrix}$$

In compact form:

$$\begin{bmatrix} \widetilde{K} & B^T \\ B & O \end{bmatrix} \begin{bmatrix} \tilde{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \tilde{f} \\ 0 \end{bmatrix}$$

The FETI-DP Algorithm

After reducing to the Lagrange multipliers:

$$F\boldsymbol{\lambda} = d$$

$$F = \underbrace{B_B K_{BB}^{-1} B_B^T}_{\text{local solvers}} + \underbrace{B_B K_{BB}^{-1} \widetilde{K}_{B\Pi} \widetilde{S}_{\Pi\Pi}^{-1} \widetilde{K}_{\Pi B} K_{BB}^{-1} B_B^T}_{\text{coarse problem; coupled!}}.$$

$\mathbf{B_B}$: Communication over the interface.
$\mathbf{K_{BB}^{-1}}$: Local direct solvers.
$\widetilde{\mathbf{S}}_{\mathbf{\Pi\Pi}}^{-1} := \widetilde{\mathbf{K}}_{\mathbf{\Pi\Pi}} - \widetilde{\mathbf{K}}_{\mathbf{\Pi B}} \mathbf{K}_{\mathbf{BB}}^{-1} \widetilde{\mathbf{K}}_{\mathbf{\Pi B}}^{\mathbf{T}}$: Exact solution of a global problem $\Rightarrow$ scaling bottleneck!

The Preconditioner

Preconditioner: $M^{-1} := B_{D,\Delta} S B_{D,\Delta}^T$ (Sum of local operators!)

1. S Schur complement of K (Interior variables eliminated). Local solvers.
2. $B_{D,\Delta}$ appropriately scaled jump operator (scaling depends on pde coeff.)

FETI-DP is PCG solving

$$M^{-1} F \boldsymbol{\lambda} = M^{-1} d$$

Hybrid FETI-DP/Multigrid - Adding a further level of parallelism

Considering the FETI-DP master system

$$\begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^T & B_B^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} & O \\ B_B & O & O \end{bmatrix} \begin{bmatrix} u_B \\ \tilde{u}_\Pi \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} f_B \\ \tilde{f}_\Pi \\ 0 \end{bmatrix}$$

we perform an elimination of u_B, which yields

$$\begin{bmatrix} \widetilde{S}_{\Pi\Pi} & -\widetilde{K}_{\Pi B}\widetilde{K}_{BB}^{-1}B_B^T \\ -B_B\widetilde{K}_{BB}^{-1}\widetilde{K}_{\Pi B}^T & -B_B\widetilde{K}_{BB}^{-1}B_B^T \end{bmatrix} \begin{bmatrix} \tilde{u}_\Pi \\ \boldsymbol{\lambda} \end{bmatrix} = \text{r.h.s.} \quad (1)$$

with $\widetilde{S}_{\Pi\Pi} := \widetilde{K}_{\Pi\Pi} - \widetilde{K}_{\Pi B} K_{BB}^{-1} \widetilde{K}_{\Pi B}^T$.

Exact solution of $\widetilde{S}_{\Pi\Pi}$ not necessary. Solution of coarse problem is moved to the preconditioner $\Rightarrow$ **Inexact solution possible. (See next slide)**.

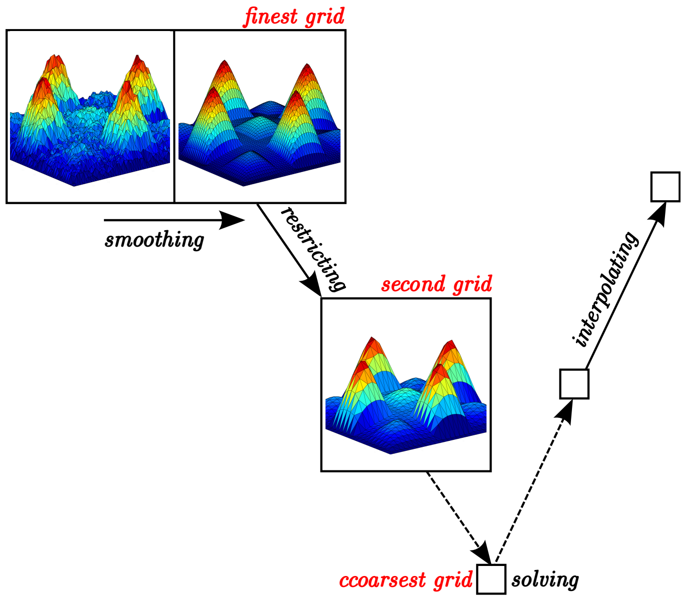
Further elimination of $\tilde{u}_\Pi$ would result in FETI-DP system $F\lambda = d$.

Hybrid FETI-DP/Multigrid

We instead solve (1) iteratively using the block-triangular preconditioner

$$\hat{\mathcal{B}}_{r,L}^{-1} = \begin{bmatrix} \hat{S}_{\Pi\Pi}^{-1} & 0 \\ -M^{-1}B_B K_{BB}^{-1} \widetilde{K}_{\Pi B}^T \hat{S}_{\Pi\Pi}^{-1} & -M^{-1} \end{bmatrix}$$

- M^{-1}: one of the standard FETI-DP preconditioners
- $\hat{S}_{\Pi\Pi}^{-1}$: some cycles of an AMG (algebraic multigrid) method, applied to $\widetilde{S}_{\Pi\Pi}$.
- If $\hat{S}_{\Pi\Pi}^{-1}$ is a good preconditioner of $\widetilde{S}_{\Pi\Pi}$, hybrid FETI-DP/Multigrid has convergence bounds of the same quality as exact FETI-DP.



One V-cycle of an AMG method.

See **Klawonn, Rheinbach (IJNME 2007, ZAMM 2010)** for details.

Newton-Krylov FETI-DP/Multigrid

Classical use of hybrid FETI-DP/Multigrid in the context of nonlinear problems:

For a given nonlinear problem

$$A(u) = 0$$

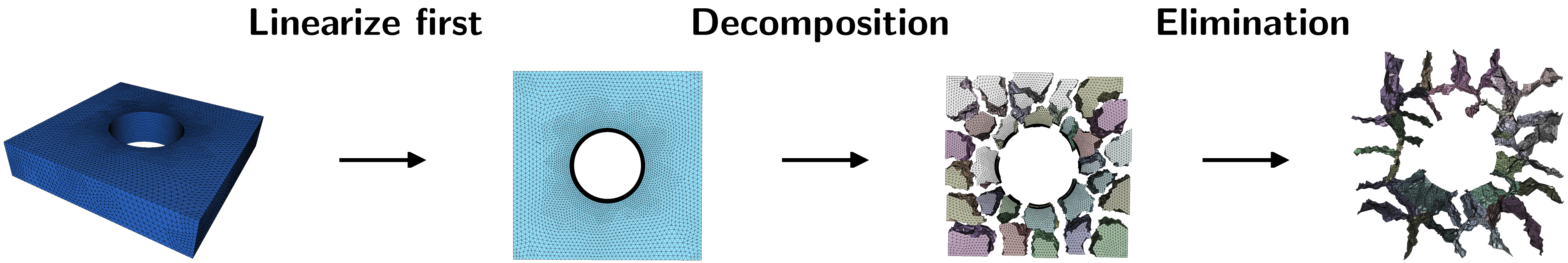
we linearize first with a Newton method:

$$u^{(k+1)} = u^{(k)} - \alpha^{(k)} \delta u^{(k)}$$

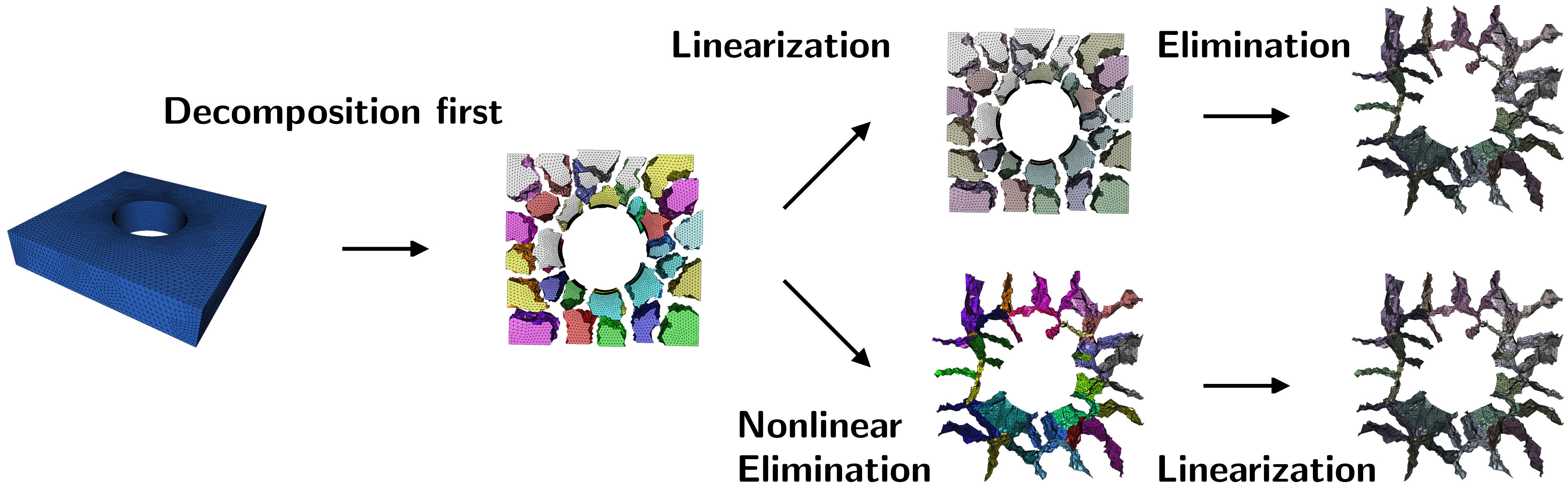
with a step length $\alpha^{(k)}$ and the update $\delta u^{(k)}$ given by:

$$DA(u^{(k)})\delta u^{(k)} = A(u^{(k)}). \qquad (2)$$

Newton-Krylov FETI-DP/Multigrid is then decomposing the computational domain and using hybrid FETI-DP/Multigrid in order to solve (2).



Nonlinear FETI-DP Methods



- Decomposition of the discretized nonlinear problem before linearization
- $\Rightarrow$ local nonlinear problems $\Rightarrow$ Increases local work
- Reduces the number of Newton steps, Krylov iterations and communication
- Combinable with hybrid FETI-DP/Multigrid methods

All nonlinear FETI-DP methods are based on the nonlinear FETI-DP saddlepoint system:

$$\begin{aligned} \widetilde{K}(\tilde{u}) + B^T\lambda - \tilde{f} &= 0 \\ B\tilde{u} &= 0 \end{aligned}$$

Summary and Implementation Remarks

- **Parallelization Strategy:** MPI based
- FE2TI is written in C/C++ using **PETSc, Umfpack, MUMPS, BoomerAMG, HDF5**
- Macroscopic problem is decomposed into many RVEs
- A subcommunicator is assigned to each RVE created by MPI_Comm_split
- Efficient solve of RVEs using a (Nonlinear/Newton-Krylov) FETI-DP/Multigrid method
- ⇒ **Embarrassingly parallel RVE solves**
- Efficient direct solver packages for local FETI-DP subdomain problems (Umfpack or MUMPS)
- Parallel AMG implementation **BoomerAMG for the global FETI-DP coarse problem**

Nonlinear Domain Decomposition

Nonlinear FETI-DP and Nonlinear BDDC: Klawonn, Lanser, Rheinbach (2012, 2013, 2014, 2015) **ASPIN:** Cai, Keyes 2002; Cai, Keyes, Marcinkowski 2002; Hwang, Cai 2005, 2007; Groß, Krause 2010,13; **MSPIN:** Keyes, Liu, 2015 **Nonlinear Neumann-Neumann:** Bordeu, Boucard, Gosselet 2009; **Nonlinear FETI-1:** Pebrel, Rey, Gosselet 2008; **Other DD work reversing linearization and decomposition:** Ganis, Juntunen, Pencheva, Wheeler, Yotov 2014; Ganis, Kumar, Pencheva, Wheeler, Yotov 2014

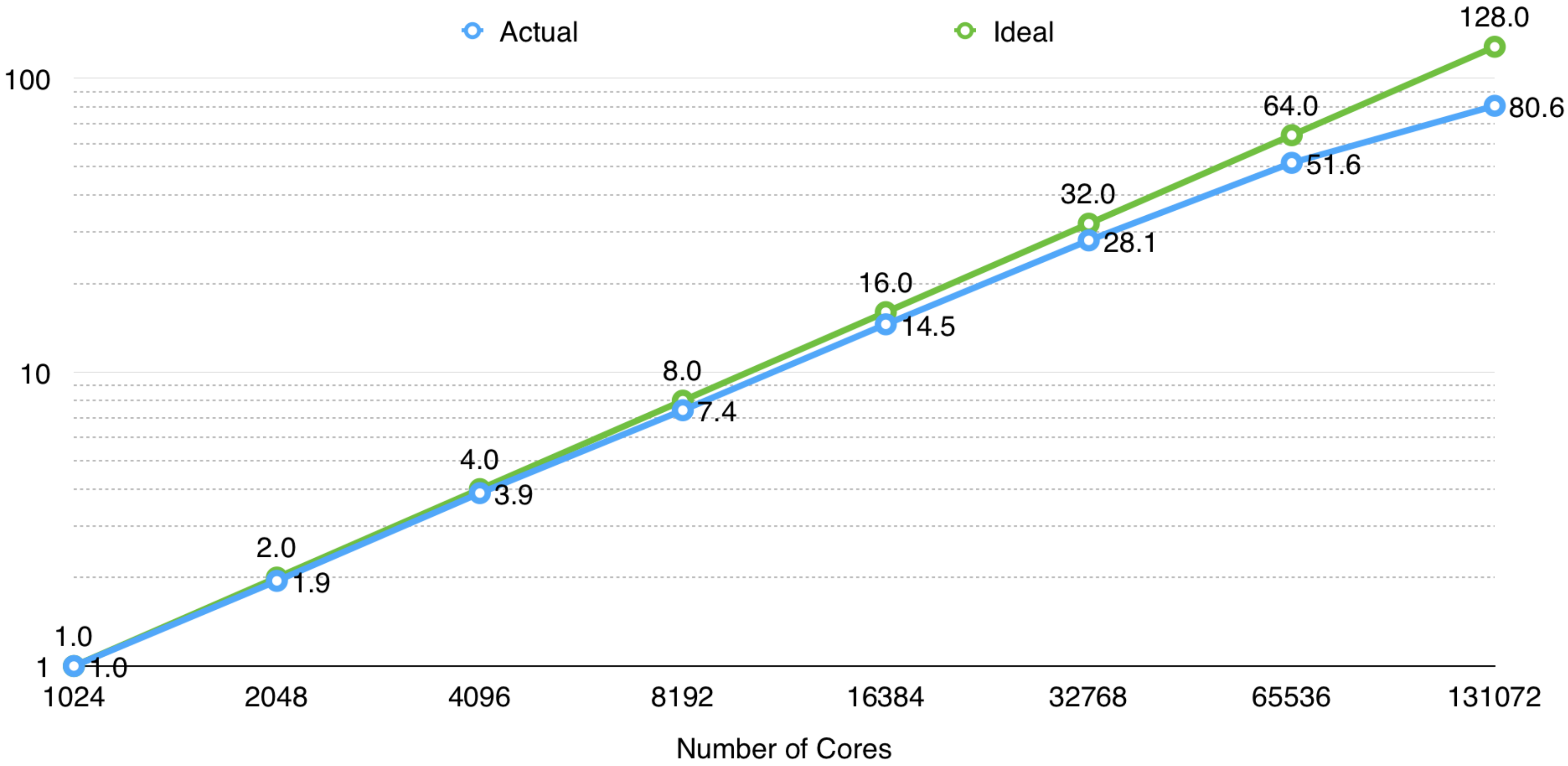
Hybrid Nonlinear FETI-DP/Multigrid - Strong Scaling

Cores	Subdomains	Problem Size	Execution Time	Actual Speedup	Ideal Speedup	Parallel Effic.
1 024	131 072	419 471 361	3 365.1s	1.0	1	100%
2 048	131 072	419 471 361	1 726.4s	1.9	2	97%
4 096	131 072	419 471 361	868.0s	3.9	4	97%
8 192	131 072	419 471 361	453.5s	7.4	8	93%
16 384	131 072	419 471 361	231.4s	14.6	16	91%
32 768	131 072	419 471 361	119.8s	28.1	32	88%
65 536	131 072	419 471 361	64.3s	51.6	64	81%
131 072	131 072	419 471 361	41.7s	80.6	128	63%

Software / machine: Vulcan BlueGene/Q at Lawrence Livermore National Laboratory; Using UMFPACK, PETSc 3.4.3 and BoomerAMG from hypre-2.9.4a package; Compiled with IBM compiler. **Problem:** 2D nonlinear hyperelasticity (Neo-Hooke); stiff circular inclusions in soft material; discretized with piecewise quadratic finite elements. **Solver:** Hybrid nonlinear FETI-DP/Multigrid

PETSc (Argonne National Laboratory): **Balay, Brown, Buschelman, Gropp, Kaushik, Knepley, Curfman McInnes, Smith and Zhang**
BoomerAMG (Lawrence Livermore National Laboratory): **Henson and Meier-Yang**

Hybrid Nonlinear FETI-DP/Multigrid - Strong Scaling

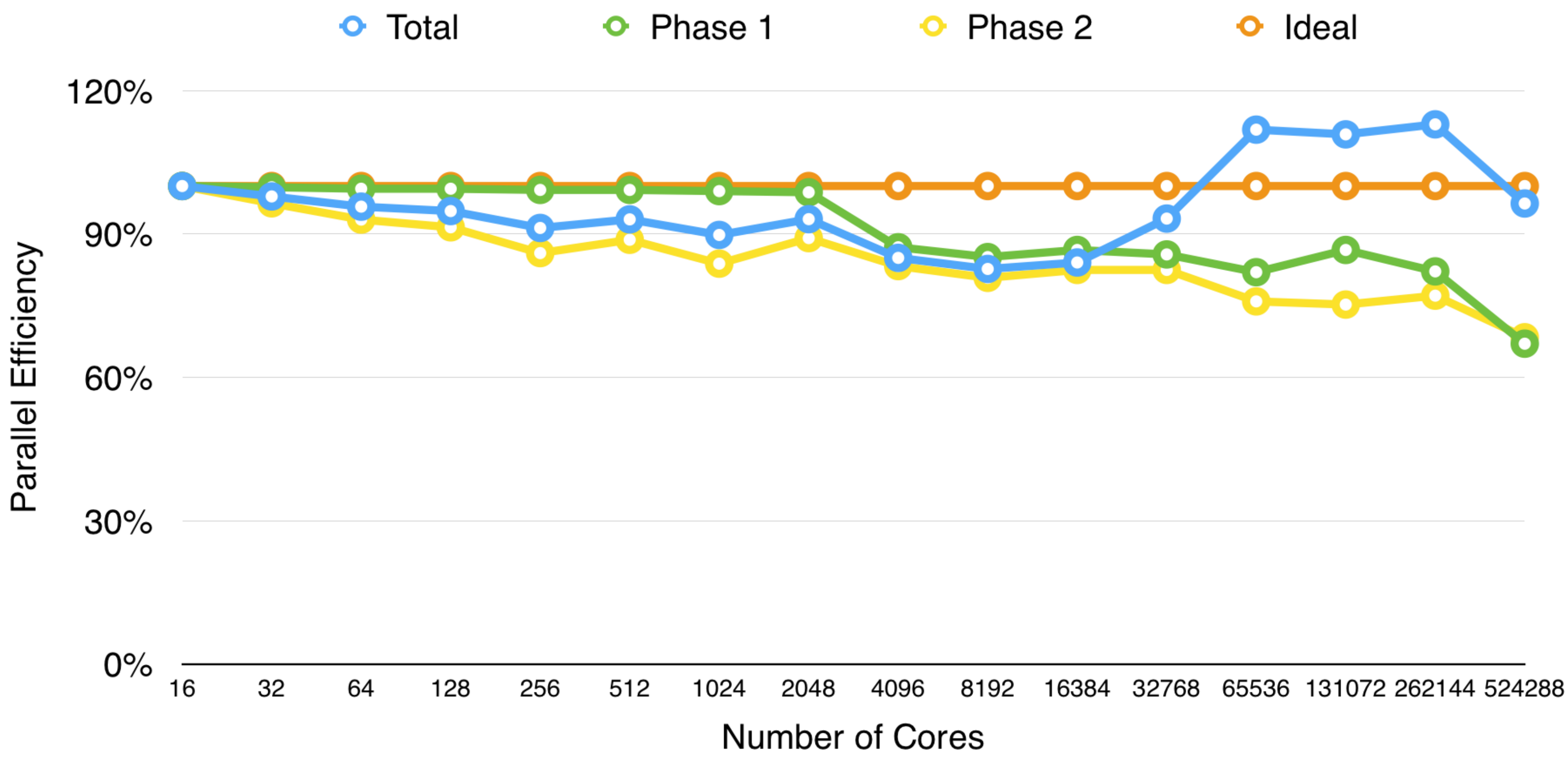


Hybrid Nonlinear FETI-DP/Multigrid - Weak Scaling

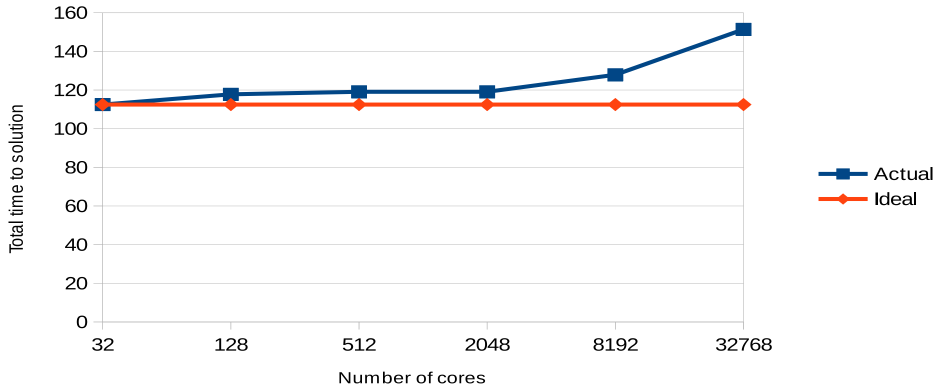
Cores	Problem Size	Phase 1 Time / Newton	Phase 2 Time / Newton	Krylov Iter	Total Time	Parallel Efficiency
16	1.3M	158.7s / 4	205.3s / 3	83	**364.0s**	100%
64	5.1M	159.5s /4	220.9s / 3	109	**380.4s**	96%
256	20M	160.1s / 4	238.9s / 3	135	**399.0s**	91%
1 024	82M	160.3s / 4	245.2s / 3	136	**405.5s**	90%
4 096	328M	182.0s / 4	246.5s / 3	110	**428.4s**	85%
8 192	655M	186.4s / 4	254.0s / 3	114	**440.4s**	83%
16 384	1 311M	137.3s / 4	249.0s / 3	110	**433.3s**	84%
32 768	2 622M	138.9s / 4	251.7s / 3	111	**390.6s**	93%
65 536	5 243M	145.3s / 4	180.3s / **2**	85	**325.6s**	112%
131 072	10 486M	147.5s / **3**	182.0s / **2**	84	**329.5s**	110%
262 144	20 972M	144.9s / **3**	177.5s / **2**	83	**322.4s**	113%
524 288	41 944M	177.6s / **3**	200.2s / **2**	82	**377.8s**	96%

Software / machine: Mira BlueGene/Q at Argonne National Laboratory; Using MUMPS, PETSc 3.5.2 and BoomerAMG from hypre-2.9.1a package; Compiled with IBM compiler. **Problem:** 2D nonlinear hyperelasticity (Neo-Hooke); stiff circular inclusions in soft material; discretized with piecewise quadratic finite elements. **Solver:** Hybrid nonlinear FETI-DP/Multigrid

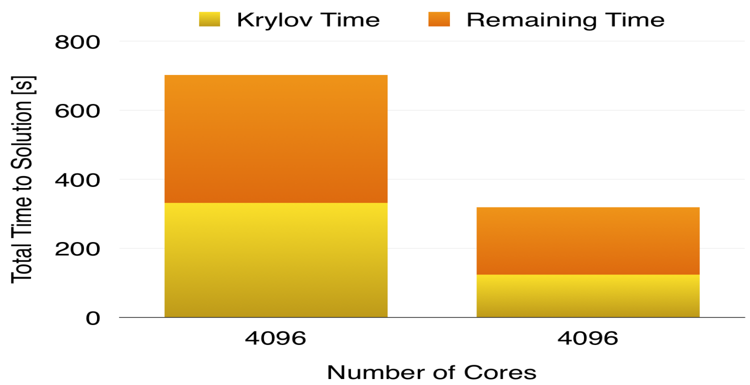
Hybrid Nonlinear FETI-DP/Multigrid - Weak Scaling



Hybrid Nonlinear FETI-DP/Multigrid - Weak Scaling



Hybrid nonlinear FETI-DP/Multigrid algorithm on the **SuperMUC** supercomputer at Leibniz-Rechenzentrum in Munich; $\Delta + 4\Delta_4$, Large subdomains $> 200\,000$ degrees of freedom.

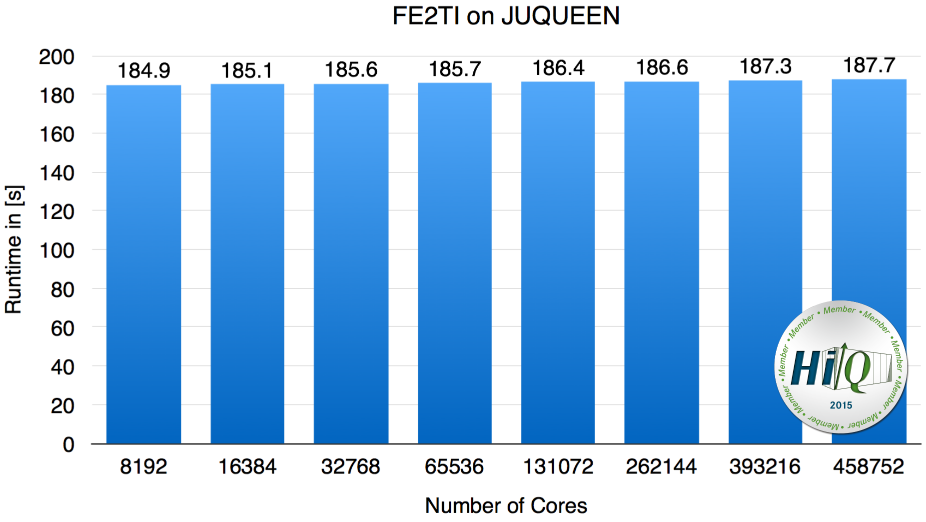


Comparison of Nonlinear FETI-DP (right) and Newton-Krylov FETI-DP (left). Performed on a **Cray System** at University of Duisburg-Essen.

Comparison to Results on BlueGene/Q

+ More memory per core and better performance of direct solvers (PARDISO from IntelMKL) $\Rightarrow$ Large FETI-DP subdomains possible $\Rightarrow$ Increased number of d.o.f. per core
+ Shorter time to solution (up to a factor of 2 depending on the problem)
- Less scalable, since communication between subdomains is more expensive.

Weak Scaling of FE2TI



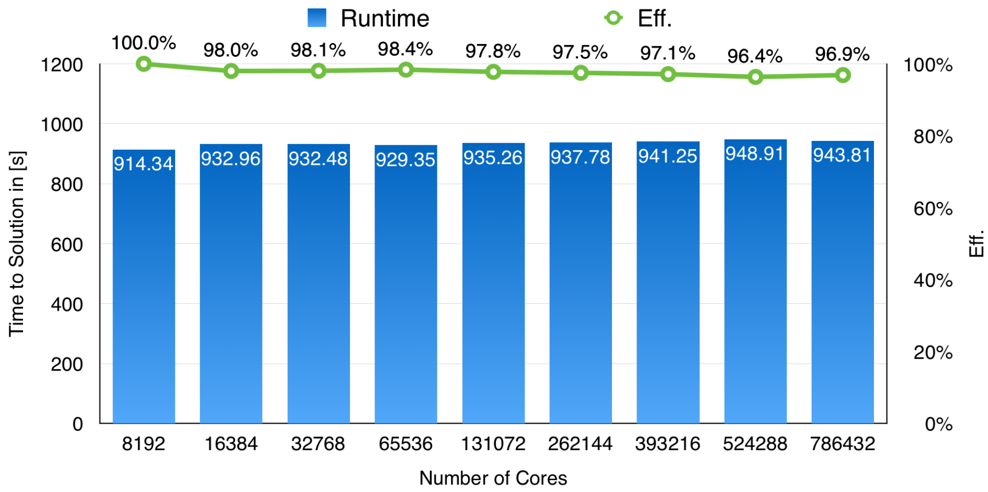
FE^{2}TI in 3D (Weak scaling; JUQUEEN)				
Cores	**MPI-ranks**	**#RVEs**	**Time**	**Par. Eff.**
8 192	8 192	16	184.86s	100.0%
16 384	16 384	32	185.09s	99.9%
32 768	32 768	64	185.61s	99.6%
65 536	65 536	128	185.72s	99.5%
131 072	131 072	256	186.43s	99.2%
262 144	262 144	512	186.61s	99.1%
393 216	393 216	768	187.32s	98.7%
458 752	458 752	896	187.65s	98.5%

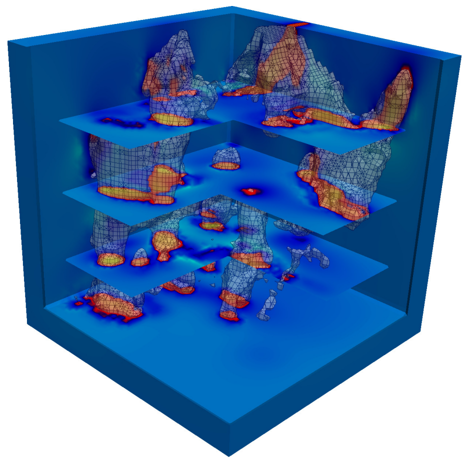
FE2TI in 3D using hybrid FETI-DP/Multigrid on each RVE; heterogeneous hyperelasticity; Q1 finite elements macro, P2 finite elements micro; 1.6m d.o.f. on each RVE; 512 subdomains for each RVE; 4 OpenMP threads per MPI-rank.

FE^{2}TI in 2D (Increasing RVE sizes; JUQUEEN)					
Cores	**MPI-ranks**	**#RVEs**	**RVE-size**	**RVE-size $\times$ #RVEs**	**Time to Solution**
458 752	458 752	1 792	5 126 402	9 186 512 384	161.78s
458 752	458 752	1 792	7 380 482	13 225 823 744	248.19s
458 752	458 752	1 792	13 117 442	23 506 456 064	483.68s
458 752	458 752	1 792	20 492 802	36 723 101 184	817.06s

Weak Scaling of FE2TI

FE2TI in 3D (Weak scaling)					
Cores	**MPI ranks**	**#RVEs**	**Total dof**	**Time to Solution**	**Eff.**
8 192	16 384	16	200M	914.34s	100.0%
16 384	32 768	32	401M	932.96s	98.0%
65 536	131 072	128	1.6B	929.35s	98.4%
131 072	262 144	256	3.2B	935.26s	97.8%
262 144	524 288	512	6.4B	937.78s	97.5%
524 288	1 048 576	1 024	12.8B	948.91s	96.4%
786 432	1 572 864	1 536	19.3B	943.81s	96.9%

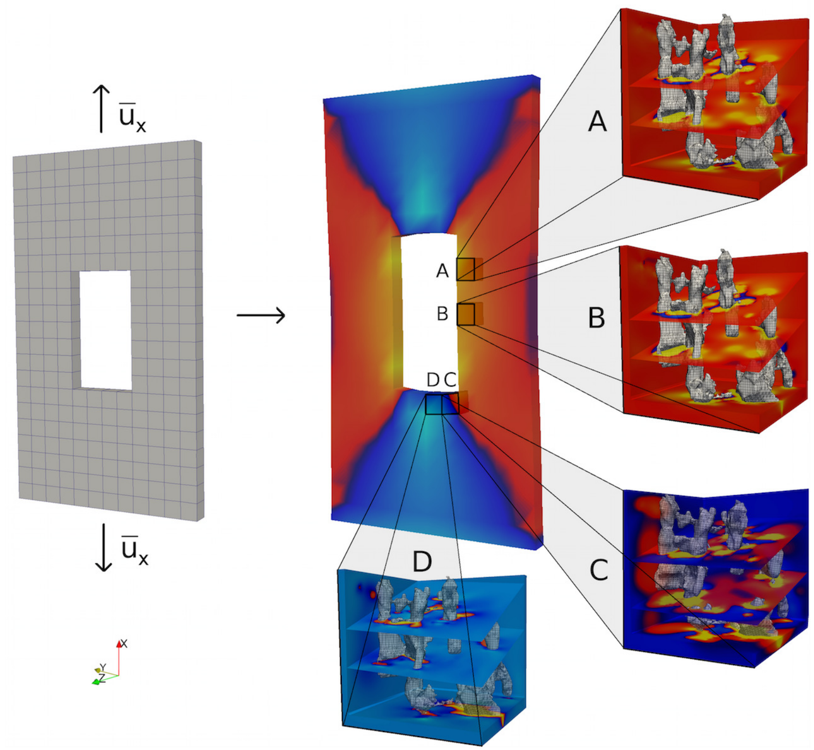




FE2TI in 3D using hybrid FETI-DP/Multigrid on each RVE; heterogeneous hyperelasticity; Q1 finite elements macro, P2 finite elements micro; 12.5 million d.o.f. on each RVE; 4 096 subdomains, 1 024 MPI ranks, and 512 cores for each RVE.

Production Runs of FE2TI on JUQUEEN

JUQUEEN - Complete FE^2 runs for elasticity						
Task	**RVE type**	**#Racks**	**#MPI-ranks**	**#RVEs**	**#Load Steps**	**Time**
Complete FE^2	Real RVE	1	32 768	64 RVEs	41 LS	16 899s
Complete FE^2	Real RVE	4	131 072	256 RVEs	41 LS	17 733s
Complete FE^2	Real RVE	28	917 504	1792 RVEs	40 LS	18 587s



FE2TI:
1 792 RVEs with real micro structure.
Uses all 28 racks of JUQUEEN.
Left: Undeformed rectangular plate with a hole; 224 Q1 (macro) finite elements with 8 Gauß points each.
Right: von Mises stresses of the deformed macroscopic problem and four exemplary RVEs in different Gauß points (A,B,C,D).
Stress peaks in microstructures are 5-7 times higher than peaks in macroscopic problem.
I/O times of approximately 2% of the runtime on 28 racks (Strategy: Writing data of all RVEs to one large, parallel HDF5 file)

Conclusion

- Scalability for FE2TI for up to 486K cores on the JUQUEEN BG/Q (FZ Jülich) and for up to 786K cores on the Mira BG/Q (ANL)
- Production runs of FE2TI on the complete JUQUEEN for a nontrivial elasticity problem
- Nonlinear FETI-DP methods show the potential to localize work and thus save communication
- Scalability for hybrid nonlinear FETI-DP/Multigrid for up to 131K cores on the Vulcan BG/Q (LLNL) and for up to 524K cores on the Mira BG/Q (ANL)

Acknowledgement

- The use of **JUQUEEN** at Jülich Supercomputing Centre (JSC) during the Workshop on "Extreme Scaling on JUQUEEN" is gratefully acknowledged.
- The authors acknowledge the Gauss Centre for Supercomputing (GCS) for providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS share of the supercomputer **JUQUEEN**.
- This research used resources (**Mira**) of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02- 06CH11357.
- The use of **Vulcan** at Lawrence Livermore National Laboratory is gratefully acknowledged.
- Support is gratefully acknowledged by Deutsche Forschungsgemeinschaft (DFG) within the priority program SPP 1648 Software for Exascale Computing

Thank you for your attention!