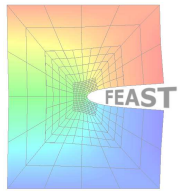
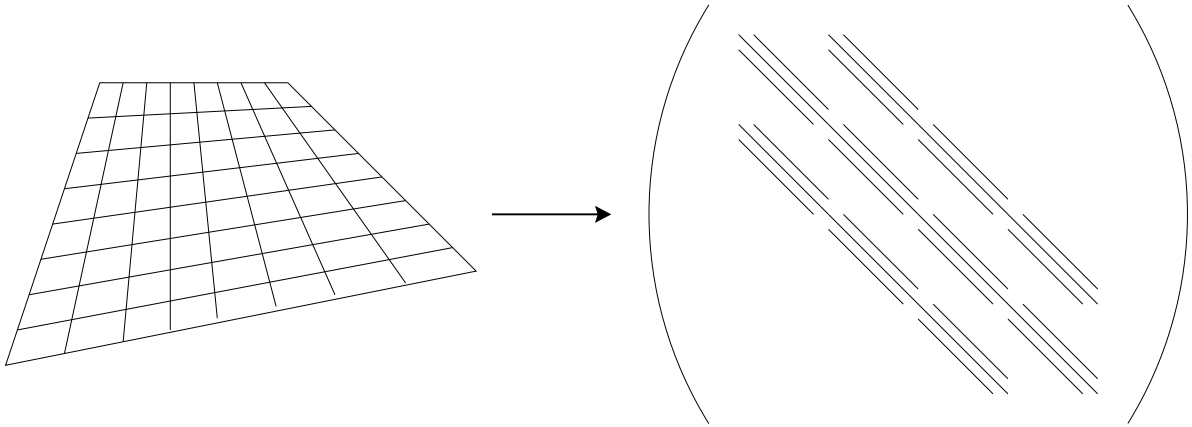
Overview (II)



FEAST's Mesh & Matrix Concept

'data moving $\gg$ data processing'

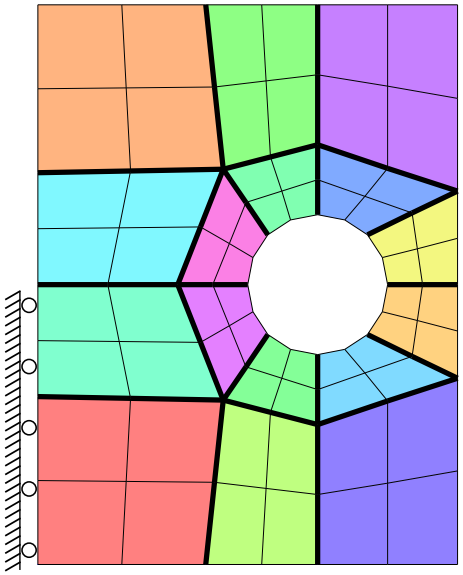
- unstructured meshes $\Rightarrow$ indirect adressing
  $\Rightarrow$ expensive memory access $\Rightarrow$ poor MFLOP/s rates
- FEAST uses generalised tensor product meshes

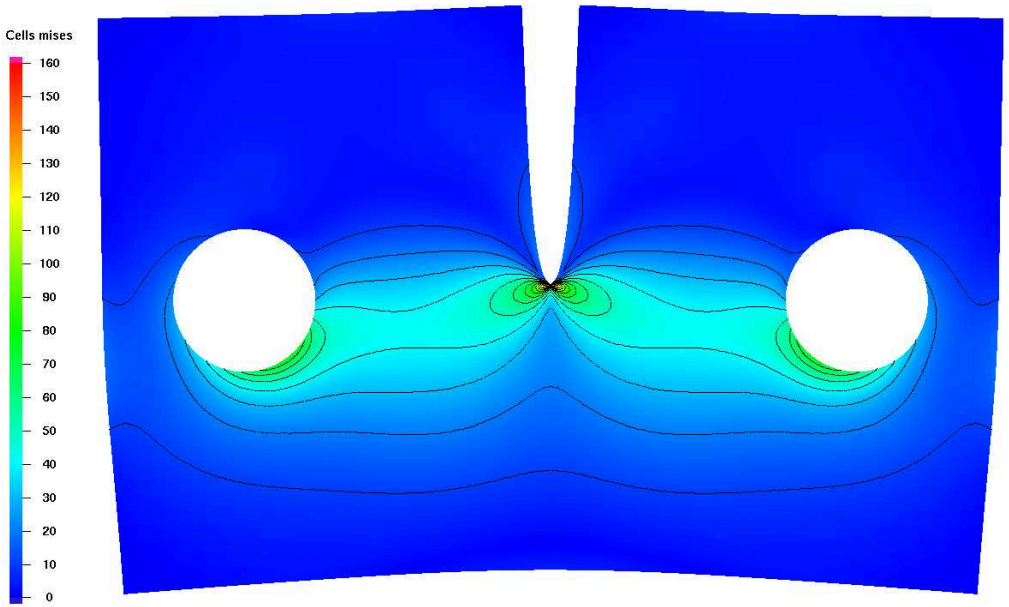


- rowwise numbering
  $\Rightarrow$ exactly 9 matrix bands for bilinear elements
- direct adressing, caching
- optimised Linear Algebra routines (SPARSE BANDED BLAS)

FEAST's Mesh & Matrix Concept

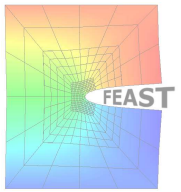
More complex domains by joining several TP meshes ('macros')





Here:

- 64 macros (=64 local matrices), each macro refined 10x
- distributed over 16 processors
- $\Rightarrow 1.34 \cdot 10^8$ DOFs in total

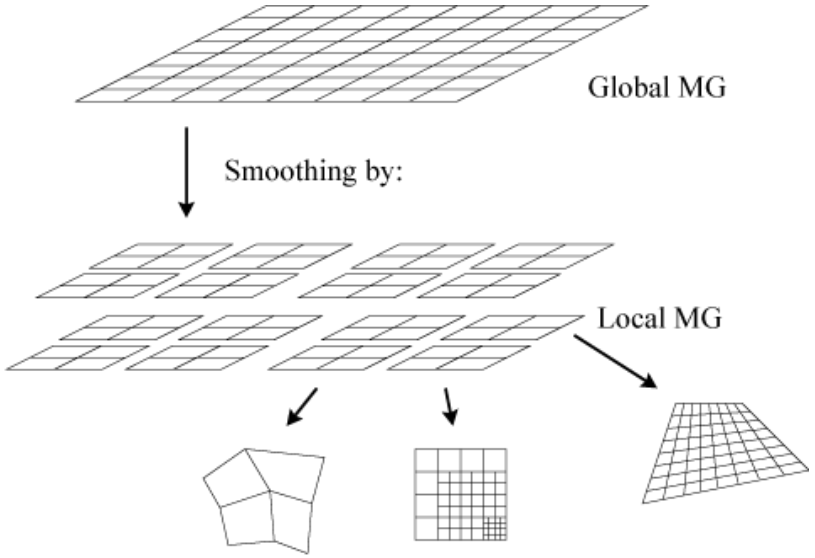


FEAST's Solver Concept

FEAST uses generalised Multigrid/Domain Decomposition methods

ScaRC
Scalable Recursive Clustering

- global MG smoothed by local MG
- locally adapted solution methods
- recursively hide local mesh irregularities



⇒ high numerical and parallel efficiency

Elasticity with FEAST

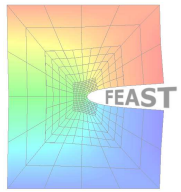
- Basic idea: Reduction to solution of scalar problems
- Example: Linear Elasticity for compressible material
- Separate displacement ordering $\Rightarrow$ block-structured linear system:

$$\begin{pmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{12}^T & \mathbf{K}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{pmatrix}$$

- $\mathbf{K}_{11}$ and $\mathbf{K}_{22}$ correspond to scalar elliptical operators
- Basic iteration: block-preconditioned Richardson method

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \tilde{\mathbf{K}}^{-1}(\mathbf{f} - \mathbf{K}\mathbf{u}^k),$$

e. g. Block-Jacobi $\quad \tilde{\mathbf{K}}^{-1} = \begin{pmatrix} \mathbf{K}_{11}^{-1} & \mathbf{0} \\ 0 & \mathbf{K}_{22}^{-1} \end{pmatrix}$



- Acceleration: Krylov-space methods, multigrid

Elasticity with FEAST

- Basic idea: Reduction to solution of scalar problems
- Example: Linear Elasticity for compressible material
- Separate displacement ordering $\Rightarrow$ block-structured linear system:

$$\begin{pmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{12}^T & \mathbf{K}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{pmatrix}$$

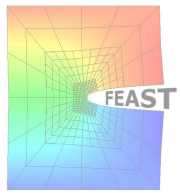
- $\mathbf{K}_{11}$ and $\mathbf{K}_{22}$ correspond to scalar elliptical operators
- Basic iteration: block-preconditioned Richardson method

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \tilde{\mathbf{K}}^{-1}(\mathbf{f} - \mathbf{K}\mathbf{u}^k),$$

e. g. Block-Jacobi $\quad \tilde{\mathbf{K}}^{-1} = \begin{pmatrix} \mathbf{K}_{11}^{-1} & \mathbf{0} \\ 0 & \mathbf{K}_{22}^{-1} \end{pmatrix}$

Exploit FEAST concepts!

- Acceleration: Krylov-space methods, multigrid



Saddle Point Solvers

- Mixed u/p formulation for (nearly) incompressible material

$$\begin{pmatrix} \mathbf{K} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}$$

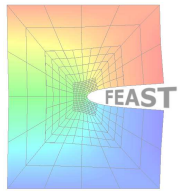
- Compressibility and LBB-stabilisation terms $\longrightarrow \mathbf{C}$
- Solving strategy: Pressure Schur Complement approach

$$\mathbf{S}\mathbf{p} = \mathbf{B}^T\mathbf{K}^{-1}\mathbf{f} - \mathbf{g}$$

  with $\mathbf{S} := \mathbf{B}^T\mathbf{K}^{-1}\mathbf{B} - \mathbf{C}$
- Basic iteration:

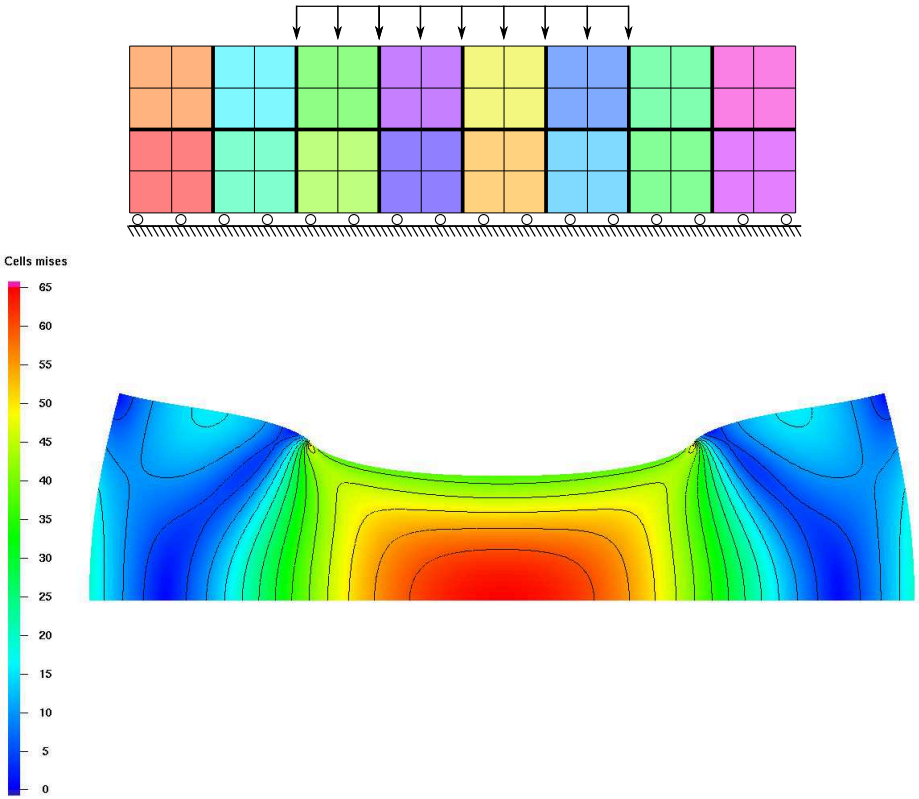
$$\mathbf{p}^{k+1} = \mathbf{p}^k + \tilde{\mathbf{S}}^{-1}(\mathbf{B}^T\mathbf{K}^{-1}\mathbf{f} - \mathbf{g} - \mathbf{S}\mathbf{p}^k)$$

- Applying SC preconditioner $\tilde{\mathbf{S}} \Rightarrow$ scalar equation!
- Acceleration: Krylov-space method



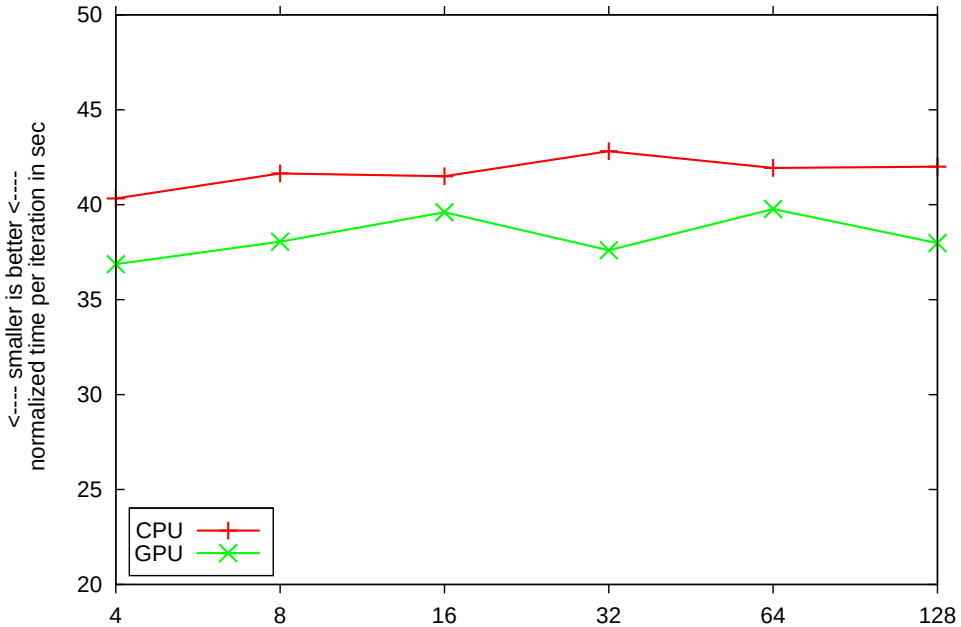
Results

- Compute cluster: 128 nodes, each one equipped with
  2 Intel XEON CPUs (3.4 GHz) and
  1 NVIDIA Quadro FX1400 GPU (350 MHz)
- Linear Elasticity, compressible material, outer BiCGstab solver,
  $\varepsilon_{\text{rel}} = 10^{-6}$



Results

#DOFs	CPU			GPU		
	#CPUs	#iters	time	#GPUs	#iters	time
3.36e+7	8	5.5	221.8	4	5.5	202.8
6.71e+7	16	6.5	270.7	8	6.5	247.3
1.34e+8	32	6	249.0	16	6	237.6
2.68e+8	64	6	250.9	32	6	225.6
5.37e+8	128	7	293.6	64	7	278.4
1.07e+9	256	6.5	273.0	128	6.5	246.9



Outlook (II)

Already possible (CSM):

- Finite Deformation (St. VK, Neo-Hooke)
- Compressible + incompressible
- Damped Newton solver, Jacobian via Finite Differences
- Transient computations

Outlook (FSI):

$$\frac{\partial \mathbf{F}(\mathbf{u},\mathbf{v},p)}{\partial(\mathbf{u},\mathbf{v},p)} = \begin{pmatrix} \mathbf{K_{uu}} & \mathbf{K_{uv}} & 0 \\ \mathbf{K_{vu}} & \mathbf{K_{vv}} & \mathbf{B_u} + \mathbf{B_v} \\ \mathbf{B_u}^T & \mathbf{B_v}^T & \mathbf{C} \end{pmatrix}$$

- Step 1: Applying fully-coupled Vanka-schemes (modified for Q1/Q1)
- Step 2: SC approach + reduction to scalar ScaRC solvers
- Step 3: Efficient Schur complement preconditioner ?!?