High Performance Computing Techniques for the FEM Simulation in Solid Mechanics

Hilmar Wobker Stefan Turek

Institute of Applied Mathematics,
Technische Universität Dortmund, 44227 Dortmund, Germany

email: hilmar.wobker@math.uni-dortmund.de

Mini-Workshop: Theory and Numerics of Fluid-Solid Interaction
Oberwolfach, November 26-30, 2007

Overview

Overview

1. Motivation
2. Concepts of FEAST
   - Meshing Concept
   - Adaptivity Concept
   - Solution Concept
3. Treating Elasticity Problems with FEAST
4. Numerical Results

Motivation

Efficiency of iterative solution methods is influenced by

- physical parameters
- algorithmic parameters
- mesh quality
- number of parallel processors

Distinguish between three aspects:

- processor efficiency
- numerical efficiency
- parallel efficiency

Motivation

Develop a solution method for problems in
Computational Solid Mechanics (CSM)
with

high efficiency in all three aspects
and
low dependency on the listed influences

Motivation

Develop a solution method for problems in
Computational Solid Mechanics (CSM)
with

high efficiency in all three aspects
and
low dependency on the listed influences

Realisation with

FEAST

Finite Element Analysis and Solution Tool

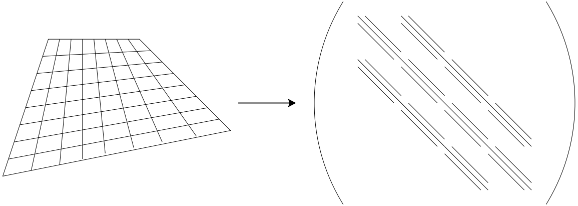
`http://www.feast.uni-dortmund.de`

Overview

FEAST's Meshing Concept

'data moving $\gg$ data processing'

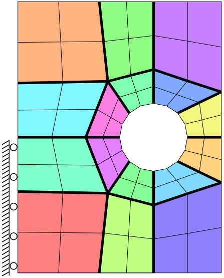
- unstructured meshes $\Rightarrow$ indirect adressing
  $\Rightarrow$ expensive memory access $\Rightarrow$ poor MFLOP/s rates
- FEAST uses generalised tensor product meshes

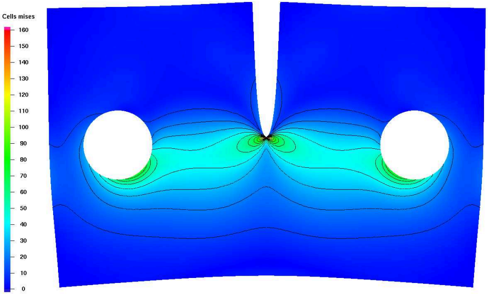


- rowwise numbering
  $\Rightarrow$ exactly 9 matrix bands for bilinear elements
- direct adressing, caching
- optimised Linear Algebra routines
  (SPARSE BANDED BLAS)

FEAST's Meshing Concept

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



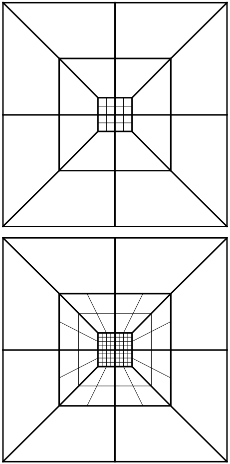


Here:

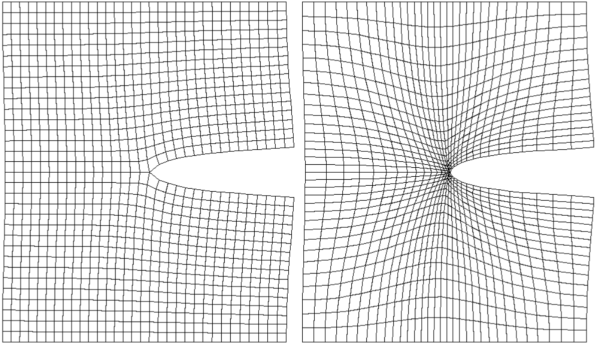
- 64 macros (=64 local matrices), each macro refined 10x
- distributed over 16 processors
- $\Rightarrow 1.34 \cdot 10^8$ DOFs in total

FEAST's Adaptivity Concept

Patch-wise Hanging Nodes:

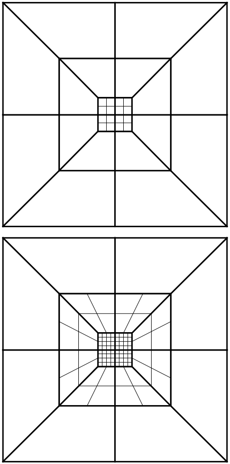


Mesh Deformation:

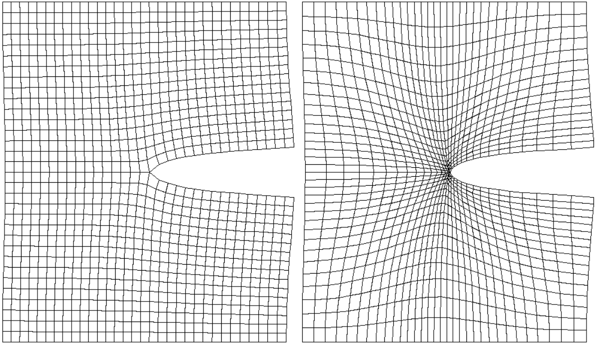


FEAST's Adaptivity Concept

Patch-wise Hanging Nodes:



Mesh Deformation:



TP property fulfilled!

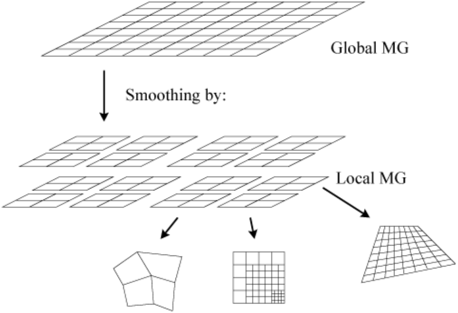
FEAST's Solution Concept

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
- minimal overlap

$\Rightarrow$ high numerical and parallel efficiency

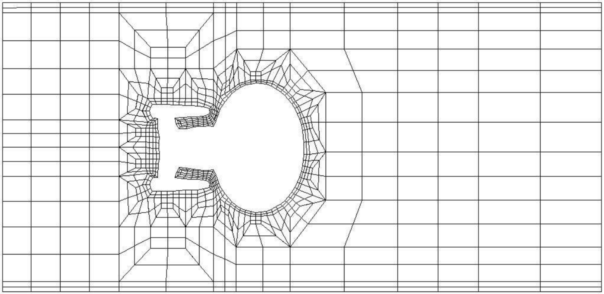


FEAST results (‘Poisson Problem’)

Local MFLOP/s rates:

#DOF	MV (Sparse)	MV (SparseBanded)	
		var	const
65^2	422	1111	1605
257^2	106	380	1214
1025^2	54	362	1140

Sun V40z ‘Opteron’(1800 MHz, peak perf. ≈ 2900 MFLOP/s)



C. Becker, 2006

FEAST results ('Poisson Problem')

Local MFLOP/s rates:

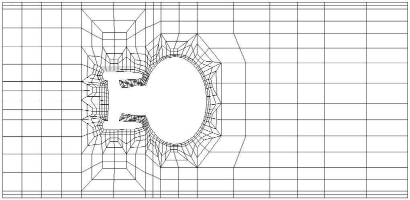
#DOF	MV (Sparse)	MV (SparseBanded) var	MV (SparseBanded) const
65^2	422	1111	1605
257^2	106	380	1214
1025^2	54	362	1140

Sun V40z 'Opteron'(1800 MHz, peak perf. ≈ 2900 MFLOP/s)

Global (parallel) convergence rates:

#DOF	$AR \approx 10$	$AR \approx 10^6$
211 K	0.17 (**8**)	0.18 (**8**)
844 K	0.17 (**8**)	0.17 (**8**)
3,375 K	0.18 (**9**)	0.19 (**9**)
13,500 K	0.19 (**9**)	0.18 (**9**)

ScaRC-CG solver,
smoothers: 1 glob. ScaRC, 1 loc. MG



C. Becker, 2006

FEAST

'HPC meets Modern Numerics'

$\Rightarrow$ fast and robust solvers for scalar, linear, elliptic equations

FEAST

'HPC meets Modern Numerics'

$\Rightarrow$ fast and robust solvers for scalar, linear, elliptic equations

Application to vector-valued, non-linear, time dependent equations in CSM ?

Overview

Example: 2D Linearised Elasticity

- Basic idea: Reduction to solution of scalar problems
- Lamé equation → weak formulation → FE discretisation
- Separate displacement ordering ⇒ 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 $\quad \tilde{\mathbf{K}}^{-1} = \begin{pmatrix} \mathbf{K}_{11}^{-1} & \mathbf{0} \\ 0 & \mathbf{K}_{22}^{-1} \end{pmatrix}$

Example: 2D Linearised Elasticity

- Basic idea: Reduction to solution of scalar problems
- Lamé equation $\longrightarrow$ weak formulation $\longrightarrow$ FE discretisation
- 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!

Example: 2D Linearised Elasticity

- Basic idea: Reduction to solution of scalar problems
- Lamé equation $\rightarrow$ weak formulation $\rightarrow$ FE discretisation
- Separate displacement ordering $\Rightarrow$ block-structured linear system:

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

- $\mathbf{K}_{11}$ 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, MG

Nearly Incompressible Material

$$\nu \to 0.5 \quad \Rightarrow \quad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \to \infty$$

Difficulties:

- Increasing degree of anisotropy in the operator

$$-(2\mu + \lambda)\,\partial_{xx} - \mu\,\partial_{yy}$$

- Deterioration of the FE approximation ('volume locking')

Remedy:

- Introduce new variable ('pressure'): $p = -\lambda \operatorname{div} u$

$$\begin{aligned} -2\mu \operatorname{div} \varepsilon(u) + \nabla p &= f, \qquad x \in \Omega \\ -\operatorname{div} u - \frac{1}{\lambda} p &= 0, \qquad x \in \Omega \end{aligned}$$

- $Q_1/Q_1 \Rightarrow$ stabilisation necessary

Solving the Saddle Point Problem

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

- compressibility and stabilisation terms $\longrightarrow \mathbf{C}$
- Schur complement $\mathbf{S} := \mathbf{B}^\mathsf{T}\mathbf{K}^{-1}\mathbf{B} - \mathbf{C}$

Solving the Saddle Point Problem

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

- compressibility and stabilisation terms $\rightarrow$ $\mathbf{C}$
- Schur complement $\mathbf{S} := \mathbf{B}^T\mathbf{K}^{-1}\mathbf{B} - \mathbf{C}$

First solving strategy: Block-Preconditioning approach

- Krylov-space method with block-preconditioner

$$\begin{pmatrix} \tilde{\mathbf{K}} & 0 \\ \mathbf{B}^T & -\tilde{\mathbf{S}} \end{pmatrix}$$

- preconditioners $\tilde{\mathbf{K}}$ and $\tilde{\mathbf{S}}$

Solving the Saddle Point Problem

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

- compressibility and stabilisation terms $\rightarrow \mathbf{C}$
- Schur complement $\mathbf{S} := \mathbf{B}^\mathsf{T}\mathbf{K}^{-1}\mathbf{B} - \mathbf{C}$

Second solving strategy: Pressure Schur Complement approach

- 'Cancelling' displacements $\mathbf{u} = \mathbf{K}^{-1}(\mathbf{f} - \mathbf{B}\mathbf{p})$

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

- Basic iteration:

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

- Acceleration: Krylov-space method

Preconditioning the Schur complement

Essential for both approaches:
efficient Schur complement preconditioner $\tilde{\mathbf{S}}^{-1}$

Preconditioning the Schur complement

Essential for both approaches:
efficient Schur complement preconditioner $\tilde{\mathbf{S}}^{-1}$

Stationary case:

$$\mathbf{S}^{-1} = (\mathbf{B}^T\mathbf{K}^{-1}\mathbf{B} - \mathbf{C})^{-1}$$

- stabilisation terms are of magnitude $\mathcal{O}(h^2)$
  $\Rightarrow$ can be omitted in preconditioner

$$\tilde{\mathbf{S}}^{-1} := \begin{cases} \left(\frac{1}{2\mu}\mathbf{M}_p\right)^{-1} & \text{if } \nu = 0.5 \\ \left(\left(\frac{1}{2\mu} + \frac{1}{\lambda}\right)\mathbf{M}_p\right)^{-1} & \text{if } \nu < 0.5 \end{cases}$$

Preconditioning the Schur complement

Transient case:

$$\begin{pmatrix} \frac{\rho}{\tau^2\beta}\mathbf{M}_u + \mathbf{K} & \mathbf{B} \\ \mathbf{B}^\mathsf{T} & \mathbf{C} \end{pmatrix}$$

$$\Rightarrow \mathbf{S}^{-1} = \left(\mathbf{B}^\mathsf{T}\left(\frac{\rho}{\tau^2\beta}\mathbf{M}_u + \mathbf{K}\right)^{-1}\mathbf{B} - \mathbf{C}\right)^{-1}$$

- Goal: efficiency for all relevant time step sizes τ

Preconditioning the Schur complement

Transient case:

$$\begin{pmatrix} \frac{\rho}{\tau^2\beta}\mathbf{M}_u + \mathbf{K} & \mathbf{B} \\ \mathbf{B}^\mathsf{T} & \mathbf{C} \end{pmatrix}$$

$$\Rightarrow \mathbf{S}^{-1} = \left(\mathbf{B}^\mathsf{T}\left(\tfrac{\rho}{\tau^2\beta}\mathbf{M}_u + \mathbf{K}\right)^{-1}\mathbf{B} - \mathbf{C}\right)^{-1}$$

- Goal: efficiency for all relevant time step sizes τ
- Idea: distinctly precondition

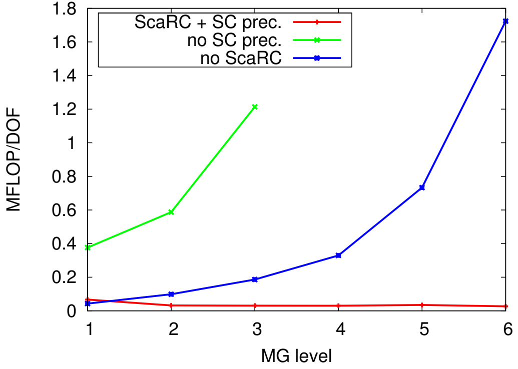
$(\mathbf{B}^\mathsf{T}\mathbf{K}^{-1}\mathbf{B})^{-1}$	with	$\mathbf{M}_p^{-1}$
$(\mathbf{B}^\mathsf{T}\mathbf{M}_u^{-1}\mathbf{B})^{-1}$	with	$\mathbf{P}_p^{-1}$

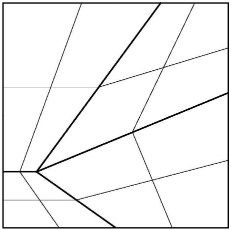
$$\boxed{\tilde{\mathbf{S}}^{-1} := \left(\tfrac{\tau^2\beta}{\rho}\mathbf{P}_p\right)^{-1} + \left(\tfrac{1}{2\mu}\mathbf{M}_p\right)^{-1}}$$

Overview

Results: Preconditioning

Influence of ScaRC and Schur complement preconditioning

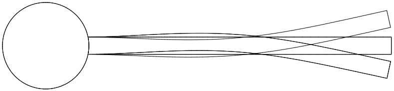




($\nu = 0.5$, block-preconditioned BiCGstab, $\varepsilon_{\text{rel}} = 1.0\text{e-}8$)

Results: Long Beam under Gravity

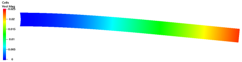
CSM part of Fluid-Solid-Interaction-Benchmark: Beam attached to cylinder in channel (DFG FOR 493)



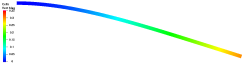
- Comparing three different lengths:

0.08 × 0.02 m

0.32 × 0.02 m

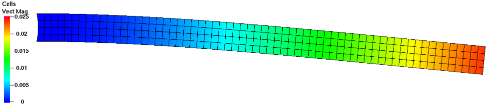


1.28 × 0.02 m

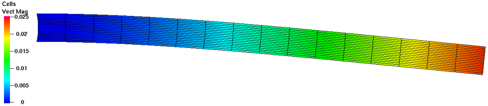


- Comparing two meshings:

isotropic elements:



anisotropic elements:



Results: Long Beam under Gravity

isotropic elements		0.08 x 0.02 m 4 macros		0.32 x 0.02 m 16 macros		1.28 x 0.02 m 64 macros	
Solver	#elem	it	sec	it	sec	it	sec
BiCG BGS	65 K	13	3.6	28	9.1	132	68.6
	262 K	14	14.9	28	32.5	113	167.3
	1049 K	14	62.3	31	142.7	114	608.8

anisotropic elements		0.08 x 0.02 m 1 macro		0.32 x 0.02 m 1 macro		1.28 x 0.02 m 1 macro	
Solver	#elem	it	sec	it	sec	it	sec
BiCG BGS	65 K	14	3.7	33	8.5	149	38.7
	262 K	16	18.9	35	38.4	154	169.1
	1049 K	14	71.9	34	156.3	210	1078.9

- Outer Krylov-subspace scheme with Block-GS preconditioning not sufficient (strong dependence on global anisotropy)

Results: Long Beam under Gravity

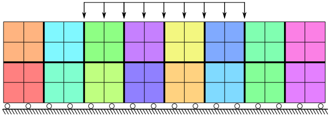
isotropic elements		0.08 × 0.02 m 4 macros		0.32 × 0.02 m 16 macros		1.28 × 0.02 m 64 macros	
Solver	#elem	it	sec	it	sec	it	sec
BiCG BGS	65 K	13	3.6	28	9.1	132	68.6
	262 K	14	14.9	28	32.5	113	167.3
	1049 K	14	62.3	31	142.7	114	608.8
BiCG MG BGS	65 K	4	4.7	5	7.3	6	19.2
	262 K	4	18.5	5	24.2	6	51.8
	1049 K	4	73.7	5	88.9	5	126.5

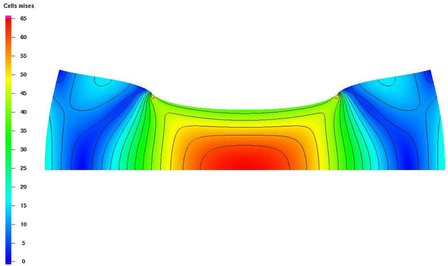
anisotropic elements		0.08 × 0.02 m 1 macro		0.32 × 0.02 m 1 macro		1.28 × 0.02 m 1 macro	
Solver	#elem	it	sec	it	sec	it	sec
BiCG BGS	65 K	14	3.7	33	8.5	149	38.7
	262 K	16	18.9	35	38.4	154	169.1
	1049 K	14	71.9	34	156.3	210	1078.9
BiCG MG BGS	65 K	4	4.2	5	5.2	9	9.4
	262 K	5	19.5	5	21.8	8	34.8
	1049 K	5	83.4	6	102.9	7	129.1

- Outer Krylov-subspace scheme with Block-GS preconditioning not sufficient (strong dependence on global anisotropy)
- Remedy: Use multigrid (applied to the whole system!) as preconditioner

Results: Parallel efficiency

- 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, $\varepsilon_{\text{rel}} = 10^{-6}$

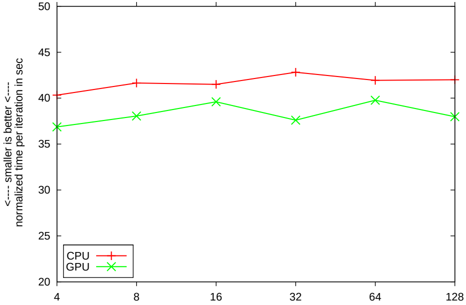




with D. Goeddeke, 2007

Results: Parallel efficiency

	CPU			GPU		
#DOFs	#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



with D. Goeddeke, 2007

Outlook

Current state (CSM):

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

Outlook (FSI):

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

- 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 ?!?

Thank you
for your attention!