

# Parallel Multilevel Solvers and their Application to Elasticity Problems (Teil der Vorlesung 'Wissenschaftliches Rechnen')

Hilmar Wobker

Institute of Applied Mathematics and Numerics, TU Dortmund, Germany  
email: [hilmar.wobker@math.tu-dortmund.de](mailto:hilmar.wobker@math.tu-dortmund.de)

SS 2010



## Parallel Multilevel Solvers for...

- 1 ...Scalar Elliptic Equations  
(16.6.2010)
- 2 ...Compressible Elasticity Problems  
(22.6.2010)
- 3 ...Incompressible Elasticity Problems in Saddle Point Form  
(23.6.2010)



## Part I

# Parallel Multilevel Solvers for Scalar Elliptic Equations



- 1 Introduction/Motivation
- 2 Parallel Multigrid Methods
- 3 Domain Decomposition Methods
- 4 FEAST and ScaRC



- 1 Introduction/Motivation
- 2 Parallel Multigrid Methods
- 3 Domain Decomposition Methods
- 4 FEAST and ScaRC



Efficiency of iterative linear solvers influenced by

- material parameters,
- nonlinear effects,
- the shape of the geometry,
- the size and the quality of the computational mesh,
- algorithmic parameters, and
- the number of processors in a parallel computing system.



Three aspects of efficiency:

- **numerical efficiency**: amount of work to achieve a desired goal; convergence rate and robustness
- **processor efficiency**: ability to exploit the full capacity of modern hardware
- **parallel efficiency**: communication vs. computation, scalability

**Hardware-oriented numerics:**  
achieve good efficiency in all three aspects

→ difficult to realise due to conflicting demands

→ our attempt: **FEAST**



- improvement of hardware characteristics per year (1990–2004):
  - processor peak performance: 60 %
  - network technology (latency, bandwidth): 30 %
- physical constraint: speed of light
- 2004: 1 inter-processor communication  $\approx$  4000 FLOPs  
2020: 1 inter-processor communication  $\approx$  670 000 FLOPs
- number of processors in high-end super computers:  
2004:  $\approx$  4000  
2010:  $>$  100 000
- similar to the 'memory wall' problem
- $\Rightarrow$  parallel efficiency determined by amount of data exchange

data locality required for  
parallel efficiency



On the other hand:

- elliptic problems: solution in a point is influenced by all boundary values
- information has to 'travel' at least once through the whole grid
- fast global data exchange necessary for good iterative convergence rates

global information required for  
numerical efficiency

- conflicting demands: data locality vs. fast global data exchange
- parallel and numerical efficiency hard to combine



Requirements of parallel solution methods:

- low inter-processor communication
- high processor loads
- good scalability
- good convergence behaviour
- robustness with respect to complicated geometries, mesh refinement level, mesh irregularities, and number/size of subdomains

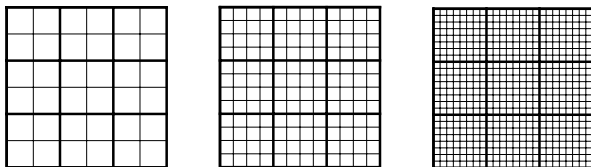
Suitable solver concepts for elliptic problems:

- multigrid methods (MG)
- domain decomposition methods (DD)



- 1 Introduction/Motivation
- 2 Parallel Multigrid Methods**
- 3 Domain Decomposition Methods
- 4 FEAST and ScaRC



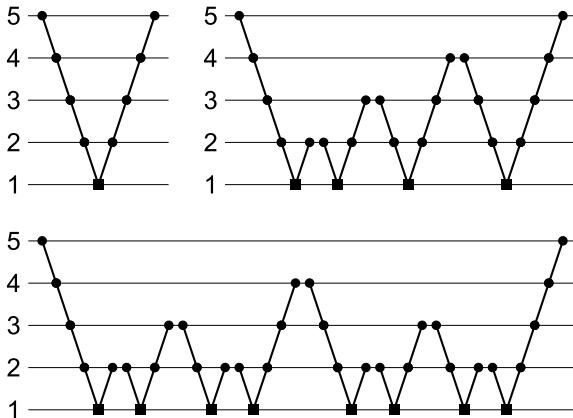


- use a hierarchy of nested grids to solve  $\mathbf{Ax} = \mathbf{b}$
- basic components:
  - smoothing (level  $l$ ):  $\mathbf{x}^{(l)} \leftarrow \mathbf{x}^{(l)} + \omega \mathbf{S}^{(l)}(\mathbf{b}^{(l)} - \mathbf{A}^{(l)}\mathbf{x}^{(l)})$
  - grid transfer: restriction ( $l \rightarrow l - 1$ ), prolongation ( $l - 1 \rightarrow l$ )
  - coarse grid solving:  $\mathbf{x}^{(1)} = (\mathbf{A}^{(1)})^{-1}\mathbf{b}^{(1)}$
- exploiting smoothing property of elementary methods like Jacobi or Gauß-Seidel
- convergence behaviour independent of the refinement level
- runtime  $O(\#\text{DOF})$

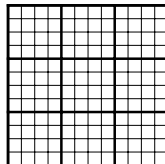
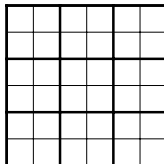


Possibilities to traverse the grid hierarchy:

- V-, F- and W-cycle
- arithmetic costs vs. convergence speed / robustness

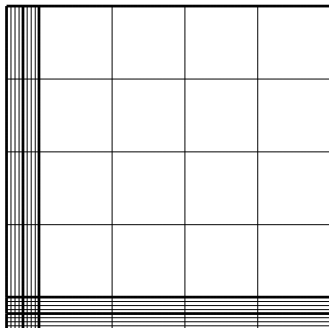
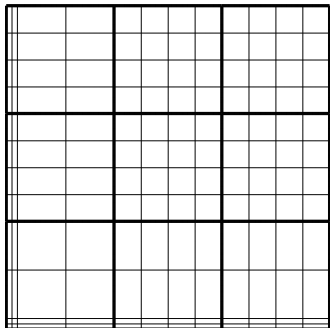


7	8	9
4	5	6
1	2	3



- smoothing operation highly recursive (Gauß-Seidel, ILU)
- bad parallelisation potential
- remedy: relax the smoothing operation by applying it *block-wise* (additively, block-Jacobi)
- use *minimal overlap*
- number of vertices  $n$ , number of subdomains  $M$ :
  - $M \rightarrow 1$ : standard multigrid with selected smoother
  - $M \rightarrow n$ : standard multigrid with Jacobi smoother





- 'mesh anisotropy' means: large aspect ratios (longest / shortest side)
- micro-anisotropies: isotropic subdomains (=macro elements), anisotropic (micro-)elements (left figure)
- macro-anisotropies (right): anisotropic subdomains and (automatically) anisotropic elements (right figure)



## Advantages:

- convergence rates independent of refinement level
- robust w.r.t. micro anisotropies (with suitable smoother)

## Disadvantages:

- block-Jacobi character  $\rightarrow$  sensitive w.r.t. macro anisotropies  
(good numerical efficiency of serial MG mainly due to recursiveness of the smoother)
- bad ratio of computation/communication on coarser levels



- 1 Introduction/Motivation
- 2 Parallel Multigrid Methods
- 3 Domain Decomposition Methods**
- 4 FEAST and ScaRC

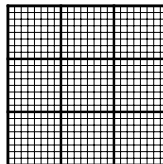


- classical 'divide & conquer' strategy:  
replace the solution of one large system (global layer) by the solution of several small systems (local layer)
- interpret DD as preconditioner  $\tilde{\mathbf{A}}$ :

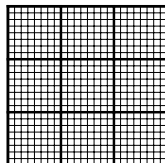
$$\mathbf{x} \leftarrow \mathbf{x} + \omega \tilde{\mathbf{A}}(\mathbf{b} - \mathbf{Ax})$$

- global coupling typically via Krylov space method
- two classes:
  - non-overlapping methods (substructuring / Schur complement),  
extra interface problem
  - overlapping Schwarz methods

7	8	9
4	5	6
1	2	3



- *additive* (treat subdomains simultaneously) vs. *multiplicative* (treat subdomains after each other)
- rule of thumb: multiplicative DD roughly halves the number of iterations
- local problems typically treated with direct or some specialised solver
- *one-level Schwarz method*: only *one* (fine) grid (right figure)  
→ strong dependence on number of subdomains, size of the overlap
- can be alleviated by adding a coarse grid problem (left figure)  
→ *two-level Schwarz method*



## Advantages:

- additive Schwarz  $\rightarrow$  perfect for parallel execution
- additional coarse grid increases numerical robustness
- robust w.r.t. to micro anisotropies (local direct solver)

## Disadvantages:

- one-level Schwarz: strong dependence on number of subdomains and size of the overlap; significant overlap necessary to achieve satisfactory convergence rates (10-20 % of subdomain size)
- additional coarse grid decreases parallel efficiency
- sensitive w.r.t. macro anisotropies
- local problems may be too large for direct solvers
- overlap complicated to implement (unstructured grids, 3D)



- 1 Introduction/Motivation
- 2 Parallel Multigrid Methods
- 3 Domain Decomposition Methods
- 4 FEAST and ScaRC**



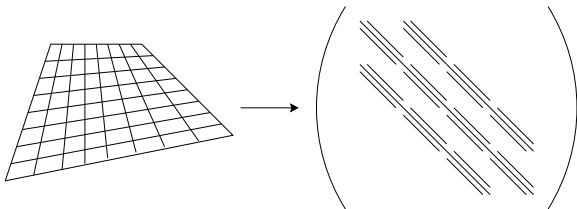
Memory wall problem:  
'data moving  $\gg$  data processing'

- improvement of hardware characteristics per year (1990–2004):
  - processor peak performance: 60 %
  - memory bandwidth: 30 %
  - memory latency: 5.5 %
- 1992: 1 memory access  $\approx$  1 FLOP  
2004: 1 memory access  $\approx$  100 FLOPs
- 'memory gap' is still broadening
- unstructured meshes  $\Rightarrow$  indirect addressing  
 $\Rightarrow$  expensive memory access
- plus: low arithmetic intensity (sparse matrices)  
 $\Rightarrow$  poor processor efficiency (low MFLOP/s rates)



Remedy: use structured data with  
high spatial and temporal locality

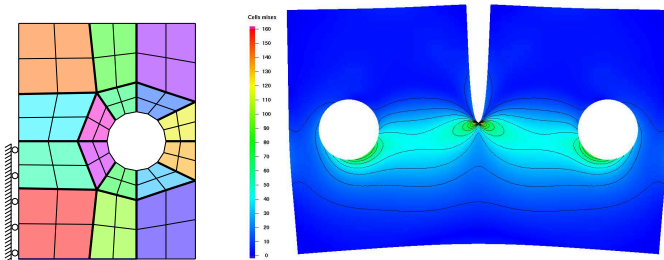
- FEAST uses **generalised tensor product meshes**



- rowwise numbering  
⇒ exactly **9 matrix bands** for bilinear elements
- direct addressing, caching
- optimised Linear Algebra routines (SparseBandedBLAS)



- more complex (unstructured) domains by joining several (structured) TP meshes
- local matrices + appropriate border data exchanges  
⇒ 'virtual' global matrix

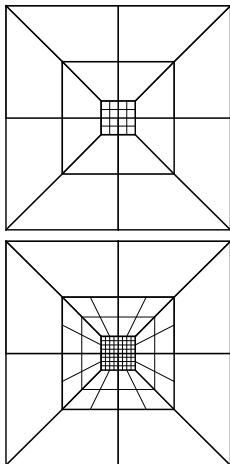


Here:

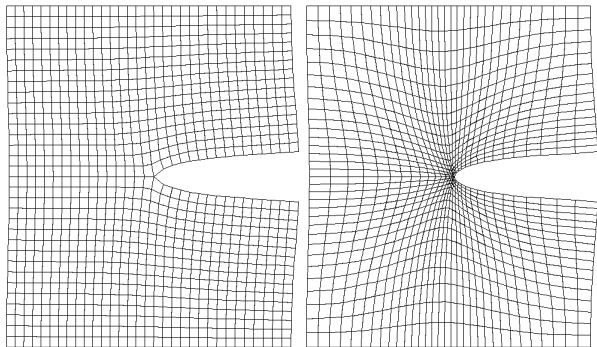
- 64 TP meshes (=64 local matrices), each refined 10x
- distributed over 16 processors
- ⇒  $1.34 \cdot 10^8$  degrees of freedom in total



Patch-wise  
Hanging Nodes:



Mesh Deformation:



TP property fulfilled!



- generalisation of two-level Schwarz: *multilevel Schwarz methods (MLDD)*
- MLDD methods can be
  - purely additive (additive within one level / additive between levels)
  - purely multiplicative (multiplicative/multiplicative)
  - hybrid (additive/multiplicative or multiplicative/additive)
- FEAST uses a generalised MG/DD solving concept called **ScaRC**
- ScaRC is a hybrid MLDD method that is
  - additive within one level
  - multiplicative between levels
- only minimal overlap

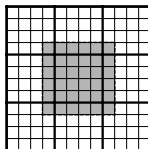
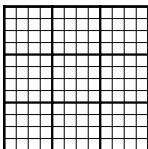


Switching between **global layer** and **local layer**:

- global set of vertices on level  $l$ :  $\mathcal{V}^{(l)}$
- set of vertices of  $i$ -th subdomain:  $\mathcal{V}_i^{(l)}$
- local index of vertex  $k$  in subdomain  $i$ :  $\text{loc}_i^{(l)}(k)$
- prolongation matrix  $\mathbf{P}_i^{(l)}$ :

$$(\mathbf{x}^{(l)})_k = (\mathbf{P}_i^{(l)} \mathbf{x}_i^{(l)})_k := \begin{cases} (\mathbf{x}_i^{(l)})_{\text{loc}_i^{(l)}(k)} & k \in \mathcal{V}_i^{(l)} \\ 0 & k \in \mathcal{V}^{(l)} \setminus \mathcal{V}_i^{(l)} \end{cases}$$

- restriction matrix:  $\mathbf{R}_i^{(l)} := (\mathbf{P}_i^{(l)})^T$
- local matrix:  $\mathbf{A}_i^{(l)} := \mathbf{R}_i^{(l)} \mathbf{A}^{(l)} \mathbf{P}_i^{(l)}$
- interpretation: 'ghost cells', **extended Dirichlet boundaries**



- sufficient for robust convergence behaviour of multilevel DD  
→ numerical efficiency
- local submeshes preserve tensor product property  
→ processor efficiency
- minimal amount of data exchange between subdomains  
→ parallel efficiency
- implementation and data structures greatly simplified



- one iteration (multiplicative part between levels):

$$\mathbf{x}^{(l)} \leftarrow \mathbf{x}^{(l)} + \tilde{\mathbf{A}}^{(l)}(\mathbf{b}^{(l)} - \mathbf{A}^{(l)}\mathbf{x}^{(l)})$$

$$l = L - 1, \dots, 2 :$$

$$\mathbf{b}^{(l)} \leftarrow \mathbf{R}^{(l)}(\mathbf{b}^{(l+1)} - \mathbf{A}^{(l+1)}\mathbf{x}^{(l+1)}), \quad \mathbf{x}^{(l)} \leftarrow \tilde{\mathbf{A}}^{(l)}\mathbf{b}^{(l)}$$

$$l = 1 :$$

$$\mathbf{b}^{(1)} \leftarrow \mathbf{R}^{(2)}(\mathbf{b}^{(2)} - \mathbf{A}^{(2)}\mathbf{x}^{(2)}), \quad \mathbf{x}^{(1)} \leftarrow (\mathbf{A}^{(1)})^{-1}\mathbf{b}^{(1)}$$

$$l = 2, \dots, L :$$

$$\mathbf{x}^{(l)} \leftarrow \mathbf{x}^{(l)} + \mathbf{P}^{(l)}\mathbf{x}^{(l-1)}, \quad \mathbf{x}^{(l)} \leftarrow \mathbf{x}^{(l)} + \tilde{\mathbf{A}}^{(l)}(\mathbf{b}^{(l)} - \mathbf{A}^{(l)}\mathbf{x}^{(l)})$$

- connection between global and local layer (additive part within one level):

$$\tilde{\mathbf{A}}^{(l)} := \widetilde{\sum_{i=1}^M \mathbf{P}_i^{(l)} \tilde{\mathbf{A}}_i^{(l)} \mathbf{R}_i^{(l)}}.$$

- local preconditioner  $\tilde{\mathbf{A}}_i^{(l)}$  for the local matrix  $\mathbf{A}_i^{(l)}$



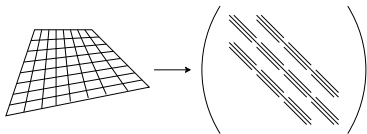
- two strategies to realise the local preconditioner  $\tilde{\mathbf{A}}_i^{(l)}$ :
  - ① apply one step of an elementary iterative scheme (Jacobi, Gauß-Seidel, ...)
  - ② apply some iterative or direct solution method to (approximately) solve the local system
- first strategy: MLDD coincides with block-smoothed MG ( $\tilde{\mathbf{A}}_i^{(l)}$  being the local smoother)
- $\Rightarrow$  MLDD generalisation of MG



First strategy:

- Notation of the local 9-band matrix (omitting superscript  $l$ ):

$$\mathbf{A}_i = (\mathbf{L}_i^L + \mathbf{L}_i^C + \mathbf{L}_i^U) + (\mathbf{C}_i^L + \mathbf{C}_i^C + \mathbf{C}_i^U) + (\mathbf{U}_i^L + \mathbf{U}_i^C + \mathbf{U}_i^U)$$



- some local smoothers:

$$\tilde{\mathbf{A}}_i^{\text{Jacobi}} := (\mathbf{C}_i^C)^{-1}$$

$$\tilde{\mathbf{A}}_i^{\text{GS}} := (\mathbf{L}_i^L + \mathbf{L}_i^C + \mathbf{L}_i^U + \mathbf{C}_i^L + \mathbf{C}_i^C)^{-1}$$

$$\tilde{\mathbf{A}}_i^{\text{TriGS}} := (\mathbf{L}_i^L + \mathbf{L}_i^C + \mathbf{L}_i^U + \mathbf{C}_i^L + \mathbf{C}_i^C + \mathbf{C}_i^U)^{-1}$$

$\tilde{\mathbf{A}}_i^{\text{MTriGS}}$  :  $\tilde{\mathbf{A}}_i^{\text{TriGS}}$  applied to column-wise numbered grid

$\tilde{\mathbf{A}}_i^{\text{ADITriGS}}$  : alternating application of  $\tilde{\mathbf{A}}_i^{\text{TriGS}}$  and  $\tilde{\mathbf{A}}_i^{\text{MTriGS}}$



Second strategy:

- solve local systems  $\mathbf{A}_i \mathbf{x}_i = \mathbf{b}_i$  (approximately) via
  - *direct solver* if the system is not too large ( $\#\text{DOF} < 20\,000$ )
  - *multigrid method*, otherwise
- local MG uses the same smoothers as block-smoothed (global) MG (see previous slide)
- ‘automatically toggle’ between MG and direct solver via *truncated multigrid method* (don’t traverse the complete mesh hierarchy)
- typical local problem size:  $\#\text{DOF} \approx 10^6$   
⇒ local multigrid mandatory

Using **local multigrid within global MLDD** is one of the core ideas of FEAST’s solution method!



Comparison of the two strategies (black dot = favoured strategy):

storage requirements	1	○ ● ○ ○	2
flexibility	1	○ ○ ● ○	2
'black box' character	1	○ ● ○ ○	2
arithmetic costs	1	● ○ ○ ○	2
communication vs. computation	1	○ ○ ○ ●	2
influence on global iteration	1	○ ○ ○ ●	2
load balancing	1	● ○ ○ ○	2
using co-processors	1	○ ○ ○ ●	2

Basic idea of FEAST's solver concept **ScaRC**  
(**Scalable** Recursive Clustering):

- allow both strategies at the same time
- choose local solver components adaptively according to the 'local situation'



- first strategy: **1-layer-ScaRC**  
(multigrid scheme only on global layer)
- second strategy: **2-layer-ScaRC**  
(multigrid schemes on global *and* local layer)
- short notation
  - 1-layer-ScaRC:  
**MG\_\_JAC\_\_D**  
**MG(1e-6,V44,0.7)\_\_JAC\_\_D**
  - 2-layer-ScaRC:  
**MG\_\_MG-ADI-D\_\_D**  
**MG(1e-6,F22,0.7)\_\_MG(1e-1,V22)-ADI-D\_\_D**
  - ‘\_\_’: layer change  
‘D’: direct coarse grid solver  
‘1e-6’: relative stopping criterion (‘gain 6 digits’)  
‘V22’: cycle type and pre-/postsmoothing steps  
‘0.7’: damping parameter

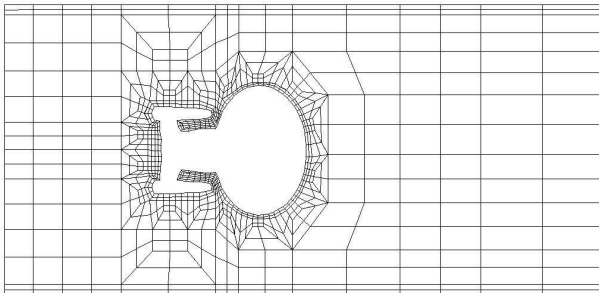


# Results (Poisson Problem)

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.  $\approx$  2900 MFLOP/s)



C. Becker, 2006

Local MFLOP/s rates:

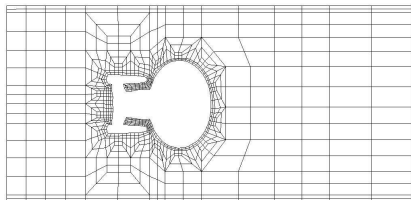
#DOF	MV (Sparse)	MV (SparseBanded)	
		var	const
$65^2$	422	1111	1605
$257^2$	106	380	1214
$1025^2$	54	362	1140

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

Global (parallel) convergence rates:  
(AR = max. element aspect ratio)

#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)

2-layer-ScaRC: CG-MG\_\_MG-ADI-D\_\_D



C. Becker, 2006

## Advantages of ScaRC:

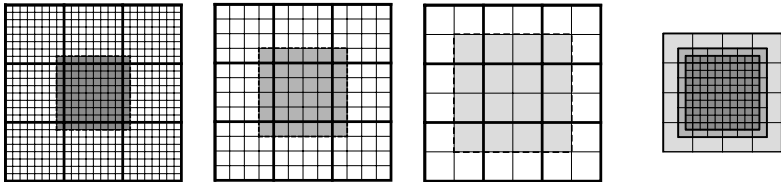
- convergence rates independent of refinement level
- convergence rates independent of number of subdomains (in case of only mild macro anisotropies)
- 'simple' local solvers/smoothers on 'simple' subdomains, 'strong' ones on 'difficult' subdomains
- good balance of computational costs and convergence behaviour
- hide local irregularities from the global solver ('gain one digit' in local solves)
- minimise number of global iterations and amount of communication
- high processor loads due to local tensor product grids



## Disadvantages of ScaRC:

- convergence rates dependent on number of subdomains in case of stronger macro anisotropies (block-Jacobi character)
  - can be alleviated by enhancing the global multilevel solver with Krylov space methods
  - further idea: *Recursive Clustering (RC)* (→ last slides)
- bad ratio computation/communication on coarser grid levels (inherent to multilevel approaches!)
- difficult to realise: 'automatic adaptation system' for local solver components
- even more difficult: dynamic load balancing
- local multigrid solvers are nonconforming (→ next slides)





- minimal overlap  $\Rightarrow$  extended Dirichlet boundaries
- local domain size increases with coarser grid levels
- nonnested local grids  $\Rightarrow$  nonconforming local multigrid method
- coarse grid correction not optimal
- (massive) convergence problems of standard multigrid schemes
- loss of level independency



Remedy:

- adaptively damp the suboptimal coarse grid correction:

$$\mathbf{x}^{(l)} \leftarrow \mathbf{x}^{(l)} + \alpha \mathbf{P}^{(l)} \mathbf{x}^{(l-1)}$$

(subscripts for local subdomains omitted)

- minimise error in energy norm:

$$(\mathbf{x}^{(l)} - \mathbf{x}^*)^T \mathbf{A}^{(l)} (\mathbf{x}^{(l)} - \mathbf{x}^*)$$

( $\mathbf{x}^*$  exact solution)

- for symmetric  $\mathbf{A}^{(l)}$ :

$$\alpha = \frac{\mathbf{c}^{(l)T} \mathbf{d}^{(l)}}{\mathbf{c}^{(l)T} \mathbf{A}^{(l)} \mathbf{d}^{(l)}}$$

( $\mathbf{c}^{(l)} := \mathbf{P}^{(l)} \mathbf{x}^{(l-1)}$  prolonged coarse grid correction,  
 $\mathbf{d}^{(l)} := \mathbf{b}^{(l)} - \mathbf{A}^{(l)} \mathbf{x}^{(l)}$  current defect)



## Advantages:

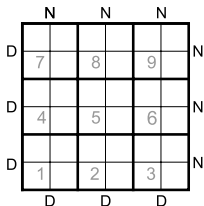
- established technique
- easy to implement
- relatively low computational costs (one matrix vector multiplication, two scalar products)
- damping parameter computed per subdomain:
  - $\alpha \ll 1$  for subdomains that suffer strongly from ext. Dirichl. bound.
  - $\alpha \approx 1$  otherwise

⇒ overall solution process not unnecessarily impaired in general

## Heuristic used in FEAST before ACGC:

- prolonged values corresponding to subdomain boundary nodes were not fully added (divided by number of incident subdomains)
- equally affects all local solves
- fails in some cases (see following slides)

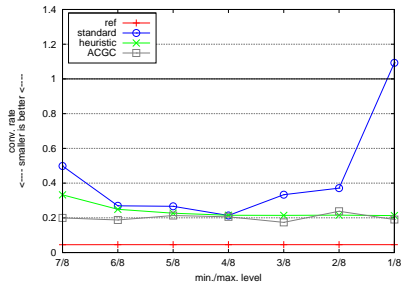
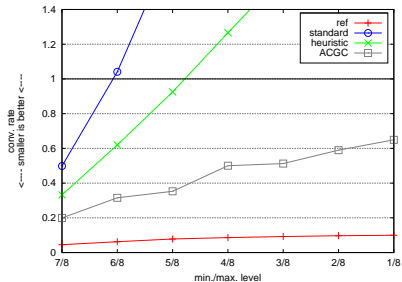




- (outer) geometric boundary conditions: Neumann ('N'), Dirichlet ('D')
- (inner) extended Dirichlet boundaries ('E')
- only consider local solve on top right subdomain (center figure)
- reference computation with standard Dirichlet instead of extended Dirichlet boundary conditions (right figure)
- solve standard Poisson problem  $-\Delta u = 1$
- local solver  $MG(1e-6, [V|F]22, 0.7)$ -JAC-D

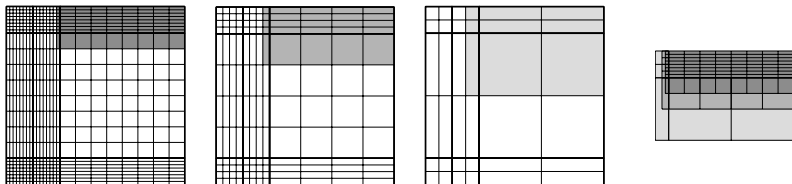


## isotropic subdomain, Jacobi smoother



V-cycle (left) and F-cycle (right)

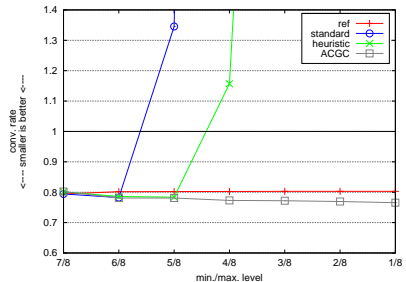
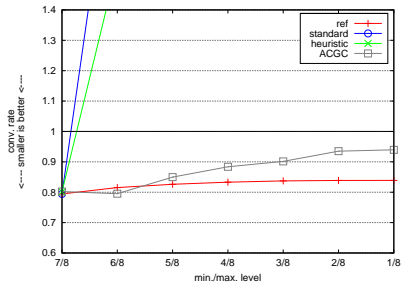




- anisotropic example
- consider top right subdomain again
- local solver  $MG(1e-6, [V|F] 22, 0.6)$  - JAC-D



## anisotropic subdomain, Jacobi smoother



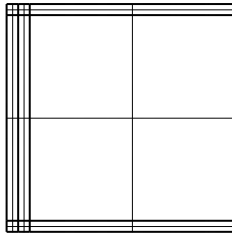
V-cycle (left) and F-cycle (right)

Summary for W-cycle:

- W-cycle with heuristic: converges in most, but not in all cases  
⇒ not reliable
- standard W-cycle: always converges, but strange oscillations depending on number of grid levels
- W-cycle + ACGC: always converges, no oscillations



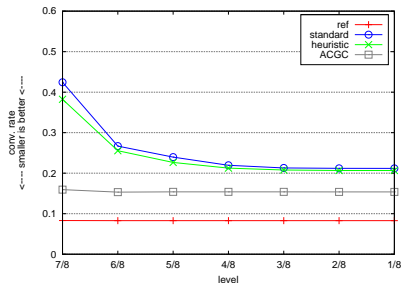
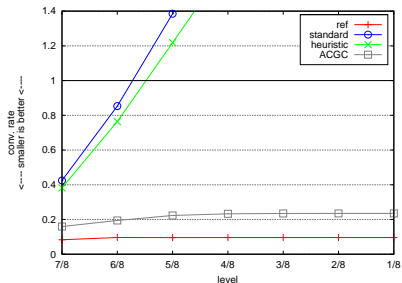
Deficiencies maybe only due to (weak) Jacobi smoother? No!



- now: use **ADITriGS** instead of Jacobi as local smoother
- increase anisotropies
- consider top right subdomain again
- local solver MG(1e-6, [V|F] 22, 1.0) -ADI-D
- solve standard Poisson problem  $-\Delta u = 1$



## anisotropic subdomain, ADITriGS smoother

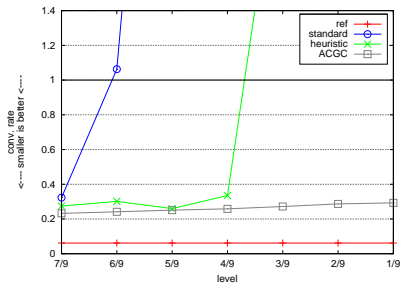
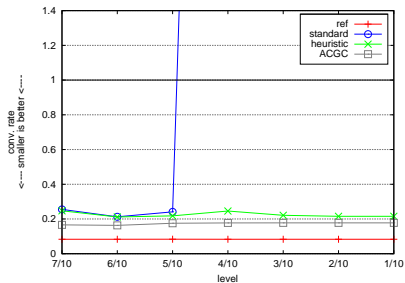


V-cycle (left) and F-cycle (right)

- hence, standard/heuristic F-cycle reliable?
- no! (next slide)



anisotropic subdomain, ADITriGS smoother, F-cycle


 isotr. operator  $-\Delta u$  (left), **anisotr. operator**  $-10\partial_{xx} u - \partial_{yy} u$  (right)

Summary for W-cycle:

- standard W-cycle and W-cycle with heuristic: always converge, but strange oscillations depending on number of grid levels
- W-cycle + ACGC: always converges, no oscillations



Now: consider **arithmetic costs of the global solver scheme**

- use **total arithmetic efficiency**:

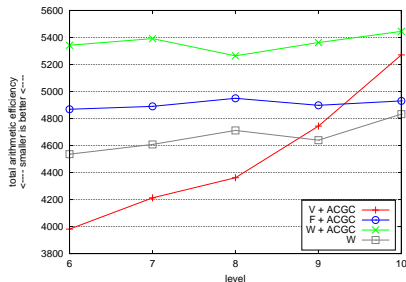
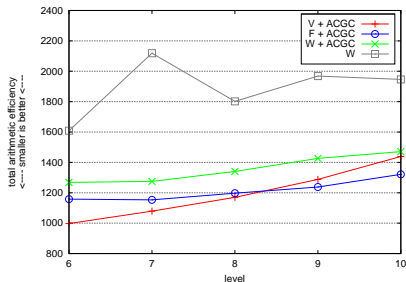
$$\text{TAE} = - \frac{\# \text{FLOPs}}{\# \text{DOF} \times \# \text{iter} \times \log_{10}(c)}$$

⇒ How many FLOPs are needed per DOF to gain one digit?

- 2-layer-ScaRC solver with Jacobi smoother:  
MG(1e-6, V11) \_\_ MG(1e-1, V22, 0.7) - JAC - D \_\_ D
- 2-layer-ScaRC solver with ADITriGS smoother:  
MG(1e-6, F22) \_\_ MG(1e-1, V22) - ADI - D \_\_ D
- only vary maximum MG level, fix coarse grid level to 1



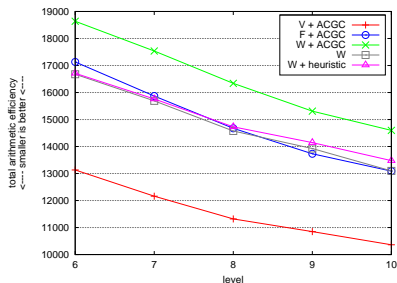
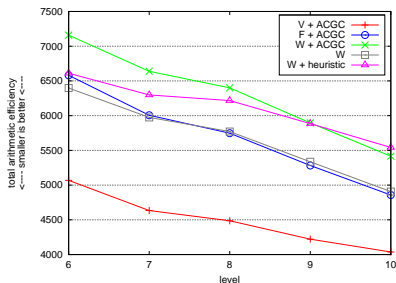
## 2-layer-ScaRC with local Jacobi smoother, isotropic operator



isotropic configuration (left), anisotropic configuration (right)



## 2-layer-ScaRC with local ADITriGS smoother, anisotropic configuration



isotropic operator (left), anisotropic operator (right)



Adaptive coarse grid correction:

- method to alleviate the negative effects of the extended Dirichlet boundary conditions ('increasing subdomain size')
- easy to implement
- low arithmetic costs
- facilitates the use of V- and F-cycle multigrid (important for GPU-computing!)
- smoothes oscillatory convergence behaviour of W-cycle
- often less expensive than standard W-cycle
- and: can also improve standard (conforming) multigrid methods

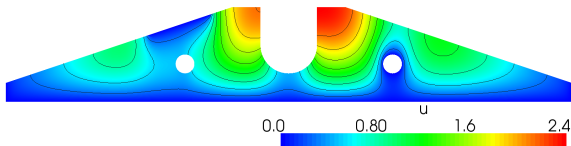
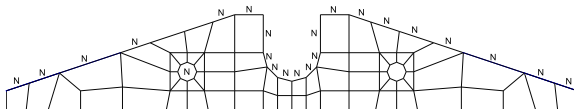


Is 2-layer-ScaRC really superior to 1-layer-ScaRC?

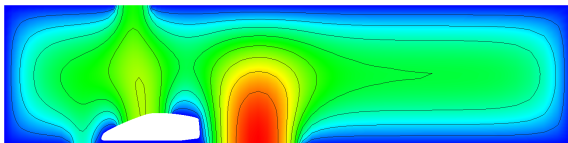
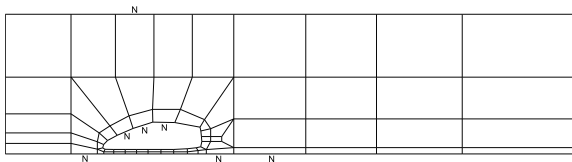
- use slightly anisotropic operator  $-\partial_{xx} u - 4\partial_{yy} u = 1$
- use multigrid-Krylov solvers:
  - 1-layer-ScaRC: MG-FGMRES4\_JAC\_D
  - 2-layer-ScaRC: MG-FGMRES4\_MG(T7)-BICG-JAC-D\_D
- use more complex grids (see next slides):
  - crossover-iso: 64 subdomains, 16.8M DOF (level 9), aspect ratio 2.91
  - crossover-aniso: aspect ratio 20.4
  - asmo: 70 subdomains, 18.4M DOF (level 9), aspect ratio 18.2



crossover

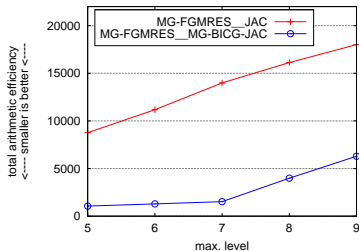
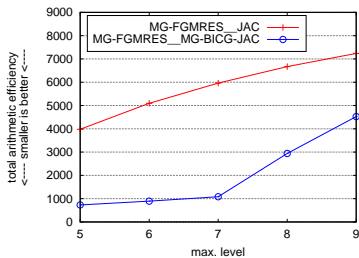
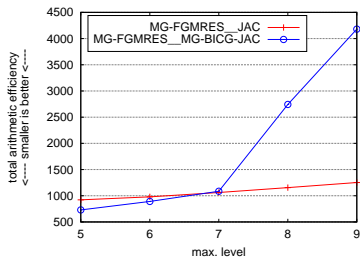


asmo



# 1-layer-ScaRC vs. 2-layer-ScaRC

crossover-iso (left), crossover-aniso (right), asmo (bottom)



Number of global smoothing steps  
( $\Rightarrow$  amount of communication)

lev	MG-FGMRES4 1-layer-ScaRC			MG-FGMRES4_MG 2-layer-ScaRC		
	cr-iso	cr-aniso	asmo	cr-iso	cr-aniso	asmo
5	88	352	776	32	32	48
6	96	464	1016	32	32	48
7	104	552	1296	32	32	48
8	112	624	1504	32	32	48
9	120	680	1688	32	32	56

- 2-layer-ScaRC successfully hides the local irregularities (anisotropies) from the global solver
- $\Rightarrow$  strongly favoured in a massively parallel computation

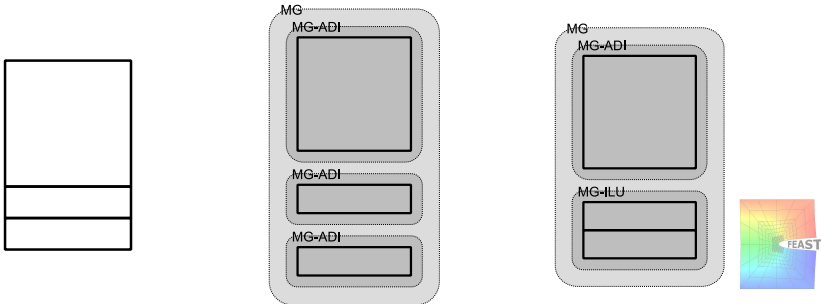


But:

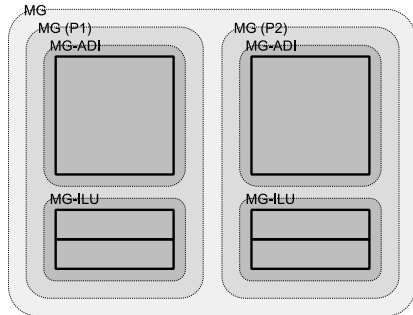
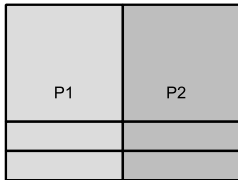
- for ADITriGS as local smoother, examples to show superiority of 2-layer-ScaRC not found yet
- current favourite: 1-layer-ScaRC solver BiCG-MG\_\_ADI\_\_D
- reason: for ADITriGS, mesh anisotropies are not really irregularities  
⇒ there is 'nothing to hide' within a 2-layer-ScaRC solver
- future attempts: massively parallel computations, more complicated physical equations, 3D



- first clustering strategy: 'merging subdomains'
- alleviates the problem of macro anisotropies (block-Jacobi character)
- idea: merge *several anisotropic* subdomains to obtain *one isotropic* subdomain
- disadvantage: tensor product property lost  
→ special smoothers (e. g., ADITriGS) have to be replaced by standard smoothers (e. g., ILU)
- gain numerical efficiency, (eventually) lose processor efficiency



- second clustering strategy: '3-layer-ScaRC'
- situation: several subdomains reside on one processor (figure: three SDs on proc.1, three SDs on proc. 2)
- add another MG scheme on 'intermediate' layer, i. e., per processor
- further minimises communication



## Part II

# Parallel Multilevel Solvers for Compressible Elasticity Problems



- 5 Introduction/Motivation
- 6 Elasticity
- 7 Discretisation
- 8 Solution Concept
- 9 Selected Numerical Examples



- 5 Introduction/Motivation
- 6 Elasticity
- 7 Discretisation
- 8 Solution Concept
- 9 Selected Numerical Examples



On the one hand:

- hardware-oriented library is tedious to develop and to maintain (multi-core architectures, vector processors, Cache-sizes, co-processors (GPUs, Cell), latency/speed of interconnects, compiler issues, ...)
- mandatory for high efficiency: a priori known data layout, especially of FE matrices

On the other hand:

- different physical applications  $\Rightarrow$  different matrix structures (heat transfer, elasticity, Navier-Stokes, Fluid-Solid-Interaction, 2D / 3D, etc.)
- 'application programmers' don't want to be bothered with technical details



Remedy: Realise *multivariate* operations (operators) as a series (set) of *scalar* operations (operators)

Advantages:

- facilitates strict separation of low-level kernel/library functionalities and high-level application code
- 'kernel programmers' can concentrate on the scalar equation case, do not have to heed all the physical applications
- 'application programmers' can concentrate on the application, do not have to heed processor architectures, MPI communication, matrix fill-in patterns, ...
- efficiency of the scalar kernel routines automatically available for multivariate problems
- kernel enhancements (new finite element, GPU solvers, ...) usable without any changes of the application code

→ prototypically demonstrated with **FEASTsolid**



- 5 Introduction/Motivation
- 6 Elasticity**
- 7 Discretisation
- 8 Solution Concept
- 9 Selected Numerical Examples



- literature:
  - Dietrich Braess. Finite Elemente. Springer, Berlin, 3rd edition, 2003.
  - Philippe G. Ciarlet. Mathematical Elasticity. Volume I: Three-Dimensional Elasticity, volume 20 of Studies in Mathematics and its Applications. North-Holland, Amsterdam, 1988.
  - Erwin Stein and Franz-Joseph Barthold. Elastizitätstheorie. In 'Der Ingenieurbau, Grundwissen: Werkstoffe, Elastizitätstheorie', Seiten 165-428. Ernst & Sohn, Berlin, 1996.
  - ...
- continuum mechanical discipline describing the deformation of solid elastic bodies
- solid and deformable: forces change the body's shape, but not its continuous coherence
- elastic: deformation process is reversible
- assumptions:
  - material is homogeneous and isotropic
  - only small deformations (linearised elasticity)



- solid body  $\bar{\Omega} \subset \mathbb{R}^3$
- boundary  $\partial\Omega =: \Gamma = \Gamma_D \cup \Gamma_N$
- fixed boundary  $\Gamma_D$  (Dirichlet) and movable boundary  $\Gamma_N$  (Neumann)
- deformation mapping  $\varphi : \bar{\Omega} \rightarrow \mathbb{R}^3$  with  $\det(\nabla\varphi) > 0$   
(orientation-preserving)
- displacements  $\mathbf{u}(\mathbf{x}) = (u_1(\mathbf{x}), u_2(\mathbf{x}), u_3(\mathbf{x}))^T$  of material point  $\mathbf{x} \in \bar{\Omega}$ ,  $\varphi = \mathbf{id} + \mathbf{u}$
- kinematic relation between displacements and strains:  
Green-St. Venant strain tensor

$$\mathbf{E} = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T + \nabla\mathbf{u}^T\nabla\mathbf{u})$$

- assume small deformations  $\rightarrow$  omit quadratic terms:  
linearised strain tensor

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T) \quad (1)$$

existence of a vector field (the Cauchy stress vector)  $\mathbf{t} : \bar{\Omega} \times \mathcal{S}_2 \rightarrow \mathbb{R}^3$   
( $\mathcal{S}_2 := \{ \mathbf{v} \in \mathbb{R}^3 \mid \|\mathbf{v}\| = 1 \}$ ) with:

- for arbitrary  $V \subset \bar{\Omega}$  ( $\mathbf{g}$  applied surface force,  $\mathbf{n}$  unit normal vector):

$$\mathbf{t}(\mathbf{x}, \mathbf{n}) = \mathbf{g}(\mathbf{x}), \quad \mathbf{x} \in \Gamma_N \cap \partial V$$

- axiom of force balance ( $\mathbf{f}$  applied body force):

$$\int_V \mathbf{f}(\mathbf{x}) \, dx + \int_{\partial V} \mathbf{t}(\mathbf{x}, \mathbf{n}) \, da = \mathbf{0}$$

- axiom of balance of angular momenta:

$$\int_V \mathbf{x} \times \mathbf{f}(\mathbf{x}) \, dx + \int_{\partial V} \mathbf{x} \times \mathbf{t}(\mathbf{x}, \mathbf{n}) \, da = \mathbf{0}$$



$\mathbf{f}$  continuous,  $\mathbf{t}(\cdot, \cdot)$  continuously differentiable w.r.t. first argument and continuous w.r.t. second argument  $\Rightarrow$  existence of a continuously differentiable symmetric tensor field  $\boldsymbol{\sigma} : \bar{\Omega} \rightarrow \mathbb{M}^3$  (Cauchy stress tensor) that satisfies

- $\mathbf{t}(\mathbf{x}, \mathbf{n}) = \boldsymbol{\sigma}(\mathbf{x})\mathbf{n}, \quad \mathbf{x} \in \bar{\Omega}, \mathbf{n} \in S_2,$

- the PDE

$$-\operatorname{div}(\boldsymbol{\sigma}(\mathbf{x})) = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \Omega,$$

- and the boundary conditions

$$\boldsymbol{\sigma}(\mathbf{x})\mathbf{n} = \mathbf{g}(\mathbf{x}), \quad \mathbf{x} \in \Gamma_N.$$

$\Rightarrow$  only 3 equations (PDE) for 9 unknowns  
(3 displacements  $\mathbf{u}$  + 6 stresses  $\boldsymbol{\sigma}$ )



- deformation depends on material properties
- relation between strains and stresses
- simplest case: Hooke's law for isotropic elastic material under small deformation:

$$\boldsymbol{\sigma} = 2\mu\boldsymbol{\varepsilon} + \lambda \operatorname{tr}(\boldsymbol{\varepsilon})\mathbf{I} \quad (2)$$

- Lamé constants  $\mu$  and  $\lambda$
- relation to Young's modulus  $E$  and Poisson ratio  $\nu$ :

$$\mu = \frac{E}{2(1 + \nu)}, \quad \lambda = \frac{E\nu}{(1 + \nu)(1 - 2\nu)}$$

- $\nu \rightarrow 0.5$  ( $\lambda \rightarrow \infty$ ): compressibility of material decreases

Material	$E$ [ N/m <sup>2</sup> ]	$\nu$ [-]	$\mu$ [ N/m <sup>2</sup> ]	$\lambda$ [ N/m <sup>2</sup> ]
Steel/Iron	$2.1 \cdot 10^{11}$	0.29	$8.14 \cdot 10^{10}$	$1.1 \cdot 10^{11}$
Lead	$1.6 \cdot 10^{10}$	0.44	$5.6 \cdot 10^9$	$4.1 \cdot 10^{10}$
Rubber	$2.5 \cdot 10^7$	0.499–0.5	$8.3 \cdot 10^6$	$4.1 \cdot 10^9 - \infty$



- pure displacement formulation: displacements as only unknowns (use (1) and (2) to eliminate  $\varepsilon$  and  $\sigma$ )
- elliptic boundary value problem - the Lamé Equation:

$$\begin{aligned} -2\mu \mathbf{div}(\varepsilon) - \lambda \nabla \operatorname{div}(\mathbf{u}) &= \mathbf{f} && \text{in } \Omega \\ \mathbf{u} &= \bar{\mathbf{u}} && \text{on } \Gamma_D \\ \sigma \mathbf{n} &= \mathbf{g} && \text{on } \Gamma_N \end{aligned}$$

- $\bar{\mathbf{u}}$  prescribed displacements on  $\Gamma_D$ ,  
 $\mathbf{f}$  given body forces (e. g., gravity),  
 $\mathbf{g}$  surface forces on  $\Gamma_N$



- bilinear form:

$$k(\mathbf{u}, \mathbf{v}) := \int_{\Omega} 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) + \lambda \operatorname{div}(\mathbf{u}) \operatorname{div}(\mathbf{v}) \, dx$$

- variational (weak) formulation: Find  $\mathbf{u} - \bar{\mathbf{u}} \in \mathbf{V} := \{\mathbf{v} \in H^1(\Omega)^3 \mid \mathbf{v} = \mathbf{0} \text{ on } \Gamma_D\}$  such that

$$k(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dx + \int_{\Gamma_N} \mathbf{g} \cdot \mathbf{v} \, da, \quad \mathbf{v} \in \mathbf{V}$$

- $k(\mathbf{u}, \mathbf{v})$  is  $\mathbf{V}$ -elliptic and continuous,

$$\exists \alpha, C > 0 : \alpha \|\mathbf{v}\|_1^2 \leq k(\mathbf{v}, \mathbf{v}) \leq C \|\mathbf{v}\|_1^2, \quad \mathbf{v} \in \mathbf{V}, \quad (3)$$

$\Rightarrow$  variational problem has a unique solution

- $\mathbf{v} \in H^m(\Omega) : \|\mathbf{v}\|_m := \sqrt{\sum_{|\alpha| \leq m} \|\partial^\alpha \mathbf{v}\|_0^2}$



- 5 Introduction/Motivation
- 6 Elasticity
- 7 Discretisation**
- 8 Solution Concept
- 9 Selected Numerical Examples



- space of bilinear polynomials over the basis element  $e^b = [-1, 1]^2$ :

$$Q_1(e^b) := \left\{ q \mid q(\xi, \eta) = \sum_{0 \leq i, j \leq 1} c_{ij} \xi^i \eta^j \right\}$$

- isoparametric concept:

$$e = \psi_e(e^b), \quad \psi_e \in Q_1(e^b)^2$$

for each element  $e$  of the triangulation  $\mathcal{T}^h$

- finite dimensional subspace  $\mathbf{V}^h \subset \mathbf{V}$  of  $Q_1$  finite elements:

$$\mathbf{V}^h := \left\{ \mathbf{v} \in C(\Omega)^2 \mid \mathbf{v}|_e \in Q_1(e)^2 \forall e \in \mathcal{T}^h, \mathbf{v} = \mathbf{0} \text{ on } \Gamma_D \right\}.$$

with

$$Q_1(e) := \left\{ q \circ \psi_e^{-1} \mid q \in Q_1(e^b) \right\},$$



- basis of  $\mathbf{V}^h$ : nodal basis functions

$$\phi_i : \bar{\Omega}^h \rightarrow \mathbb{R}, \quad \phi_i(\mathbf{x}_j) = \delta_{ij},$$

- finite element solution  $\mathbf{u}^h = (u_1^h, u_2^h) \in \mathbf{V}^h$  given in terms of the nodal basis functions

$$u_j^h(\mathbf{x}) = \sum_{i=1}^n (\mathbf{u}_j)_i \phi_i(\mathbf{x}), \quad j = 1, 2,$$

with unknown coefficient vector  $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2)^T \in \mathbb{R}^{2n}$

- resulting linear system:

$$\mathbf{K}\mathbf{u} = \mathbf{f}$$

with positive definite stiffness matrix  $\mathbf{K} \in \mathbb{R}^{2n \times 2n}$



- error between exact solution  $\mathbf{u}$  and computed FE solution  $\mathbf{u}^h$ ?
- Céa-Lemma:

$$\|\mathbf{u} - \mathbf{u}^h\|_1 \leq \frac{C}{\alpha} \inf_{\mathbf{v} \in \mathbf{V}^h} \|\mathbf{u} - \mathbf{v}\|_1.$$

relates discretisation error to approximation error  
(constants  $\alpha$  and  $C$  from (3))

- sufficient regularity  $\Rightarrow$  error estimates:

$$\|\mathbf{u} - \mathbf{u}^h\|_1 \leq ch \|\mathbf{u}\|_2$$

$$\|\mathbf{u} - \mathbf{u}^h\|_0 \leq ch^2 \|\mathbf{u}\|_2$$

( $h$  mesh width parameter)



- general problem: degenerating parameter ( $\rightarrow 0$  or  $\rightarrow \infty$ )
- non-uniform convergence w.r.t. this parameter
- possibly large errors in the solution, especially on coarser meshes
- ill-conditioned linear systems
- often results in too small displacements  $\Rightarrow$  'locking'
- 'volume locking' (part 3 of this lecture)
- 'shear locking' (next slide)



- problematic: long thin structures and bending dominated deformations
- cantilever beam with  $L \gg H$  (here:  $\frac{L}{H} \in \{2, 8, 32\}$ )



- Korn's inequality:

$$\exists c_K = c_K(\Omega, \Gamma_D) > 0 : \quad \|\varepsilon(\mathbf{v})\|_0^2 \geq c_K \|\mathbf{v}\|_1^2, \quad \mathbf{v} \in \mathbf{V}$$

(required to prove the ellipticity (3) of the bilinear form)

- Korn constant depends on the domain shape and boundary conditions





- inverse of the Korn constant:

$$\frac{1}{c_K} = \sup_{\mathbf{v} \in \mathbf{V}, \epsilon(\mathbf{v}) \neq 0} \frac{\|\mathbf{v}\|_1}{\|\epsilon(\mathbf{v})\|_0}$$

- Example: prescribed analytical solution

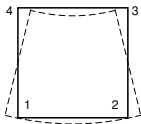
$$\mathbf{v} = \begin{pmatrix} x_1 x_2 \\ -\frac{1}{2} x_1^2 \end{pmatrix}, \quad \epsilon(\mathbf{v}) = \begin{pmatrix} x_2 & 0 \\ 0 & 0 \end{pmatrix},$$

i. e.

$$\frac{L}{H} \rightarrow \infty \quad \Rightarrow \quad \frac{1}{c_K} \rightarrow \infty$$

- $\frac{1}{c_K}$  enters Céa-Lemma and a priori estimates





- special problem of  $Q_1$  bilinear finite elements
- illustration: basis element under pure bending

$$\begin{aligned}(\mathbf{u}_1)_1 &= -(\mathbf{u}_1)_2 = (\mathbf{u}_1)_3 = -(\mathbf{u}_1)_4, \\ (\mathbf{u}_2)_1 &= (\mathbf{u}_2)_2 = -(\mathbf{u}_2)_3 = -(\mathbf{u}_2)_4.\end{aligned}$$

- characterised by zero shear strains:

$$\varepsilon_{12}(\mathbf{u}^h) = \frac{1}{2} \left( \frac{\partial u_1^h}{\partial \eta} + \frac{\partial u_2^h}{\partial \xi} \right) = 0 \quad (4)$$

- four bilinear basis functions:

$$\begin{aligned}\phi_1(\xi, \eta) &= \frac{1}{4}(1 - \xi)(1 - \eta), & \phi_2(\xi, \eta) &= \frac{1}{4}(1 + \xi)(1 - \eta), \\ \phi_3(\xi, \eta) &= \frac{1}{4}(1 + \xi)(1 + \eta), & \phi_4(\xi, \eta) &= \frac{1}{4}(1 - \xi)(1 + \eta)\end{aligned}$$



- resulting finite element function:

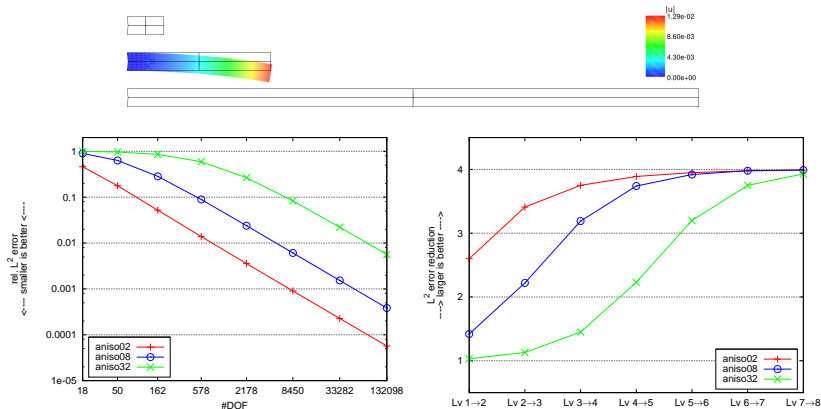
$$u_1^h(\xi, \eta) = \sum_{i=1}^4 (\mathbf{u}_1)_i \phi_i(\xi, \eta) = (\mathbf{u}_1)_1 \xi \eta,$$

$$u_2^h(\xi, \eta) = \sum_{i=1}^4 (\mathbf{u}_2)_i \phi_i(\xi, \eta) = -(\mathbf{u}_2)_1 \eta$$

$$\Rightarrow \epsilon_{12}(\mathbf{u}^h) = \frac{1}{2}(\mathbf{u}_1)_1 \xi$$

- zero shear strains (4)  $\Rightarrow (\mathbf{u}_1)_1 = 0$
- only zero displacements can fulfil the pure bending condition
- non-zero displacements produce spurious shear strains





- large errors on coarser meshes
- expected  $L^2$  error reduction of 4 only on finer meshes



- 5 Introduction/Motivation
- 6 Elasticity
- 7 Discretisation
- 8 Solution Concept**
- 9 Selected Numerical Examples



- motivation: **reduction to scalar components**
- rewrite left hand side of BVP (now 2D):

$$\begin{aligned} & -2\mu \mathbf{div}(\boldsymbol{\varepsilon}) - \lambda \nabla \operatorname{div}(\mathbf{u}) \\ = & \begin{pmatrix} (2\mu + \lambda)\partial_{11} + \mu\partial_{22} & (\mu + \lambda)\partial_{12} \\ (\mu + \lambda)\partial_{21} & \mu\partial_{11} + (2\mu + \lambda)\partial_{22} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \end{aligned}$$

- rewrite left hand side of weak form:

$$\begin{aligned} k(\mathbf{u}, \mathbf{v}) &= \int_{\Omega} (2\mu + \lambda)\partial_1 u_1 \partial_1 v_1 + \mu\partial_2 u_1 \partial_2 v_1 \, dx \\ &+ \dots \text{ (mixed terms)} \\ &+ \int_{\Omega} \mu\partial_1 u_2 \partial_1 v_2 + (2\mu + \lambda)\partial_2 u_2 \partial_2 v_2 \, dx \\ &= \dots \text{ (next slide)} \end{aligned}$$



- rewrite left hand side of weak form:

$$\begin{aligned}
 k(\mathbf{u}, \mathbf{v}) = & \underbrace{\int_{\Omega} \left[ \begin{pmatrix} 2\mu+\lambda & 0 \\ 0 & \mu \end{pmatrix} \nabla u_1 \right] \cdot \nabla v_1 \, dx}_{=: k_{11}(u_1, v_1)} + k_{12}(u_2, v_1) \\
 & + k_{21}(u_1, v_2) + \underbrace{\int_{\Omega} \left[ \begin{pmatrix} \mu & 0 \\ 0 & 2\mu+\lambda \end{pmatrix} \nabla u_2 \right] \cdot \nabla v_2 \, dx}_{=: k_{22}(u_2, v_2)}
 \end{aligned}$$

- FE discretisation ( $Q_1$ )  $\Rightarrow$  block structured linear equation system:

$$\begin{pmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{pmatrix}$$

(‘separate displacement ordering’; not used, e. g., in FEAP)

- $\mathbf{K}_{11}$  and  $\mathbf{K}_{22}$  correspond to scalar elliptic operators (‘anisotropic Laplace operator’)



- basic iteration - block preconditioned Richardson method:

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

- defect computation:

$$\mathbf{f} - \mathbf{K}\mathbf{u}^k = \begin{pmatrix} \mathbf{f}_1 - \mathbf{K}_{11}\mathbf{u}_1^k - \mathbf{K}_{12}\mathbf{u}_2^k \\ \mathbf{f}_2 - \mathbf{K}_{21}\mathbf{u}_1^k - \mathbf{K}_{22}\mathbf{u}_2^k \end{pmatrix}$$

- e.g., block Jacobi or block Gauß-Seidel preconditioner:

$$\tilde{\mathbf{K}}_{\text{BJac}}^{-1} = \begin{pmatrix} \mathbf{K}_{11}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{22}^{-1} \end{pmatrix} \quad \tilde{\mathbf{K}}_{\text{BGS}}^{-1} = \begin{pmatrix} \mathbf{K}_{11} & \mathbf{0} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{pmatrix}^{-1}$$

- reduction to scalar components  $\Rightarrow$  exploit FEAST concepts!
- note: each global matrix  $\mathbf{K}_{ij}$  is represented by local 9-band matrices corresponding to TP subdomains



- condition number of the preconditioned system:

$$\kappa := \kappa(\tilde{\mathbf{K}}_{\text{BJac}}^{-1} \mathbf{K}) \leq \frac{1}{c_K} \left( 1 + \frac{1}{1 - 2\nu} \right)$$

- three observations:

- $\kappa$  does not depend on mesh size parameter  $h$   
 $\Rightarrow$  single-grid outer solver (i. e., Krylov method) can be sufficient
- $\kappa$  depends on Korn's constant  $c_K = c_K(\Omega, \Gamma_D)$   
 $\Rightarrow$  convergence depends on boundary conditions / geometry  
(e. g., cantilever beam)
- $\kappa$  depends on Poisson ratio  $\nu$   
 $\Rightarrow$  condition of the system increases with incompressibility of the material, i. e., when  $\nu \rightarrow 0.5$  (part 3 of this lecture)

- block Gauß-Seidel or block SOR instead of block Jacobi  
 $\Rightarrow$  no such estimate, but more efficient in most cases



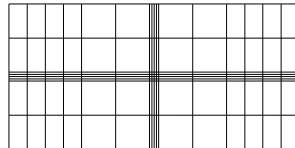
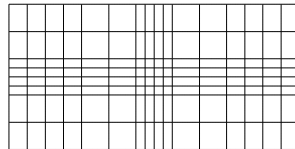
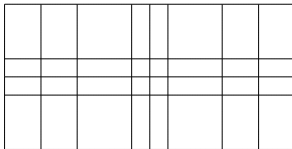
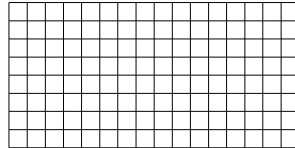
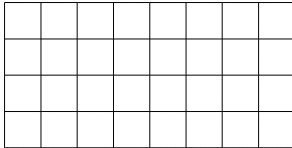
- 5 Introduction/Motivation
- 6 Elasticity
- 7 Discretisation
- 8 Solution Concept
- 9 Selected Numerical Examples**



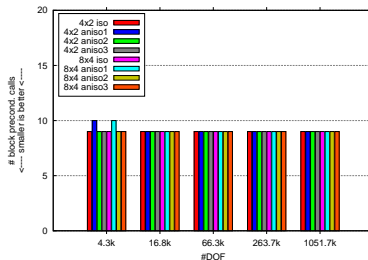
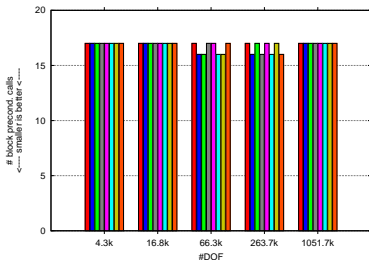
- dependence on mesh anisotropies
- dependence on geometry
- 1-layer-ScaRC vs. 2-layer-ScaRC
- parallel efficiency



$4 \times 2$  subdomains (left),  $8 \times 4$  subdomains (right)  
(figures: each subdomain consists of 4 coarse grid cell)



BiCG-BJac (left), BiCG-BSor (right)



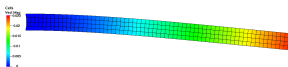
- convergence independent of mesh refinement level (outer single-grid Krylov solver!)
- BSor roughly halves number of iterations compared to BJac
- anisotropies are fully 'absorbed' by the scalar subsolves



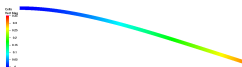
aniso04



aniso16

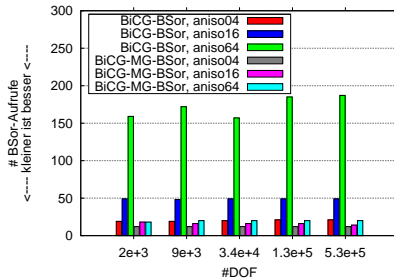


aniso64



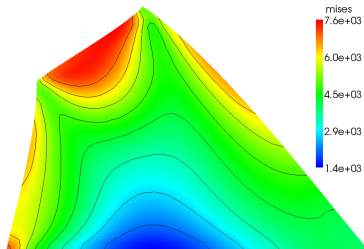
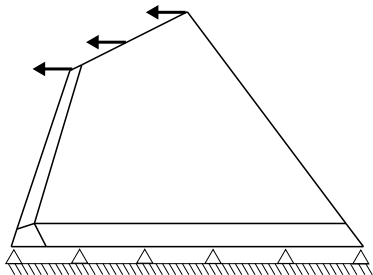
Two solvers:

- **BiCGstab-BSor**:  
MG only on scalar layer
- **BiCGstab-MG-BSor**:  
*additional* MG on multivariate layer



- scalar MG guarantees independence of mesh refinement level
- but: strong dependence on geometry (length of the beam)
- additional multivariate MG significantly decreases this dependency
- $\Rightarrow$  strong *scalar* methods are not sufficient to efficiently handle numerical difficulties that only exist on the multivariate layer!





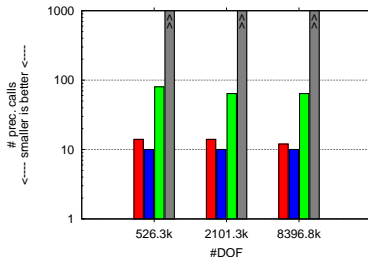
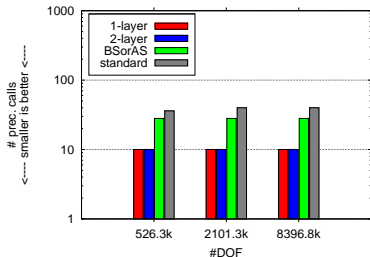
- two variants: iso (max. aspect ratio 2.2) and aniso (max. aspect ratio 63.6) (only aniso shown)
- compare three solvers (ScaRC-solver in square brackets):
  - using 1-layer-ScaRC:
 

```
BiCG-MG(1,V11)-BSor [MG(1e-1)-FGMRES4__JAC__D]
```
  - using 2-layer-ScaRC:
 

```
BiCG-MG(1,V11)-BSor [MG(1e-1)-FGMRES4__MG-BICG-JAC-D__D]
```
  - 'standard solver': MG(F11)-BiCG(2)-Jac (using point Jacobi smoother, disregarding the block structure)



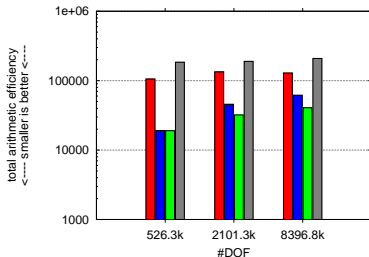
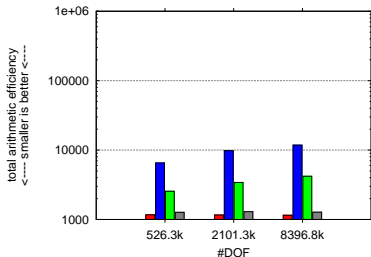
iso (left), aniso (right)  
(ignore solver BSorAS)



- convergence behaviour of 1-layer-ScaRC and 2-layer-ScaRC variants (nearly) independent of the configuration
- the standard solver suffers significantly
- 2-layer-ScaRC always needs 1 iteration per call, 1-layer-ScaRC between 6 (iso) and 70 (aniso) iterations  
⇒ communication amount of 2-layer-ScaRC variant much smaller!

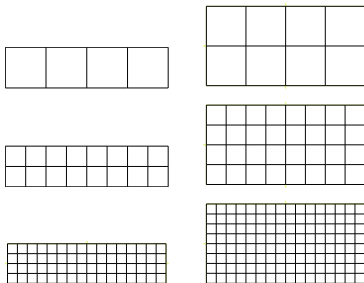
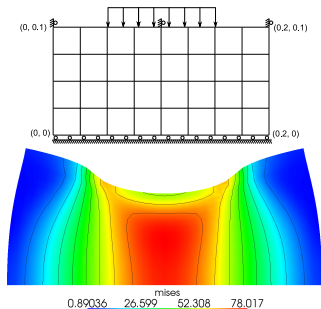


iso (left), aniso (right)



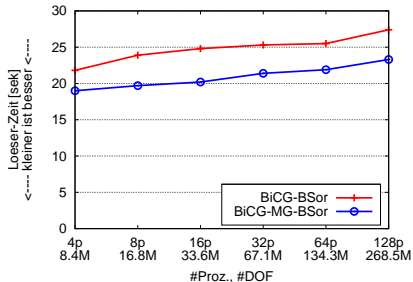
- arithmetic costs of 2-layer-ScaRC very high on iso configuration
- arithmetic costs of 1-layer-ScaRC and the standard solver increase drastically on the aniso configuration  $\Rightarrow$  2-layer-ScaRC superior
- just 'proof of concept'! (with ADITriGS: TAE  $\approx$  1000)





- weak scalability: increase problem size and resources by the same factor
- 'perfect weak scalability': runtime remains constant
- smallest problem:  $4 \times 1$  subdomains, 4.2 M vertices, 4 processors
- largest problem:  $16 \times 8$  subdomains, 134.2 M vertices, 128 proc.





- good weak scalability
- comparable to scalar FEAST solvers
- parallel efficiency of the scalar FEAST library fully transfers to multivariate elasticity solvers



## Part III

# Parallel Multilevel Solvers for Incompressible Elasticity Problems in Saddle Point Form



- 10 Volume Locking
- 11 Mixed Formulation
- 12 Discretisation
- 13 Saddle Point Solvers



- 10 Volume Locking
- 11 Mixed Formulation
- 12 Discretisation
- 13 Saddle Point Solvers



- rubber-like materials are (nearly) incompressible, important for many industrial applications
- sensitive w.r.t. to volume changes, large increase of stored energy
- stored energy function corresponding to Hooke's law (2):

$$\check{W}(\boldsymbol{\varepsilon}) = \mu \operatorname{tr}(\boldsymbol{\varepsilon}^2) + \frac{\lambda}{2} \operatorname{tr}(\boldsymbol{\varepsilon})^2 = \mu \operatorname{tr}(\boldsymbol{\varepsilon}^2) + \frac{\lambda}{2} \operatorname{div}(\mathbf{u})^2$$

- only small volume changes:  $\operatorname{div}(\mathbf{u}) \rightarrow 0$   
(linearisation of  $\det(\mathbf{F}) \rightarrow 1$ , where  $\mathbf{F} = \mathbf{Grad}(\boldsymbol{\varphi}) = \mathbf{I} + \mathbf{Grad}(\mathbf{u})$  is the deformation gradient)
- material constant  $\lambda \rightarrow \infty$
- limit case: (pure) incompressibility:  $\lambda = \infty, \det(\mathbf{F}) = 1, \operatorname{div}(\mathbf{u}) = 0$
- additional constraint  $\Rightarrow$  system over-determined



- bilinear form

$$k(\mathbf{u}, \mathbf{v}) := \int_{\Omega} 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) + \lambda \operatorname{div}(\mathbf{u}) \operatorname{div}(\mathbf{v}) \, dx$$

is  $\mathbf{V}$ -elliptic and continuous (see eq. (3)):

$$\exists \alpha, C > 0 : \alpha \|\mathbf{v}\|_1^2 \leq k(\mathbf{v}, \mathbf{v}) \leq C \|\mathbf{v}\|_1^2, \quad \mathbf{v} \in \mathbf{V}$$

- Hence:  $\alpha \lesssim \mu$  and  $C \gtrsim \mu + \lambda$ , i.e.,

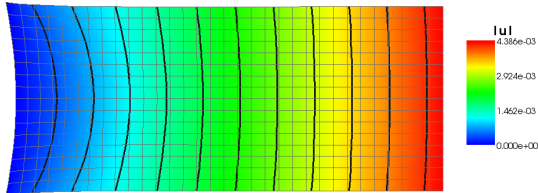
$$\lambda \rightarrow \infty \quad \Rightarrow \quad \frac{C}{\alpha} \gtrsim \frac{\mu + \lambda}{\mu} \rightarrow \infty$$

- $\frac{C}{\alpha}$  enters Céa-Lemma and a priori estimates
- pure displacement formulation  $\rightarrow$  'volume locking' ('material locking')

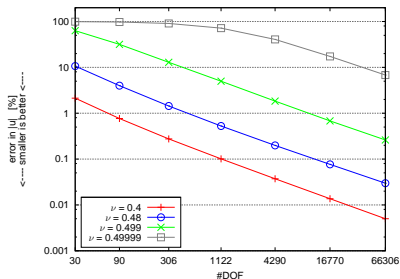
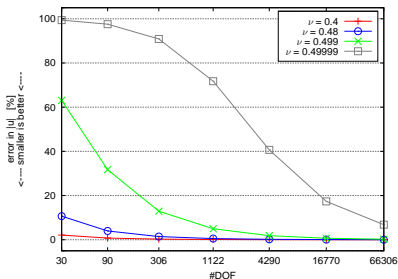


# Volume Locking Example

stretched beam



## relative displacement error in point A



- large errors on coarser meshes
- expected error reduction only on finer meshes
- plus: severe problems for linear solvers
- more critical than the shear locking effect
- $\Rightarrow$  pure displacement formulation not suited for incompressible material



- enhanced assumed strains (EAS)  
(nonconforming FE method with additional local DOF)
- selected reduced integration (SRI)  
(the part  $\lambda \operatorname{div}(\mathbf{u}) \operatorname{div}(\mathbf{v})$  is integrated with one instead of four cubature points)
- pro: both result in positive definite systems, no additional unknowns
- contra: checkerboard modes, special transfer and smoothing operations for multigrid method necessary, hourglass instabilities (EAS), not usable for  $\nu = 0.5$  (pure incompressibility)

our approach: mixed formulation



- 10 Volume Locking
- 11 Mixed Formulation**
- 12 Discretisation
- 13 Saddle Point Solvers



- pure displacement formulation:

$$-2\mu \mathbf{div}(\boldsymbol{\varepsilon}) - \lambda \nabla \operatorname{div}(\mathbf{u}) = \mathbf{f}$$

- replace critical part that reacts sensitively to volume changes by pressure-like variable:

$$p := -\lambda \operatorname{div}(\mathbf{u})$$

- resulting BVP:

$$\begin{aligned} -2\mu \mathbf{div}(\boldsymbol{\varepsilon}) + \nabla p &= \mathbf{f} && \text{in } \Omega \\ -\operatorname{div}(\mathbf{u}) - \frac{1}{\lambda} p &= 0 && \text{in } \Omega \\ \mathbf{u} &= \bar{\mathbf{u}} && \text{on } \Gamma_D \\ \boldsymbol{\sigma} \mathbf{n} &= \mathbf{g} && \text{on } \Gamma_N \end{aligned}$$

- → mixed displacement pressure ( $\mathbf{u}/p$ ) formulation



- function spaces:

$$\mathbf{V} := \{ \mathbf{v} \in H^1(\Omega)^3 \mid \mathbf{v} = \mathbf{0} \text{ on } \Gamma_D \} \quad \text{and} \quad M := L_2(\Omega) \quad (5)$$

- variational formulation: Find  $(\mathbf{u} - \bar{\mathbf{u}}, p) \in \mathbf{V} \times M$  such that

$$\begin{aligned} \int_{\Omega} 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) \, dx - \int_{\Omega} \operatorname{div}(\mathbf{v}) p \, dx &= L(\mathbf{v}), & \mathbf{v} \in \mathbf{V} \\ - \int_{\Omega} \operatorname{div}(\mathbf{u}) q \, dx - \frac{1}{\lambda} \int_{\Omega} p q \, dx &= 0, & q \in M \end{aligned}$$

where

$$L(\mathbf{v}) := \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dx + \int_{\Gamma_N} \mathbf{g} \cdot \mathbf{v}$$

and  $A : B$  matrix scalar product of two matrices A and B.



- bilinear forms:

$$a(\mathbf{u}, \mathbf{v}) := \int_{\Omega} 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) \, dx$$

$$b(\mathbf{v}, p) := - \int_{\Omega} \operatorname{div}(\mathbf{v}) p \, dx$$

$$c(p, q) := - \int_{\Omega} pq \, dx$$

- variational formulation: Find  $(\mathbf{u} - \bar{\mathbf{u}}, p) \in \mathbf{V} \times M$  such that

$$a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = L(\mathbf{v}), \quad \mathbf{v} \in \mathbf{V} \quad (6a)$$

$$b(\mathbf{u}, q) + \frac{1}{\lambda} c(p, q) = 0, \quad q \in M \quad (6b)$$

- penalty term:  $\frac{1}{\lambda} \rightarrow 0$  for  $\nu \rightarrow 0.5$
- $\Rightarrow$  saddle point problem with a penalty term



- $\nu = 0.5$ , penalty term vanishes:  $\frac{1}{\lambda} = 0$
- variational formulation: Find  $(\mathbf{u} - \bar{\mathbf{u}}, p) \in \mathbf{V} \times M$  such that

$$a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = L(\mathbf{v}), \quad \mathbf{v} \in \mathbf{V} \quad (7a)$$

$$b(\mathbf{u}, q) = 0, \quad q \in M \quad (7b)$$

- corresponding PDE:

$$\begin{aligned} -2\mu \operatorname{div}(\boldsymbol{\varepsilon}) + \nabla p &= \mathbf{f} && \text{in } \Omega \\ \operatorname{div}(\mathbf{u}) &= 0 && \text{in } \Omega \end{aligned}$$

- replace strain tensor  $\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$  (= 'symmetric gradient') by gradient  $\nabla \mathbf{u}$ , use relation  $\Delta \mathbf{u} = \operatorname{div}(\nabla \mathbf{u})$
- $\Rightarrow$  *classical Stokes equation for viscous incompressible fluids:*

$$\begin{aligned} -2\mu \Delta \mathbf{u} + \nabla p &= \mathbf{f} && \text{in } \Omega \\ \operatorname{div}(\mathbf{u}) &= 0 && \text{in } \Omega. \end{aligned}$$

with  $\Omega$  fluid,  $\mathbf{u}$  fluid's velocity,  $p$  pressure,  $2\mu$  dynamic viscosity



- continuity and  $\mathbf{V}$ -ellipticity of bilinear form  $a(\cdot, \cdot)$  not sufficient
- compatibility condition between  $\mathbf{V}$  and  $M$  necessary
- *Babuška-Brezzi conditions*:
  - 1 The continuous bilinear form  $a(\cdot, \cdot)$  is  $\mathbf{V}_0$ -elliptic:

$$\exists \alpha > 0 : a(\mathbf{v}, \mathbf{v}) \geq \alpha \|\mathbf{v}\|_1^2, \quad \mathbf{v} \in \mathbf{V}_0,$$

where  $\mathbf{V}_0 := \{\mathbf{v} \in \mathbf{V} \mid b(\mathbf{v}, q) = 0, \quad q \in M\}$  (weaker than  $\mathbf{V}$ -ell.)

- 2 The continuous bilinear form  $b(\cdot, \cdot)$  fulfils the inf-sup condition:

$$\exists \beta > 0 : \inf_{0 \neq q \in M} \sup_{0 \neq \mathbf{v} \in \mathbf{V}} \frac{b(\mathbf{v}, q)}{\|\mathbf{v}\|_1 \|q\|_0} \geq \beta. \quad (8)$$

- bilinear forms  $a(\cdot, \cdot)$  and  $b(\cdot, \cdot)$  and spaces (5) fulfil these conditions  
 $\Rightarrow$  problem (7) has a unique solution
- Additionally:  $c(\cdot, \cdot)$  is continuous and  $c(q, q) \leq 0, q \in M$   
 $\Rightarrow$  (6) has a unique solution



## Advantages:

- mathematically well established
- pure incompressibility ( $\nu = 0.5$ ) computable
- exploit discretisation and solution techniques from the field of Computational Fluid Dynamics (CFD)
- standard multigrid components sufficient

## Disadvantages:

- additional pressure variable  $\Rightarrow$  larger linear systems
- saddle point structure  $\Rightarrow$  special solving techniques necessary
- finite element spaces for  $\mathbf{u}$  and  $p$  cannot be chosen arbitrarily



- 10 Volume Locking
- 11 Mixed Formulation
- 12 Discretisation**
- 13 Saddle Point Solvers



- Discrete variational form: Find  $(\mathbf{u}^h - \bar{\mathbf{u}}, p^h) \in \mathbf{V}^h \times M^h$  such that

$$\begin{aligned} a(\mathbf{u}^h, \mathbf{v}) + b(\mathbf{v}, p^h) &= L(\mathbf{v}), & \mathbf{v} \in \mathbf{V}^h \\ b(\mathbf{u}^h, q) + \frac{1}{\lambda} c(p^h, q) &= 0, & q \in M^h. \end{aligned}$$

- FE discretisation  $\rightarrow$  linear saddle point system:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C}^c \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix}.$$

- matrix  $\mathbf{C}^c$  contains the 'compressibility terms', i. e.,  $\mathbf{C}^c = 0$  in the incompressible limit
- $\mathbf{A} \in R^{n \times n}$  symmetric positive definite,  $\mathbf{C}^c \in R^{m \times m}$  negative semidefinite (possibly zero),  $\mathbf{B}$  has full rank
  - $\Rightarrow$  saddle point system is indefinite with  $n$  positive and  $m$  negative eigenvalues
  - $\Rightarrow$  Schur complement matrix  $\mathbf{S} := \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} - \mathbf{C}^c$  is symmetric negative definite



- *stable FE pairs*:  $\mathbf{V}^h$  and  $M^h$  fulfil discrete analogon of the Babuška-Brezzi conditions, especially the discrete inf-sup condition:

$$\exists \beta > 0 : \inf_{0 \neq q^h \in M^h} \sup_{0 \neq \mathbf{v}^h \in \mathbf{V}^h} \frac{b(\mathbf{v}^h, q^h)}{\|\mathbf{v}^h\|_1 \|q^h\|_0} \geq \beta.$$

- Examples:
  - $Q_k/P_{k-1}$ ,  $k \geq 2$  (often used:  $k = 2$ )
  - $P_2/P_1$  (Taylor-Hood)
  - $Q_2/Q_1$  serendipity (omit central displacement node)
  - ...
- $\Rightarrow$  robust convergence  $\mathbf{u}^h \rightarrow \mathbf{u}$  ( $h \rightarrow 0$ ) w.r.t. critical parameter  $\lambda$



- *unstable FE pairs*: inf-sup condition not fulfilled
- Examples:
  - equal order pairs:  $P_k/P_k$ ,  $Q_k/Q_k$
  - $Q_1/P_0$  (checkerboard instabilities)
  - ...

- interpretation with alternative formulation of inf-sup condition:

$$\exists \beta > 0 \forall q^h \in M^h \exists \mathbf{v}^h \in \mathbf{V}^h : b(\mathbf{v}^h, q^h) \geq \beta \|\mathbf{v}^h\|_1 \|q^h\|_0$$

- assume

$$\exists q^h \neq 0 \in M^h \forall \mathbf{v}^h \in \mathbf{V}^h : b(\mathbf{v}^h, q^h) = 0$$

(non-trivial kernel)

- 'undetected', spurious pressure mode  
⇒ problem not uniquely solvable or unstable
- pressure space is 'too large' w.r.t. to displacement space → not enough displacement functions to 'detect' these pressure modes



- equal-order pairs very attractive from implementational point of view
- idea: add bilinear form that provides additional ‘detection’ of spurious pressure modes
- relax the continuity equation by adding element-wise contributions

$$c_{\text{GLS}}(\mathbf{u}^h, p^h; \mathbf{v}, q) := -\frac{\alpha}{2\mu} \sum_e h_e^2 \left( -2\mu \mathbf{div}(\boldsymbol{\varepsilon}(\mathbf{u}^h)) + \nabla p^h, -2\mu \mathbf{div}(\boldsymbol{\varepsilon}(\mathbf{v})) + \nabla q \right)_e$$

with  $(\mathbf{a}, \mathbf{b})_e := \int_e \mathbf{a}(\mathbf{x}) \cdot \mathbf{b}(\mathbf{x}) dx$ ,  $\alpha$  stabilisation parameter and  $h_e$  characteristic element size

- → *Galerkin Least-Squares stabilisation, residual-based stabilisation*
- consistent (stabilisation terms vanish when exact solution is entered)



- difficult to compute second derivatives with  $Q_1$  functions
- simplification:

$$c_{PPP}(p^h, q) := -\frac{\alpha}{2\mu} \sum_e h_e^2 (\nabla p^h, \nabla q)_e$$

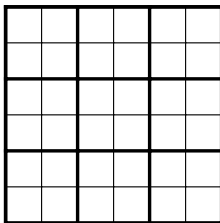
- *penalty pressure Poisson stabilisation*
- non-consistent (penalty error of  $\|\operatorname{div}(\mathbf{u})\|_0 = O(h)$ )
- robust convergence w.r.t. to critical parameters  $\lambda$  and  $h$



possibilities to compute characteristic element size:

$$h_e^{\text{area}} := \sqrt{\text{area}(e)}, \quad h_e^{\text{min}} := \min_{i=1,\dots,4} \{h_e^i\}, \quad h_e^{\text{max}} := \max_{i=1,\dots,4} \{h_e^i\},$$

where  $h_e^i$  is the length of the  $i$ -th edge of element  $e$



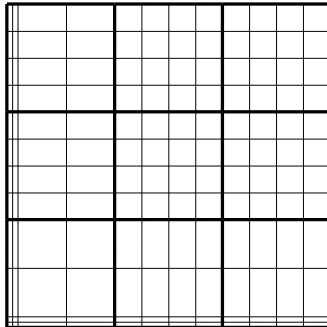
→ suited for isotropic (or mildly anisotropic) meshes, hence

$$\Rightarrow c_{\text{PPP}}(p^h, q) = -\frac{\alpha}{2\mu} \sum_e h_e^2 (\nabla p^h, \nabla q)_e$$

is appropriate for such meshes

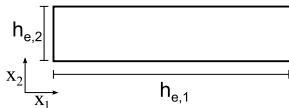


Problematic: anisotropic cartesian meshes



$$h_e = ?$$





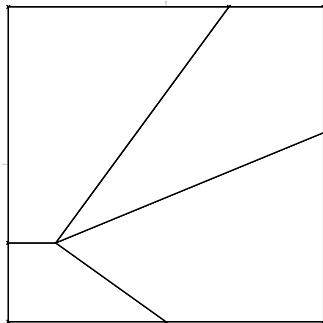
consider partial derivatives (Becker/Rannacher):

$$c_{\text{PPBR}}(p^h, q) := -\frac{\alpha}{2\mu} \sum_e h_{e,1}^2 (\partial_1 p^h, \partial_1 q)_e + h_{e,2}^2 (\partial_2 p^h, \partial_2 q)_e$$

→ only suited for cartesian meshes

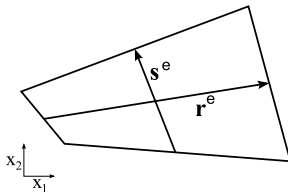


Problematic: irregular meshes (anisotropic and non-cartesian)



$$h_e = ???$$



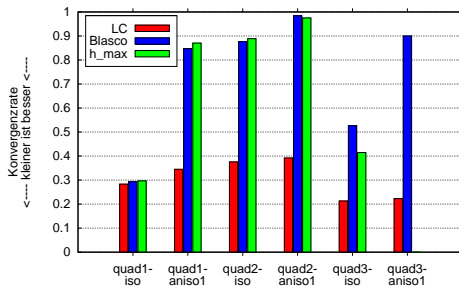
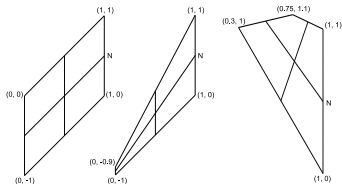


idea: include element geometry  $\rightarrow$  local coordinate system / directional derivatives:

$$c_{PPPLC}(p^h, q) := -\frac{\alpha}{2\mu} \sum_e (\mathbf{r}^e \cdot \nabla p^h, \mathbf{r}^e \cdot \nabla q)_e + (\mathbf{s}^e \cdot \nabla p^h, \mathbf{s}^e \cdot \nabla q)_e.$$



- accuracy of the FE solution  
 ⇒ no significant differences, no 'favourite', depends on configuration
- but: significant effects on iterative saddle point solvers



solver: BiCGstab-SCBTria[BSor]

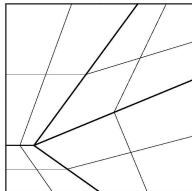
(ignore 'Blasco' bars)

stabilisation using local coordinate system superior!



$L^2$ -error reduction rates  
and timings

pure displacement  
vs.  
mixed formulation

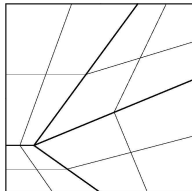


<b>displ</b>	$\nu = 0.4$		$\nu = 0.49999$	
	# elem.	red	sec	red
1024	3.961	0.1	1.055	0.6
4096	3.987	0.2	1.102	3.4
16 K	3.996	0.7	1.307	22.2
66 K	3.998	2.9	1.812	107.7
262 K	4.000	12.9	2.053	381.4
$L^2$ -error	1.490e-06		2.573e-03	

<b>mixed</b>	$\nu = 0.4$		$\nu = 0.49999$	
	# elem.	red	sec	red
1024	3.988	0.3	4.080	0.3
4096	3.997	0.7	4.051	0.7
16 K	4.000	2.4	4.031	3.1
66 K	4.004	9.4	4.018	14.5
262 K	4.023	34.4	4.039	45.5
$L^2$ -error	1.047e-06		8.520e-07	

$L^2$ -error reduction rates  
and timings

pure displacement  
vs.  
mixed formulation



<b>displ</b>	$\nu = 0.4$		$\nu = 0.49999$	
	# elem.	red	sec	red
1024	3.961	0.1	1.055	0.6
4096	3.987	0.2	1.102	3.4
16 K	3.996	0.7	1.307	22.2
66 K	3.998	2.9	1.812	107.7
262 K	4.000	12.9	2.053	381.4
$L^2$ -error	1.490e-06		2.573e-03	

<b>mixed</b>	$\nu = 0.4$		$\nu = 0.49999$	
	# elem.	red	sec	red
1024	3.988	0.3	4.080	0.3
4096	3.997	0.7	4.051	0.7
16 K	4.000	2.4	4.031	3.1
66 K	4.004	9.4	4.018	14.5
262 K	4.023	34.4	4.039	45.5
$L^2$ -error	1.047e-06		8.520e-07	

- 10 Volume Locking
- 11 Mixed Formulation
- 12 Discretisation
- 13 Saddle Point Solvers**

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

- compressibility and stabilisation terms  $\rightarrow \mathbf{C}$
- **Schur complement**  $\mathbf{S} := \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} - \mathbf{C}$   
(dense matrix  $\Rightarrow$  only implicitly available)
- two categories: **decoupled** and **coupled** solvers

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

- compressibility and stabilisation terms  $\rightarrow \mathbf{C}$
- Schur complement  $\mathbf{S} := \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} - \mathbf{C}$   
(dense matrix  $\Rightarrow$  only implicitly available)
- two categories: decoupled and coupled solvers

First decoupled solver: **Block-Preconditioning approach**

- Krylov-space method with block-preconditioner (SCBTria)

$$\begin{pmatrix} \tilde{\mathbf{A}} & \mathbf{B} \\ \mathbf{0} & -\tilde{\mathbf{S}} \end{pmatrix}^{-1} = \begin{pmatrix} \tilde{\mathbf{A}}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{I} & -\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\tilde{\mathbf{S}}^{-1} \end{pmatrix}$$

- $\tilde{\mathbf{A}}$  and  $\tilde{\mathbf{S}}$  preconditioners for  $\mathbf{A}$  and  $\mathbf{S}$
- $\tilde{\mathbf{A}}^{-1} \hat{=}$  approximate solution with techniques from pure displacement formulation (e. g., one step BSor with local ScaRC solves)

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

- compressibility and stabilisation terms  $\rightarrow \mathbf{C}$
- Schur complement  $\mathbf{S} := \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} - \mathbf{C}$   
(dense matrix  $\Rightarrow$  only implicitly available)
- two categories: decoupled and coupled solvers

Second decoupled solver: **Pressure Schur Complement approach**

- ‘Cancelling’ displacements  $\mathbf{u} = \mathbf{A}^{-1}(\mathbf{f} - \mathbf{B}\mathbf{p})$

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

- Basic iteration (*scalar* system!):

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

- Acceleration: Krylov-space method

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

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

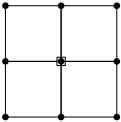
- stabilisation terms are of magnitude  $\mathcal{O}(h^2)$   
⇒ can be omitted for preconditioning
- $\mathbf{S}$  is spectrally equivalent to the (scalar!) pressure mass matrix  $\mathbf{M}_p$   
⇒  $\mathbf{M}_p$  suited preconditioner

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

Solution of the saddle point problem mainly reduced to  
the solution of scalar systems

Second category: **coupled** multigrid scheme with **Vanka-smoother**

- Idea: coupling  $\mathbf{u}$  and  $\mathbf{p}$  on element layer, successive solution of small dense local systems (multiplicative DD-method)
- modification for  $Q_1/Q_1$  (Becker, 1995) (patch of 4 elements, pressure DOF only in the center, i. e.,  $n = 2 \cdot 9 = 18$ )

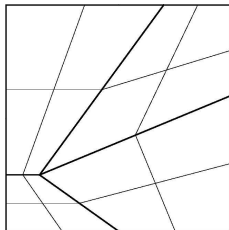
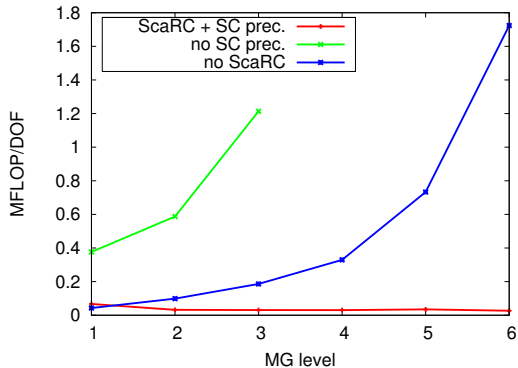


$$\begin{pmatrix} a_{11} & \dots & a_{1n} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{n1} & \dots & a_{nn} & b_n \\ b_1 & \dots & b_n & c \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_n \\ p \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_n \\ d \end{pmatrix}$$

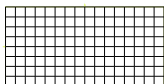
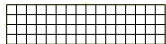
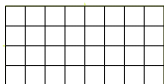
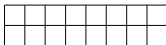
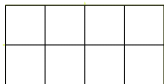
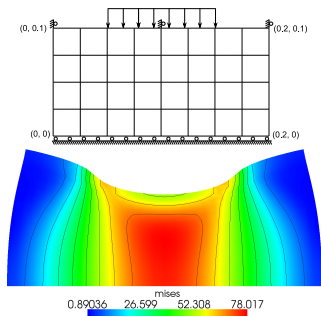
- + no Schur complement preconditioner necessary
- special modifications for anisotropic meshes necessary
- reduction to scalar systems not possible  $\Rightarrow$  **ScaRC solvers not usable!**

**fundamental alternative of the strategy  
'reduction to scalar systems'**

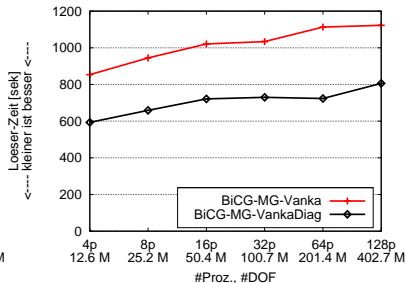
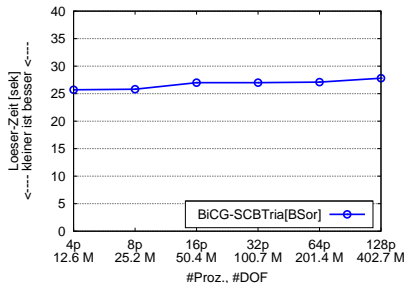
## Influence of ScaRC and Schur complement preconditioning



( $\nu = 0.5$ , BiCGstab-SCBTria[BSor])



- consider: weak scalability
- 12.6 M to 402.7 M unknowns, 4 to 128 processors
- nearly incompressible material



decoupled method (based on ScaRC)  
ca. 20-40 times faster