

# Multilevel Methods with Local Smoothing

Bärbel Janssen,  
joint work with Guido Kanschat

Institute of Applied Mathematics  
University of Heidelberg

August 24, 2010



## Convergence rates

may not be significantly lower than on regular meshes without local refinement

## Optimal complexity

each step should be performed with optimal computational complexity

## Matrix structures

involved must be easy to obtain in a finite element code and may not severely increase memory requirements



## Smoothing

should only happen on subgrids without hanging nodes to simplify cell-based smoothers

## Continuity

the scheme should be able to use the continuity conditions across faces for any finite element



## WARNING

- Multigrid on locally refined meshes only works with **discontinuous finite elements** right now.
- It is not clear, whether the paradigm of local smoothing we use is applicable to continuous elements with hanging nodes.
- In fact, most people you meet on conferences seem to deny this.

[taken from the documentation of deal.II]



## WARNING

- Multigrid on locally refined meshes only works with **discontinuous finite elements** right now.
- It is not clear, whether the paradigm of local smoothing we use is applicable to continuous elements with hanging nodes.
  - In fact, most people you meet on conferences seem to deny this.

[taken from the documentation of deal.II]



## WARNING

- Multigrid on locally refined meshes only works with **discontinuous finite elements** right now.
- It is not clear, whether the paradigm of local smoothing we use is applicable to continuous elements with hanging nodes.
- In fact, most people you meet on conferences seem to deny this.

[taken from the documentation of deal.II]



## What was available in deal.II?

- MG for discontinuous finite elements (global and local refinement)
- MG for continuous finite elements (**only for global refinement**)

## MG for continuous finite elements on locally refined grids

Idea: Adjust the existing MG for discontinuous elements to continuous elements.

## Problem

How do we have to treat the hanging nodes correctly?

## Poisson Problem

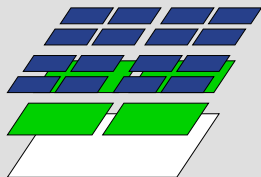
Find  $u \in V$  s.t.

$$a(u, v) = (\nabla u, \nabla v) = (f, v) \quad \forall v \in V.$$

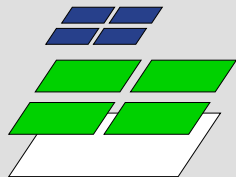
Here  $V$  denotes the subspace of  $H^1(\Omega)$  with suitable boundary conditions.

We obtain a linear system  $Au = f$ .

### global refinement



### local refinement







## Problem on mesh $\mathbb{T}_\ell$

$$A_\ell u_\ell = f_\ell.$$

to be solved by a Krylov-space solver. To understand this, it is sufficient to consider the

## Richardson iteration

$$u_\ell^{k+1} = u_\ell^k + \omega r_\ell^k$$

$r^k$  being the residual of the current iteration step  $k$ .

## Preconditioned iteration

$$u_\ell^{k+1} = u_\ell^k + \omega P_\ell^{-1} r_\ell^k.$$



## Algorithm (V-Cycle)

Let  $P_0 = A_0$ . Set  $x^{(0)} = 0$  and compute  $P_\ell^{-1}d_\ell$  by the following steps:

- 1 (Pre-smoothing) Compute  $x^{(m_\ell)}$  iteratively by

$$x^{(i)} = x^{(i-1)} + S_\ell^{(i)}(d_\ell - A_\ell x^{(i-1)}), \quad i = 1, \dots, m_\ell.$$

- 2 (Coarse grid correction) Let

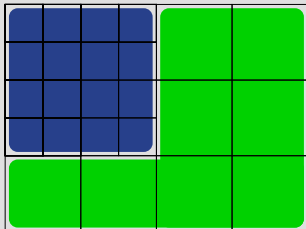
$$y^{(0)} = x^{(m_\ell)} + R_{\ell-1}^T P_{\ell-1}^{-1} R_{\ell-1} (d_\ell - A_\ell x^{(m_\ell)}).$$

- 3 (Post-smoothing) Compute  $y^{(m_\ell)}$  iteratively by

$$y^{(i)} = y^{(i-1)} + S_\ell^{(m_\ell+i)}(d_\ell - A_\ell y^{(i-1)}), \quad i = 1, \dots, m_\ell.$$

- 4 Set  $P_\ell^{-1}d_\ell = y^{(m_\ell)}$ .

Subspaces on locally refined meshes for DG case



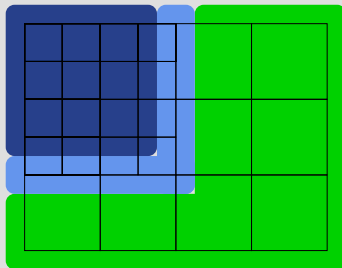
Splitting with discontinuous elements

$$\begin{pmatrix} A_\ell^{SS} & A_\ell^{SL} \\ A_\ell^{LS} & A_\ell^{LL} \end{pmatrix} \begin{pmatrix} x_\ell^S \\ x_\ell^L \end{pmatrix} = \begin{pmatrix} f_\ell^S \\ f_\ell^L \end{pmatrix}$$

$$V_\ell = V_\ell^S \oplus V_\ell^L$$

[G. Kanschat: Multi-level methods for discontinuous Galerkin FEM on locally refined meshes, Comput. & Struct., vol. 82, pp. 2437-2445, 2004]

## Subspaces on locally refined meshes for continuous case



## Splitting with continuous elements

$$\begin{pmatrix} A_l^{SS} & A_l^{SE} \\ A_l^{ES} & A_l^{EE} \\ A_l^{LE} & A_l^{LL} \end{pmatrix} \begin{pmatrix} x_l^S \\ x_l^E \\ x_l^L \end{pmatrix} = \begin{pmatrix} f_l^S \\ f_l^E \\ f_l^L \end{pmatrix}, \text{ with } V_l = V_l^S \oplus V_l^E \oplus V_l^L$$



## Algorithm (V-Cycle)

Let  $P_0 = A_0$ . Set  $x^{(0)} = 0$  and compute  $P_\ell^{-1}d_\ell$  by the following steps:

- 1 (Pre-smoothing) Compute  $x^{(m_\ell)}$  iteratively by

$$x^{(i)} = x^{(i-1)} + S_\ell^{(i)}(d_\ell - A_\ell x^{(i-1)}), \quad i = 1, \dots, m_\ell.$$

- 2 (Coarse grid correction) Let

$$y^{(0)} = x^{(m_\ell)} + R_{\ell-1}^T P_{\ell-1}^{-1} R_{\ell-1} (d_\ell - A_\ell x^{(m_\ell)}).$$

- 3 (Post-smoothing) Compute  $y^{(m_\ell)}$  iteratively by

$$y^{(i)} = y^{(i-1)} + S_\ell^{(m_\ell+i)}(d_\ell - A_\ell y^{(i-1)}), \quad i = 1, \dots, m_\ell.$$

- 4 Set  $P_\ell^{-1}d_\ell = y^{(m_\ell)}$ .



## Algorithm (V-Cycle)

Let  $P_0 = A_0$ . Set  $x^{(0)} = 0$  and compute  $P_\ell^{-1}d_\ell$  by the following steps:

- 1 (Pre-smoothing) On the subspace  $V_\ell^S$  only, compute  $x^{(m_\ell)}$  iteratively by

$$x^{(i)} = x^{(i-1)} + \tilde{S}_\ell^{(i)}(d_\ell - A_\ell x^{(i-1)}), \quad i = 1, \dots, m_\ell.$$

- 2 (Coarse grid correction) Let

$$y^{(0)} = x^{(m_\ell)} + R_{\ell-1}^T P_{\ell-1}^{-1} R_{\ell-1} (d_\ell - A_\ell x^{(m_\ell)}).$$

- 3 (Post-smoothing) Compute  $y^{(m_\ell)}$  iteratively by

$$y^{(i)} = y^{(i-1)} + S_\ell^{(m_\ell+i)}(d_\ell - A_\ell y^{(i-1)}), \quad i = 1, \dots, m_\ell.$$

- 4 Set  $P_\ell^{-1}d_\ell = y^{(m_\ell)}$ .



## Algorithm (V-Cycle)

Let  $P_0 = A_0$ . Set  $x^{(0)} = 0$  and compute  $P_\ell^{-1}d_\ell$  by the following steps:

- 1 (Pre-smoothing) On the subspace  $V_\ell^S$  only, compute  $x^{(m_\ell)}$  iteratively by

$$x^{(i)} = x^{(i-1)} + \tilde{S}_\ell^{(i)}(d_\ell - A_\ell x^{(i-1)}), \quad i = 1, \dots, m_\ell.$$

- 2 (Coarse grid correction) Let

$$y^{(0)} = x^{(m_\ell)} + R_{\ell-1}^T P_{\ell-1}^{-1} (R_{\ell-1}^S (d_\ell^S - A_\ell^{SS} x_S^{(m_\ell)}) + d_\ell^E - A_\ell^{ES} x_S^{(m_\ell)}).$$

- 3 (Post-smoothing) Compute  $y^{(m_\ell)}$  iteratively by

$$y^{(i)} = y^{(i-1)} + S_\ell^{(m_\ell+i)}(d_\ell - A_\ell y^{(i-1)}), \quad i = 1, \dots, m_\ell.$$

- 4 Set  $P_\ell^{-1}d_\ell = y^{(m_\ell)}$ .



## Algorithm (V-Cycle)

Let  $P_0 = A_0$ . Set  $x^{(0)} = 0$  and compute  $P_\ell^{-1}d_\ell$  by the following steps:

- 1 (Pre-smoothing) On the subspace  $V_\ell^S$  only, compute  $x^{(m_\ell)}$  iteratively by

$$x^{(i)} = x^{(i-1)} + \tilde{S}_\ell^{(i)}(d_\ell - A_\ell x^{(i-1)}), \quad i = 1, \dots, m_\ell.$$

- 2 (Coarse grid correction) Let

$$y^{(0)} = x^{(m_\ell)} + R_{\ell-1}^T P_{\ell-1}^{-1} (R_{\ell-1}^S (d_\ell^S - A_\ell^{SS} x_S^{(m_\ell)}) + d_\ell^E - A_\ell^{ES} x_S^{(m_\ell)}).$$

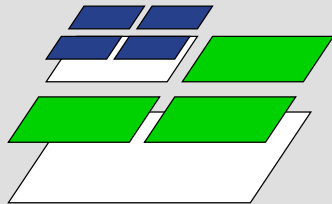
- 3 (Post-smoothing) Compute  $y^{(m_\ell)}$  iteratively by

$$y^{(i)} = y^{(i-1)} + \tilde{S}_\ell^{(m_\ell+i)}(d_\ell - A_\ell y^{(i-1)}), \quad i = 1, \dots, m_\ell.$$

- 4 Set  $P_\ell^{-1}d_\ell = y^{(m_\ell)}$ .



## A hierarchy of three meshes with local refinement



Active cells are colored.

We restrict the smoothing to the subspace  $V_\ell^S$ .

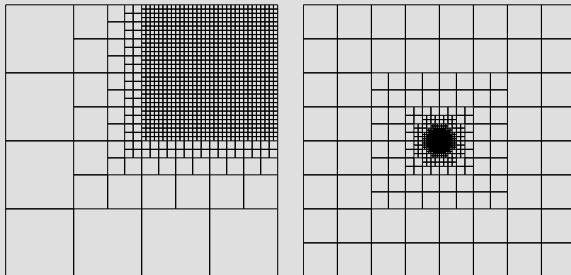
## Prolongation and restriction

- restrict without creating any contributions in  $V_\ell^E$
- prolongate and take the coupling in  $V_\ell^E$  into account

The locally refined grids have to fulfill certain criteria:

- Cells sharing a common face may only differ by one refinement level.
- Levels of all active cells sharing one vertex differ by a maximum of one.

## Refinement of the positive quadrant and a circle

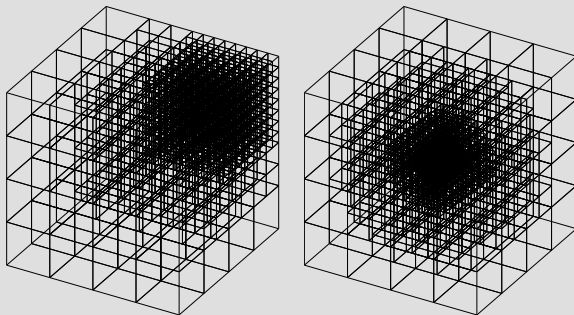




## Iteration steps and convergence rates with $Q_1$ elements in 2D

$L$	global		quadrant		circle	
	$n$	$r$	$n$	$r$	$n$	$r$
2	1	16.00	1	15.94	4	2.92
3	4	2.92	6	2.13	6	1.84
4	6	2.03	7	1.57	7	1.72
5	6	1.81	7	1.54	7	1.66
6	6	1.71	7	1.55	8	1.43
7	6	1.67	7	1.57	7	1.58
8	7	1.65	7	1.58	7	1.49

## Refinement of the first octant and a ball





## Iteration steps and convergence rates in 3D

$L$	$Q_1$ -elements						$Q_2$ -elements					
	global		octant		ball		global		octant		ball	
	$n$	$r$	$n$	$r$	$n$	$r$	$n$	$r$	$n$	$r$	$n$	$r$
2	1	16.	1	16.	4	3.2	4	2.7	5	2.2	6	1.9
3	4	3.2	6	2.1	6	1.8	6	1.9	7	1.6	7	1.6
4	5	2.0	7	1.5	6	1.8	7	1.6	7	1.5	7	1.6
5	6	1.8	7	1.5	6	1.7	7	1.5	7	1.5	7	1.5
6	6	1.7	7	1.5	8	1.4	7	1.4	7	1.5	8	1.3
7	7	1.7	7	1.5	7	1.4	8	1.4	7	1.5	7	1.5



## Conclusions

- Multigrid for adaptive meshes was introduced
- Finite elements are continuous
- Hanging nodes were taken into account
- Optimal convergence due to local smoothing

## Outlook

- Other smoothers for higher order elements
- Multigrid for Raviart-Thomas elements

Thank you very much for your attention.