Hybridized Discontinuous Galerkin Methods
Theory and Implementation in DUNE

Christian Waluga

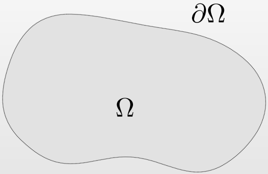
Joint work with Herbert Egger (Uni Graz)

1st DUNE User Meeting, Stuttgart

Outline

Outline

Model problem



Model problem: Poisson's equation

Given $f \in L^2(\Omega)$, find $u \in H^1(\Omega)$, such that

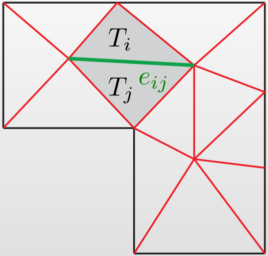
$$\begin{aligned} -\Delta u &= f, && \text{on } \Omega \\ u &= 0, && \text{on } \partial\Omega \end{aligned}$$

Assume $\Omega \subset \mathbb{R}^d$ is bounded, sufficiently regular domain.

Notation:

- $(u, v)_D := \int_D u\, v\, dx \quad \forall\, D \subseteq \Omega$
- $\|v\|_{0,D} := \sqrt{(v, v)_D}, \quad |v|_{1,D} := \sqrt{(\boldsymbol{\nabla} v, \boldsymbol{\nabla} v)_D}$
- We will abbreviate $\boldsymbol{H}^1(\Omega) := \left[H^1(\Omega)\right]^d$ etc.

Meshes and basic notation



Meshes: (hanging nodes also possible)

$$\mathcal{T}_h := \{\, T_i \,\}, \text{ where } \overline{\Omega} = \bigcup \overline{T_i}$$
$$\partial\mathcal{T}_h := \{\, \partial T_i \,\}$$
$$\mathcal{E}_h := \{\, e_{ij} \;:\; e_{ij} = \partial T_i \cap \partial T_j,\; i > j \,\}$$

Broken Sobolev spaces:

$$H^s(\mathcal{T}_h) := \{\, v \in L^2(\Omega) \;:\; v|_T \in H^s(T) \text{ for all } T \in \mathcal{T}_h \,\}$$
$$(u,v)_{\mathcal{T}_h} := \sum_{T\in\mathcal{T}_h} (u,v)_{L^2(T)}, \quad \|v\|_{\mathcal{T}_h} := (v,v)^{1/2}_{\mathcal{T}_h}, \quad \text{etc.}$$

Jump and average:

$$[\![v]\!] := v|_{T_i} - v|_{T_j}, \quad \{v\} := \tfrac{1}{2}(v|_{T_i} + v|_{T_j})$$

Note: functions defined on $\mathcal{E}_h$ can be interpreted as functions on $\partial\mathcal{T}_h$.

A simple DG method

Example: Symmetric Interior Penalty Galerkin (SIPG[1])

Find $u \in H^s(\mathcal{T}_h) \cap H^1_0(\Omega)$, $s > 3/2$, such that

$$a_h^{SIPG}(u,v) = (f,v)_{\mathcal{T}_h}, \qquad \forall\, v \in H^s(\mathcal{T}_h),$$

where we define a symmetric bilinear form

$$\begin{aligned} a_h^{SIPG}(u,v) :=& (\boldsymbol{\nabla} u, \boldsymbol{\nabla v})_{\mathcal{T}_h} - (\{\partial_n u\}, [\![v]\!])_{\mathcal{E}_h} \\ & - (\{\partial_n v\}, [\![u]\!])_{\mathcal{E}_h} + \tau([\![u]\!],[\![v]\!])_{\mathcal{E}_h}. \end{aligned}$$

Here, $\tau := \alpha \frac{p^2}{h}$ is a penalty parameter, with $\alpha \in \mathbb{R}^+$.

Natural energy norm for the discrete analysis:

$$\|v\|_{1,h} := \left(\|\boldsymbol{\nabla} v\|^2_{\mathcal{T}_h} + \|\tau^{1/2}[\![v]\!]\|^2_{\mathcal{E}_h} \right)^{1/2}.$$

[1] e.g. [Arnold 1982], [Rivière 2008]

Hybridized DG method

Example: Hybridized Symmetric Interior Penalty Galerkin

Find $(u, \hat{u}) \in (H^s(\mathcal{T}_h) \cap H^1_0(\Omega)) \times L^2(\mathcal{E}_h)$, $s > 3/2$, such that

$$a_h(u, \hat{u}; v, \hat{v}) = (f, v)_{\mathcal{T}_h}, \qquad \forall\, (v, \hat{v}) \in H^s(\mathcal{T}_h) \times L^2(\mathcal{E}_h),$$

where $\hat{u} := \{u\}$ and $\hat{v} := \{v\}$. We define a symmetric bilinear form

$$\begin{aligned} a_h(u, \hat{u}; v, \hat{v}) :=& (\boldsymbol{\nabla} u, \boldsymbol{\nabla} \boldsymbol{v})_{\mathcal{T}_h} - (\partial_n u, v - \hat{v})_{\partial\mathcal{T}_h} \\ & - (\partial_n v, u - \hat{u})_{\partial\mathcal{T}_h} + \tau(u - \hat{u}, v - \hat{v})_{\partial\mathcal{T}_h}. \end{aligned}$$

Here, $\tau := \alpha \frac{p^2}{h}$ is a penalty parameter, with $\alpha \in \mathbb{R}^+$.

Natural energy norm for the discrete analysis:

$$\|(v, \hat{v})\|_{1,h} := \left(\|\boldsymbol{\nabla} v\|^2_{\mathcal{T}_h} + \|\tau^{1/2}(v - \hat{v})\|^2_{\partial\mathcal{T}_h} \right)^{1/2}.$$

Hybridization: From DG to HDG (4/4)

Given a triangulation $\mathcal{T}_h$, define finite dimensional spaces V_h and $\widehat{V}_h$:

$$V_h := \{ v \in H^1(\mathcal{T}_h) \ : \ v|_T \in \mathcal{P}^p(T),\ \forall\, T \in \mathcal{T}_h \}$$
$$\hat{V}_h := \{ \hat{v} \in L^2(\mathcal{E}_h) \ : \ v|_E \in \mathcal{P}^p(E),\ \forall\, E \in \mathcal{E}_h,\ \hat{v} = 0 \text{ on } \partial\Omega \}$$

Discrete problem: Find $(u_h, \hat{u}_h) \in V_h \times \widehat{V}_h$ such that

$$a_h(u_h, \hat{u}_h; v_h, \hat{v}_h) = (f, v_h)_\Omega, \qquad \forall (v_h, \hat{v}_h) \in V_h \times \widehat{V}_h.$$

Stability analysis: For any $\alpha > \alpha_0$, there holds[2]

$$a_h(v_h, \hat{v}_h; v_h, \hat{v}_h) \geq \tfrac{1}{2} \|(v_h, \hat{v}_h)\|^2_{1,h}, \qquad \forall (v_h, \hat{v}_h) \in V_h \times \widehat{V}_h,$$

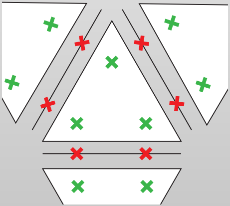
where α_0 depends on the element shape. One can choose α *'large enough'* (e.g. $\alpha = 10$) or explicitly compute $\alpha_0|_T$ on d-simplices and d-hypercubes.

[2] e.g. [Egger 2008]

Some remarks

- The hybridized method is consistent by construction!
- Standard error analysis yields optimal error estimates.
- Stability is independent of the particular choice of $\widehat{V}_h$.
- More general boundary conditions are possible.
- Other problems were also investigated, e.g.
  - Convection-diffusion [Egger and Schöberl 2009]
  - Stokes problem [Cockburn et. al. 2010], [Egger and W. 2010b]
  - Oseen problem (in preparation)

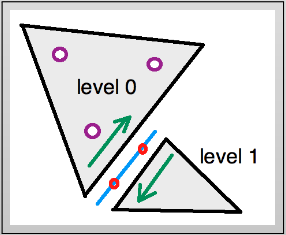
- Assembly in an element-wise fashion.
- Static condensation on element level.
- Upwinding can be easily incorporated.
- Locally varying polynomial degrees and nonconforming h-refinements possible.



HDG methods: Implementation (1/2)

Implementation in DUNE

- Implementation uses `dune-pdelab` and the core modules.
- Approximations in the interior of elements by monomials (using `MonomLocalFiniteElementMap`)
- Approximations on the element borders require some extra work:
  - We use the `IntersectionIndexSet` provided in PDELab and a self-written `FaceMonomLocalFiniteElementMap` to define a grid function space on the faces of the mesh.
  - **Problem:** The orientation of intersections may differ in two intersecting elements.

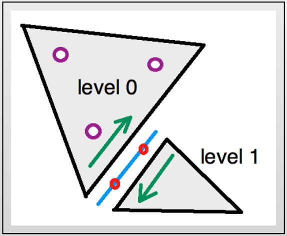


This causes problems when mapping from intersection coordinates to coordinates inside the element.

HDG methods (2/2)

Implementation in DUNE

- The only solution we know so far is a little helper that finds the corresponding intersection in the outside element if it has a lower index or a higher level than the current intersection.



- We then map the coordinates as follows:

```
if(wrongintersection)
    return rightintersection->geometryInOutside().global(x);
else
    return rightintersection->geometryInInside().global(x);
```

- The big disadvantage here is that we need to instantiate a quadratic number of intersections (performance issues?!)

Numerical example: Kovaznay (1/3)

Example: Oseen problem; $\Omega = (-0.5, 2) \times (-0.5, 1.5)$ [Kovasznay 1947]

$$\left.\begin{aligned} -\Delta\,\boldsymbol{u} + \boldsymbol{w}\nabla\boldsymbol{u} + \nabla p &= \boldsymbol{0} \\ \operatorname{div}\boldsymbol{u} &= 0 \end{aligned}\right\} \text{ on } \Omega, \qquad \boldsymbol{u} = \boldsymbol{u}_{exact} \text{ on } \partial\Omega$$

Exact solution:

$$\boldsymbol{u}(x,y) = \begin{pmatrix} 1 - \exp(\lambda x)\cos(2\pi y) \\ \frac{\lambda}{2\pi}\exp(\lambda x)\sin(2\pi y) \end{pmatrix},$$

$$p(x,y) = -\tfrac{1}{2}\exp(2\lambda x) + \overline{p},$$

with parameters λ and $\overline{p}$ given as

$$\lambda := \frac{-8\pi^2}{\nu^{-1} + \sqrt{\nu^{-2} + 16\pi^2}}$$

and $\overline{p} = 2\int_{-\frac{1}{2}}^{\frac{3}{2}} \exp(2\lambda x)\,dx$.

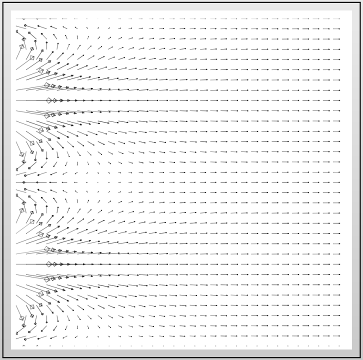


Figure: Velocity vectors ($\nu = 1/10$)

Numerical example: Kovaznay (2/3)

level	dimension		L^2 error		energy error	
$\mathbf{P_1 - P_0}$	K	S	error	rate	error	rate
0	578	326	37.5126	–	229.691	–
1	2248	1240	12.6979	1.56	143.706	0.68
2	8864	4832	3.70855	1.78	80.5438	0.84
3	35200	19072	0.97223	1.93	41.9258	0.94
4	140288	75766	0.24589	1.98	21.2533	0.98
$\mathbf{P_2 - P_1}$	K	S	error	rate	error	rate
0	1056	468	29.0005	–	217.573	–
1	4128	1776	4.34306	2.74	20.0395	3.44
2	16320	6912	0.63415	2.78	5.61554	1.84
3	64896	27264	0.08338	2.93	1.45829	1.95
4	258816	108288	0.01054	2.98	0.36827	1.99
$\mathbf{P_3 - P_2}$	K	S	error	rate	error	rate
0	1660	610	10.7941	–	93.9881	–
1	6512	2312	0.97279	3.47	14.7614	2.67
2	25792	8992	0.07140	3.77	2.08979	2.82
3	102656	34456	0.00461	3.95	0.26982	2.95
4	409600	140800	0.00029	3.98	0.03397	2.99

Table: **Kovsznay flow:** Errors of the numerical solution for different inf-sup stable finite element approximations and a sequence of uniformly refined meshes.

Numerical example: Kovaznay (3/3)

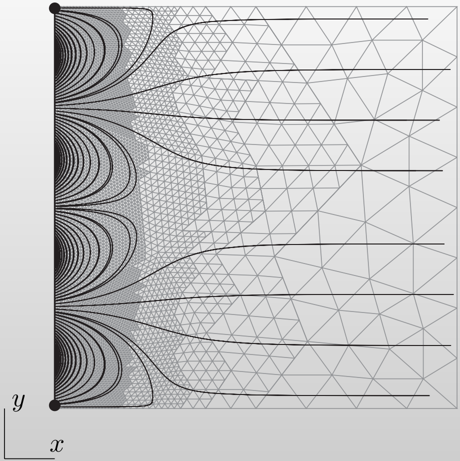


Figure: **Kovasznay flow** ($\nu = 1$): Streamlines and nonconforming mesh after 4 subsequent h-refinements.

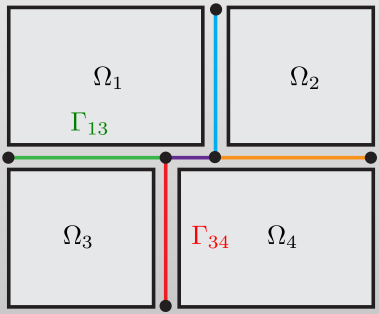
Outline

Hybridized mortar method (1/2)

Similarly as for the HDG method, we can define a **hybridized mortar method**[3]:

$$V_h := \{\, v \in H_0^1(\Omega_h) \;:\; v|_T \in \mathcal{P}^p(T),\ \forall\, T \in \mathcal{T}_h \,\}$$

$$\hat{V}_h := \{\, \hat{v} \in L^2(\Gamma_h) \;:\; v|_E \in \mathcal{P}^p(E),\ \forall\, E \in \mathcal{T}_h(\Gamma_h) \,\}$$



partition:

$$\Omega_h := \{\Omega_1, \Omega_2, \ldots, \Omega_N\}$$

interfaces:

$$\Gamma_{ij} := \partial\Omega_i \cap \partial\Omega_j$$

$$\Gamma_h := \{\Gamma_{ij}\}$$

skeleton:

$$\Gamma := \bigcup \Gamma_{ij}$$

[3] cf. [Egger 2008]

Hybridized mortar method (2/2)

Discrete problem: Find $(u_h, \hat{u}_h) \in V_h \times \widehat{V}_h$ such that

$$a_h(u_h, \hat{u}_h; v_h, \hat{v}_h) = (f, v_h)_\Omega, \qquad \forall (v_h, \hat{v}_h) \in V_h \times \widehat{V}_h.$$

where we define

$$\begin{aligned} a_h(u, \hat{u}, v, \hat{v}) :=& (\nabla u, \nabla v)_{\Omega_h} - (\partial_n u, v - \hat{v})_{\partial\Omega_h} \\ & - (\partial_n v, u - \hat{u})_{\partial\Omega_h} + \tau (u - \hat{u}, v - \hat{v})_{\partial\Omega_h}. \end{aligned}$$

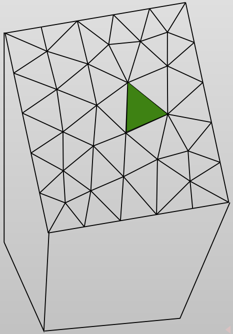
Important results:

- No direct coupling between subdomain solutions.
- *'Elimination'* on subdomains leads to a Schur complement system for the hybrid variables only ($\rightarrow$ domain decomposition methods).
- Space for the hybrid variable can be chosen with great flexibility (no inf-sup-condition necessary for multiplier).
- Hybridized mortar methods for other problems were also analyzed, e.g. Maxwell [Hollaus et. al. 2010], Stokes [Egger and W. 2010a].
- For the finest partition $\Omega_h = \mathcal{T}_h$, we recover the hybridized DG method.

Hybrid mortar: Implementation

Implementation in DUNE

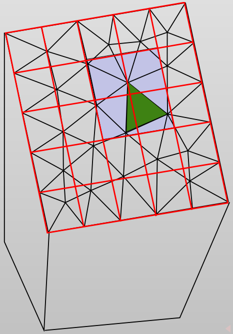
- In many applications, there exists a parametrization for the interface (e.g. planar, cylindrical, spherical, ...)
- **Idea:** Lagrange multipliers live on $d-1$ dimensional structured meshes that are transformed to the physical space.
- Overlaps between multiplier mesh and subdomain meshes are computed using `dune-grid-glue` by C. Engwer and O. Sander.



Hybrid mortar: Implementation

Implementation in DUNE

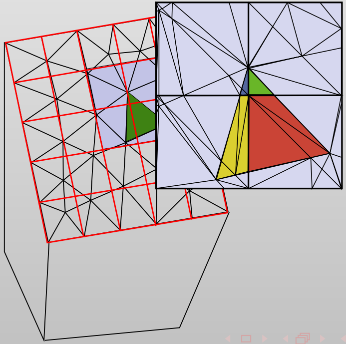
- In many applications, there exists a parametrization for the interface (e.g. planar, cylindrical, spherical, ...)
- **Idea:** Lagrange multipliers live on $d-1$ dimensional structured meshes that are transformed to the physical space.
- Overlaps between multiplier mesh and subdomain meshes are computed using `dune-grid-glue` by C. Engwer and O. Sander.



Hybrid mortar: Implementation

Implementation in DUNE

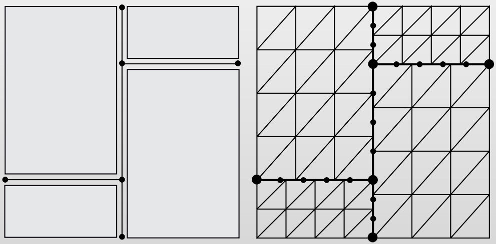
- In many applications, there exists a parametrization for the interface (e.g. planar, cylindrical, spherical, ...)
- **Idea:** Lagrange multipliers live on $d-1$ dimensional structured meshes that are transformed to the physical space.
- Overlaps between multiplier mesh and subdomain meshes are computed using `dune-grid-glue` by C. Engwer and O. Sander.



Hybrid mortar: Stokes problem

Stokes problem [Egger and W. 2010a]:

- Partitioning and triangulations:



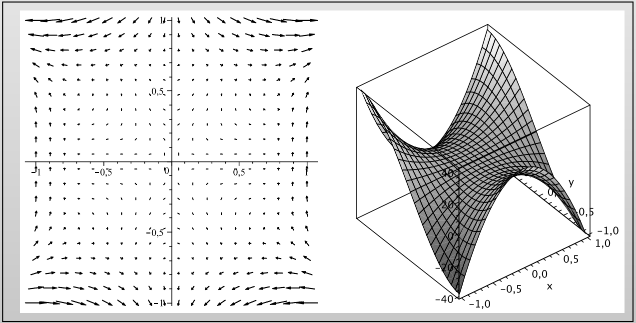
- The subdomain interfaces are extracted using a `Codim1Extractor`
- The entire `OneDGrid` is extracted with a `Codim0Extractor`
- A transformation is given to place the interface grids on the coupling boundaries.
- We use the `PSurface` backend to generate remote intersections.

Numerical example

Example: Stokes problem: Colliding flow, $\Omega = (-1,1)^2$

$$\left.\begin{array}{rcl} -\Delta\,\boldsymbol{u} + \nabla p & = & \mathbf{0} \\ \operatorname{div}\boldsymbol{u} & = & 0 \end{array}\right\} \text{ on } \Omega, \qquad \boldsymbol{u} = \boldsymbol{u}_{exact} \text{ on } \partial\Omega$$

$$\boldsymbol{u}_{exact} = \left(20xy^3, 5x^4 - 5y^4\right), \; p_{exact} = 60x^2y - 20y^3$$



Plot of analytic solution; velocity vectors and pressure field

Mortar: Numerical results (1/2)

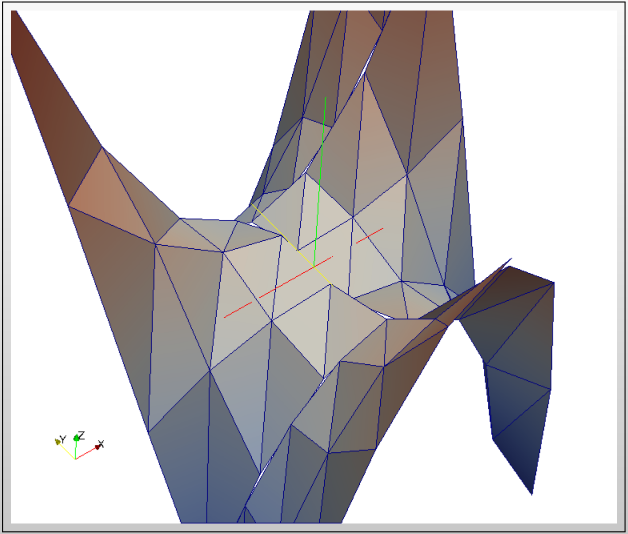


Figure: Numerical results (p=2): h: 1.00; L^2-error: 1.418; energy error: 3.253

Mortar: Numerical results (1/2)

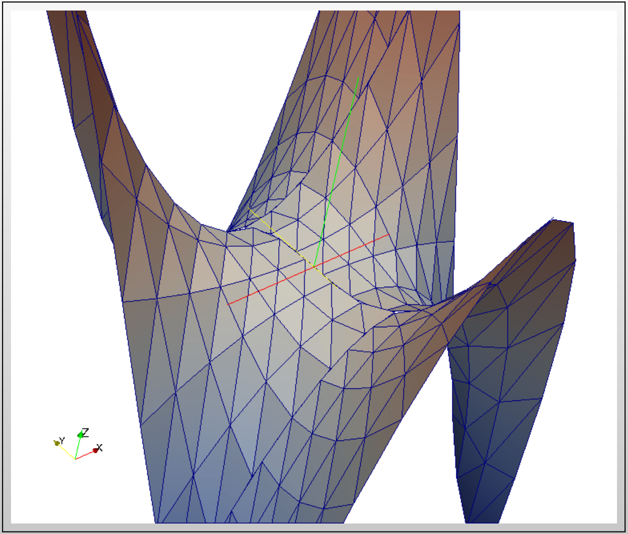


Figure: Numerical results (p=2): h: 0.50; L^2-error: 0.171; energy error: 0.761

Mortar: Numerical results (1/2)

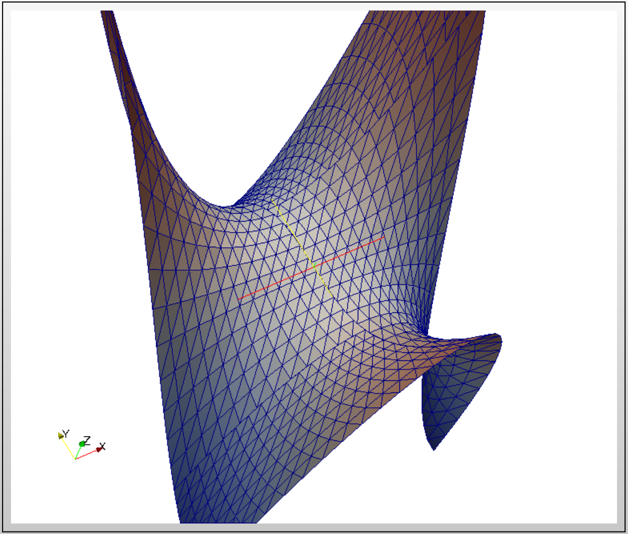


Figure: Numerical results (p=2): h: 0.25; L^2-error: 0.021; energy error: 0.183

Mortar: Numerical results (2/2)

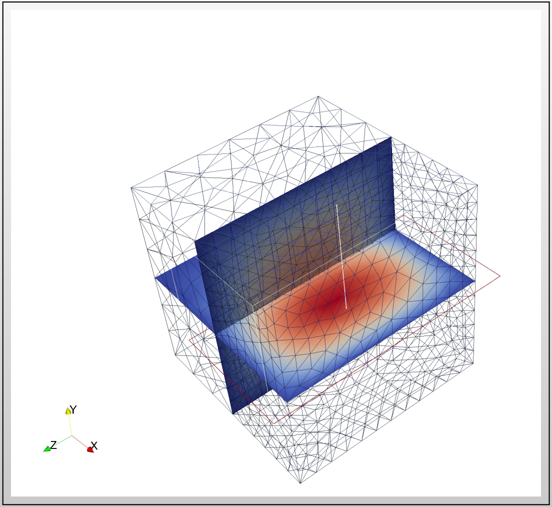


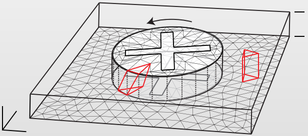
Figure: Simple example for a diffusion problem with non-matching meshes in 3D

Outline

Conclusion

Summary

- Presentation of hybridized DG methods.
- Implementation of HDG and hybridized mortar methods in DUNE.
- Possible applications include different interface problems (e.g. propellers).



Known issues

- HDG methods in DUNE not (yet) naturally implementable?

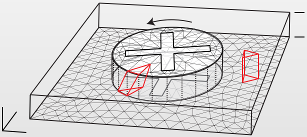
Future work

- Implementation of parallel codes for high performance computing.
- Analysis and implementation for time-dependent problems.
- Use of efficient domain decomposition solvers.

Conclusion

Summary

- Presentation of hybridized DG methods.
- Implementation of HDG and hybridized mortar methods in DUNE.
- Possible applications include different interface problems (e.g. propellers).



Known issues

- HDG methods in DUNE not (yet) naturally implementable?

Future work

- Implementation of parallel codes for high performance computing.
- Analysis and implementation for time-dependent problems.
- Use of efficient domain decomposition solvers.

Thanks for your attention!

Acknowledgement

Financial support from the
German Research Association (DFG)
through grant GSC 111
is gratefully acknowledged.

Deutsche
Forschungsgemeinschaft
DFG

Literature

D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini.
Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems.
SIAM Journal for Numerical Analysis, 39:1749–1779, 2002.

R. Becker, P. Hansbo, and R. Stenberg.
A finite element method for domain decomposition with non-matching grids.
ESAIM: Mathematical Modeling and Numerical Analysis, 37(2):209–225, 2003.

B. Cockburn, J. Gopalakrishnan, and R. Lazarov.
Unified hybridization of discontinuous Galerkin, mixed and continuous Galerkin methods for second order elliptic problems.
SIAM Journal for Numerical Analysis, 47(2):1319–1365, 2009.

H. Egger.
A class of hybrid mortar finite element methods for interface problems with non-matching meshes.
Technical report, AICES-2009-2, RWTH-Aachen University, 2009.

H. Egger, and C. Waluga.
A hybridized discontinuous Galerkin method for incompressible flow. Part I: Stokes problem.
in preparation, 2010.

H. Egger, and C. Waluga.
A hybrid mortar finite element method for the Stokes problem.
submitted, 2010.