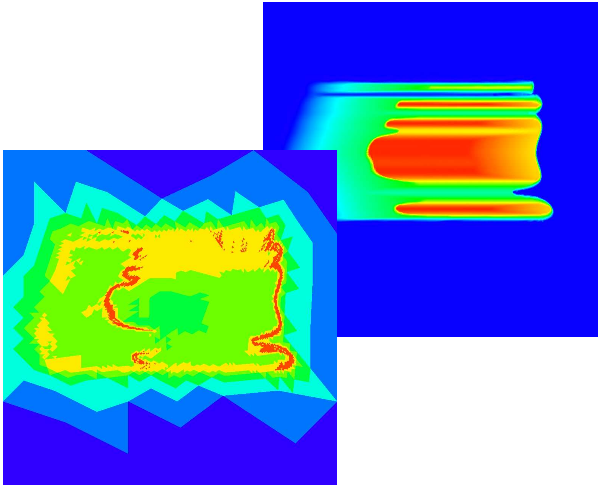
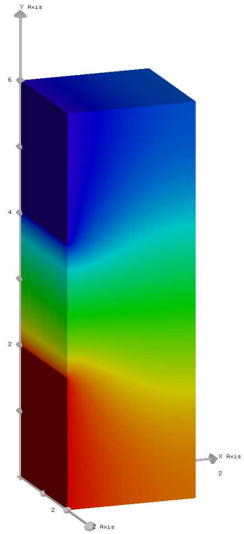
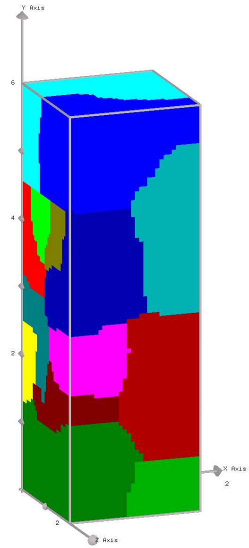
Efficient Simulation of Convection Diffusion Equations







Mario Ohlberger, Universität Münster

Computational Methods with Applications, Harrachov 2007

mario.ohlberger@uni-muesnter.de

www.uni-muenster.de/math/u/ohlberger

Outline

- Introduction
- General concept for obtaining error control
- Higher order DG for conservation laws
- DUNE – adaptive and parallel programming
- Application: Simulation of PEM fuel cells

Basic model problem: convection-diffusion equation

$$\partial_t \mathbf{u} \quad + \quad \nabla \cdot \mathbf{F}(\mathbf{u}) \quad - \quad \Delta D(\mathbf{u}) = 0.$$

accumulation convection diffusion

Special interest: Convection dominated flow ($D' \ll |\mathbf{F}'|$).

Goal: A posteriori error control and adaptivity!

Goal: A posteriori error estimates and adaptivity

Situation: $\mathbf{u}$ **exact solution,** $\mathbf{u_h}$ **approximate solution.**

Goal: A posteriori error estimates and adaptivity

Situation: **u** **exact solution,** **u_h** **approximate solution.**

First step: **A posterirori error estimate.**

$$||u - u_h||_1 \leq \eta(u_h).$$

Goal: A posteriori error estimates and adaptivity

Situation: $\mathbf{u}$ exact solution, $\mathbf{u_h}$ approximate solution.

First step: A posterirori error estimate.

$$||u - u_h||_1 \leq \eta(u_h).$$

Second step: Definition of local error indicators.

$$\eta(u_h) = \sum_j \eta_j(u_h).$$

Goal: A posteriori error estimates and adaptivity

Third step: **Equidistribution strategy.**

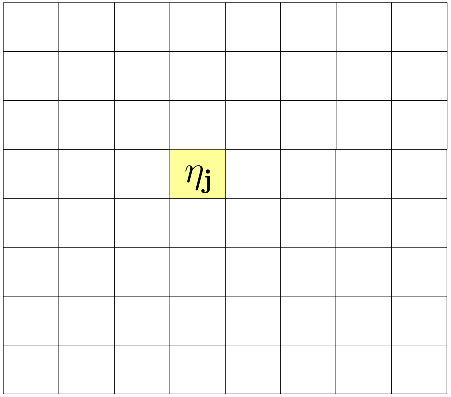
Choose local mesh size such that all $\eta_{\mathbf{j}}(\mathbf{u_h})$ are approximately of the same size, and $\eta(u_h) \leq TOL$!

Goal: A posteriori error estimates and adaptivity

Third step: **Equidistribution strategy.**

Choose local mesh size such that all $\eta_j(u_h)$ are approximately of the same size, and $\eta(u_h) \leq TOL$!

This is done by the estimate–mark–adapt algorithm:



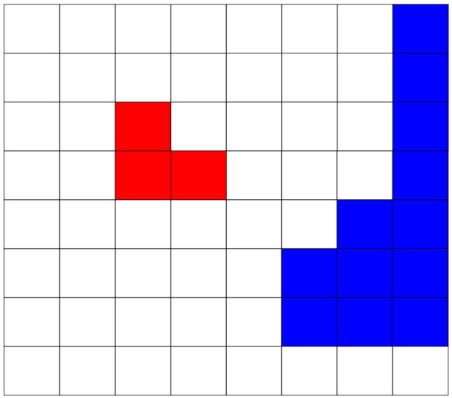
"estimate"

Goal: A posteriori error estimates and adaptivity

Third step: **Equidistribution strategy**

Choose local mesh size such that all $\eta_j(u_h)$ are approximately of the same size, and $\eta(u_h) \leq TOL$!

This is done by the estimate–mark–adapt algorithm:



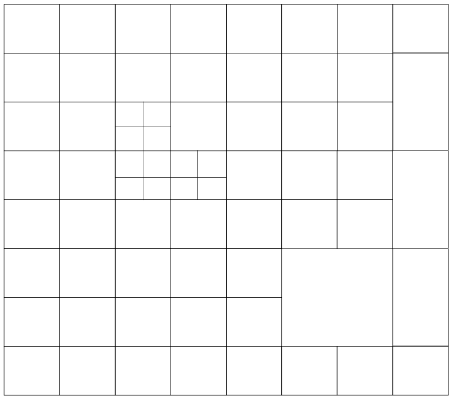
”mark”

Goal: A posteriori error estimates and adaptivity

Third step: **Equidistribution strategy**

Choose local mesh size such that all $\eta_{\mathbf{j}}(\mathbf{u_h})$ are approximately of the same size, and $\eta(u_h) \leq TOL$!

This is done by the estimate–mark–adapt algorithm:



"adapt"

General concept for obtaining error control ...

(The hyperbolic case ($D \equiv 0$)!)

Analytical framework: Entropy weak solution

u is called an entropy weak solution of the conservation law, if u satisfies for all entropy pairs (S, F_S), and for all $\phi \in C_0^1(\mathbb{R}^d \times \mathbb{R}^+, \mathbb{R}^+)$:

$$\int\limits_{\mathbb{R}^d}\int\limits_{\mathbb{R}^+} \left(S(u)\partial_t\phi + F_S(u)\cdot\nabla\phi\right)\,dt\,dx + \int\limits_{\mathbb{R}^d} S(u_0)\phi(x,0)\,dx \geq 0.$$

Recall that (S, F_S) is called an entropy - entropy flux pair, iff S is convex and

$$F_S' = S'f'$$

.

General concept for obtaining error control

Entropy Residual $R_S(v)$ for any given approximation v:

$$\langle R_S(v), \phi \rangle := \int_{\mathbb{R}^2 \times R^+} S(v)\partial_t \phi + F_S(v) \cdot \nabla \phi + \int_{\mathbb{R}^2} S(u_0)\phi(\cdot, 0).$$

Fundamental error estimate:

[Eymard, Gallouët, Ghilani, Herbin '98], [Chainais-Hillairet '99]

Let $S(u) := |u - \kappa|$ be the Kruzkov entropy. Suppose that for v there exist measures $\mu_v \in \mathcal{M}(\mathbb{R}^d \times \mathbb{R}^+)$ and $\nu_v \in \mathcal{M}(\mathbb{R}^d)$ independent of κ such that

$$\langle R_S(v), \phi \rangle \geq -(\langle |\partial_t \phi| + |\nabla \phi|, \mu_v \rangle + \langle |\phi(\cdot, 0)|, \nu_v \rangle).$$

General concept for obtaining error control

Entropy Residual $R_S(v)$ for any given approximation v:

$$\langle R_S(v), \phi \rangle := \int_{\mathbb{R}^2 \times R^+} S(v)\partial_t \phi + F_S(v) \cdot \nabla \phi + \int_{\mathbb{R}^2} S(u_0)\phi(\cdot, 0).$$

Fundamental error estimate:

[Eymard, Gallouët, Ghilani, Herbin '98], [Chainais-Hillairet '99]

Let $S(u) := |u - \kappa|$ be the Kruzkov entropy. Suppose that for v there exist measures $\mu_v \in \mathcal{M}(\mathbb{R}^d \times \mathbb{R}^+)$ and $\nu_v \in \mathcal{M}(\mathbb{R}^d)$ independent of κ such that

$$\langle R_S(v), \phi \rangle \geq -(\langle |\partial_t \phi| + |\nabla \phi|, \mu_v \rangle + \langle |\phi(\cdot, 0)|, \nu_v \rangle).$$

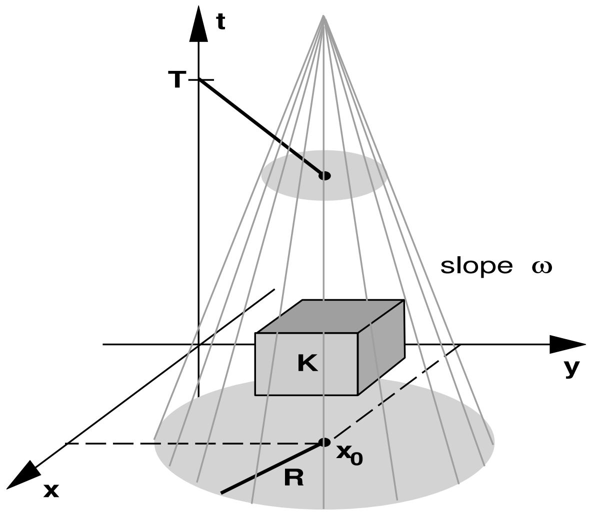
Then the following error estimate holds:

$$||u - v||_{L^1(K)} \leq T(\nu_v(B_{R+\delta}(x_0)) + C_1 \mu_v(D_\delta) + C_2 \sqrt{\mu_v(D_\delta)}).$$

General concept for obtaining error control

$$||u-v||_{L^1(K)} \leq T(\nu_v(B_{R+\delta}(x_0)) + C_1\mu_v(D_\delta) + C_2\sqrt{\mu_v(D_\delta)}).$$

Cone of dependence D_δ:



for details see also
[Kröner, Ohlberger '00]

A posteriori results in this context

1) Hyperbolic conservation laws:

1995 Cockburn, Gau: 2000 Kröner, Ohlberger:	Finite volume schemes;
2000 Gosse, Makridakis:	Relaxation schemes;
2003 Küther, Ohlberger:	Central staggered schemes;
2006 Ohlberger, Vovelle:	Boundary value problems;
2007 Dedner, Makridakis, Ohlberger:	Discontinuous Galerkin;

A posteriori results in this context

1) Hyperbolic conservation laws:

1995 Cockburn, Gau:	Finite volume schemes;
2000 Kröner, Ohlberger:	
2000 Gosse, Makridakis:	Relaxation schemes;
2003 Küther, Ohlberger:	Central staggered schemes;
2006 Ohlberger, Vovelle:	Boundary value problems;
2007 Dedner, Makridakis, Ohlberger:	Discontinuous Galerkin;

2) Degenerate parabolic problems:

2001 Ohlberger:	Vertex centered FV;
2002 Ohlberger, Rohde:	Weakly coupled systems;
2002 Herbin, Ohlberger:	Cell centered FV
2004 Ohlberger:	Higher order FV;
2004 Chen, Ji:	Finite element schemes;

A posteriori results in this context

1) Hyperbolic conservation laws:

1995 Cockburn, Gau: 2000 Kröner, Ohlberger:	Finite volume schemes;
2000 Gosse, Makridakis:	Relaxation schemes;
2003 Küther, Ohlberger:	Central staggered scheme;
2006 Ohlberger, Vovelle:	Boundary value problems;
2007 Dedner, Makridakis, Ohlberger:	Discontinuous Galerkin;

2) Degenerate parabolic problems:

2001 Ohlberger:	Vertex centered FV;
2002 Ohlberger, Rohde:	Weakly coupled systems;
2002 Herbin, Ohlberger:	Cell centered FV
2004 Ohlberger:	Higher order FV;
2004 Chen, Ji:	Finite element schemes;

Error control for Discontinuous Galerkin approximations of nonlinear conservation laws

[Dedner, Makridakis, Ohlberger '07]

DG for nonlinear conservation laws

$$\begin{aligned} \partial_t u + \nabla \cdot f(u) &= 0 \qquad \text{in } \mathbb{R}^d \times \mathbb{R}^+ \ , \\ u(\cdot, 0) &= u_0 \qquad \text{in } \mathbb{R}^d. \end{aligned}$$

Sought: $u \in BV(\mathbb{R}^d \times \mathbb{R}^+)$

Given: Nonlinear flux: $f \in C^1(\mathbb{R})$

Initial data: $u_0 \in BV(\mathbb{R}^d)$

Notation (fixed mesh for all times)

Decomposition of R^d: $\mathcal{T}$ with control volumes $T_j \in \mathcal{T}, j \in J$ and faces $S_{jl} = \bar{T}_j \cap \bar{T}_l$.

DG approximation space: $V_h^p := \{v_h \in BV(\mathbb{R}^d) | \, v_h|_{T_j} \in \mathbb{P}_p \; \forall j \in J\}$.

Notation (fixed mesh for all times)

Decomposition of R^d: $\mathcal{T}$ with control volumes $T_j \in \mathcal{T}, j \in J$ and faces $S_{jl} = \bar{T}_j \cap \bar{T}_l$.

DG approximation space: $V_h^p := \{v_h \in BV(\mathbb{R}^d) | \, v_h|_{T_j} \in \mathbb{P}_p \ \forall j \in J\}$.

Semi-discrete DG approximation without stabilization

$$\frac{d}{dt}(u_j(t), v_j)_{T_j} - (f(u_j(t)), \nabla v_j)_{T_j} + \sum_{l \in N(j)} (f_{jl}(u_j(t), u_l(t)), v_j)_{S_{jl}} = 0,$$

$$\text{for all } v_j \in \mathbb{P}_p, T_j \in \mathcal{T}.$$

Notation (adaptive meshes)

Decomposition of $[0,T]$: $\mathcal{I} = \{t^0, ..., t^N\}$, $I^n := (t^n, t^{n+1}]$, $\Delta t^n := |I^n|$.

Decomposition of R^d in I^n : $\mathcal{T}^n$ with control volumes $T_j \in \mathcal{T}^n, j \in J^n$.

DG space in I^n: $V^p_{h,n} := \{v_h \in BV(\mathbb{R}^d) |\, v_h|_{T_j} \in \mathbb{P}_p \;\forall j \in J^n\}$.

Notation (adaptive meshes)

Decomposition of $[0, T]$: $\mathcal{I} = \{t^0, ..., t^N\}$, $I^n := (t^n, t^{n+1}]$, $\Delta t^n := |I^n|$.

Decomposition of R^d in I^n : $\mathcal{T}^n$ with control volumes $T_j \in \mathcal{T}^n, j \in J^n$.

DG space in I^n: $V_{h,n}^p := \{v_h \in BV(\mathbb{R}^d) | \, v_h|_{T_j} \in \mathbb{P}_p \; \forall j \in J^n\}$.

Limiting projection operators on I^n

For $t \in I^n$ let $\mathbf{\Lambda_h^{n,t}} : V_{h,n}^p \to V_{h,n}^p$ and $\mathbf{\Lambda_h^{n,t^n}} : V_{h,n-1}^p \to V_{h,n}^p$
be two projection operators that are mass conservative.

For $u_h \in V_{h,n}^p$ define the projected approximation $\widetilde{u}_h$ through:

$$\widetilde{\mathbf{u}}_\mathbf{h}(t) = \mathbf{\Lambda_h^{n,t}}(u_h(t)) \quad \text{for } t \in I^n, \quad n = 0, \dots, N-1 \ .$$

Semi-discrete DG approximation on adaptive meshes

Set $u_h^{-1} := \Pi_{V_{h,0}^p}(u_0)$.

For $n = 0, ..., N-1$, $u_h^n \in C^1(I^n; V_{h,n}^p)$ is defined through

$$u_h^n(t^n) \;:=\; \mathbf{\Lambda_h^{n,t^n}}(u_h^{n-1}(t^n))\,,$$

$$\frac{\mathrm{d}}{\mathrm{dt}}(u_j^n(t), v_j)_{T_j} - (f(\tilde{\mathbf{u}}_{\mathbf{j}}^{\mathbf{n}}(t)), \nabla v_j)_{T_j} + \sum_{l\in N(j)} (f_{jl}(\tilde{\mathbf{u}}_{\mathbf{j}}^{\mathbf{n}}(t), \tilde{\mathbf{u}}_{\mathbf{l}}^{\mathbf{n}}(t)), v_j)_{S_{jl}} = 0$$

$$\text{for all} \quad v_j \in \mathbb{P}_p, j \in J^n, t \in I^n\,.$$

The global approximation $u_h \in L^\infty(0,T; V_{h,n}^p)$ is defined through $u_h|_{I^n} := u_h^n$.

A posteriori error estimate [Dedner, Makridakis, Ohlberger '05]

$$||(u - u_h)(T)||_{L^1(B_R(x_0))} \;\le\; ||(\widetilde{u}_h - u_h)(T)||_{L^1(B_R(x_0))} + \eta_0 + \sqrt{K_1 \eta_1} + \sqrt{K_2 \eta_2}$$

with

$$\eta_0 = \sum_{j \in J^0} \int_{T_j} |u_0 - \widetilde{u}_j^0(0)|,$$

$$\eta_1 = \sum_n \sum_{j \in J^n} \Big[\int_{t^n}^{t^{n+1}} \quad \mathbf{h_j} \quad \mathbf{R^n_{T,j}} + \frac{1}{2} \int_{t^n}^{t^{n+1}} \sum_{l \in N(j)} \quad \mathbf{h_{jl}} \quad \mathbf{R^n_{S,jl}} + \quad \mathbf{h_j} \quad \mathbf{R^n_{\Lambda,j}} \Big],$$

$$\eta_2 = \sum_n \sum_{j \in J^n} \Big[\int_{t^n}^{t^{n+1}} ||\overline{\widetilde{\mathbf{u}}^{\mathbf{n}}_{\mathbf{j}}} - \widetilde{\mathbf{u}}^{\mathbf{n}}_{\mathbf{j}}||_\infty \mathbf{R^n_{T,j}} + \frac{1}{2} \int_{t^n}^{t^{n+1}} \sum_{l \in N(j)} \max_{\mathbf{k} \in \{\mathbf{j},\mathbf{l}\}} ||\overline{\widetilde{\mathbf{u}}^{\mathbf{n}}_{\mathbf{k}}} - \widetilde{\mathbf{u}}^{\mathbf{n}}_{\mathbf{k}}||_\infty \mathbf{R^n_{S,jl}} + ||\overline{\widetilde{\mathbf{u}}^{\mathbf{n-1}}}(\mathbf{t^n}) - \widetilde{\mathbf{u}}^{\mathbf{n-1}}(\mathbf{t^n})||_\infty \mathbf{R^n_{\Lambda,j}} \Big],$$

$$\mathbf{R^n_{T,j}} = \int_{T_j} \Big| \partial_t \widetilde{u}_j + \nabla \cdot f(\widetilde{u}_j) \Big|, \qquad \mathbf{R^n_{S,jl}} = \int_{S_{jl}} Q_{jl} |\widetilde{u}_j - \widetilde{u}_l|, \qquad \mathbf{R^n_{\Lambda,j}} = \int_{T_j} |\widetilde{u}^n(t^n) - \widetilde{u}^{n-1}(t^n)|$$

element residual — jump residual — projection residual

Choice of the projection operators

Condition from the a posteriori error estimate:

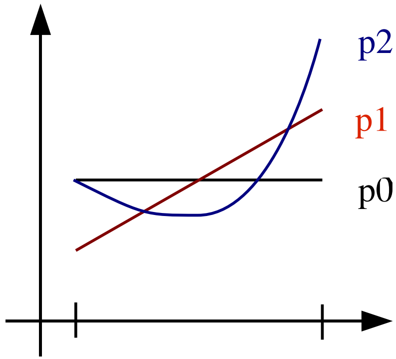
Construct projection operator, such that

$$||\overline{\widetilde{u}_j^n(\cdot,t)} - \widetilde{u}_j^n(\cdot,t)||_{L^\infty(T_j)} \leq \lambda_j^n(t).$$

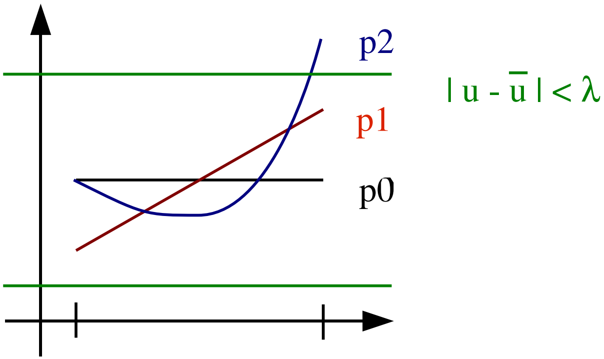
with

$$\lambda_j^n(t) \leq \begin{cases} h_j & \text{near discontinuities,} \\ \text{large} & \text{in smooth regions.} \end{cases}$$

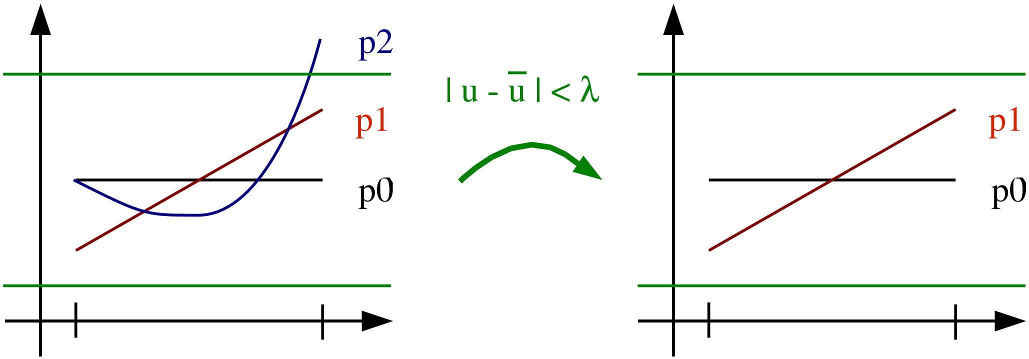
P–adaptive method:



P–adaptive method:



P–adaptive method:



Numerical experiment .....

Buckley–Leverett problem

$$\begin{aligned} u_t + \partial_x f(u) &= 0, \quad \text{on } (-1,1) \times (0, 0.4), \\ u(\cdot, 0) &= u_0, \quad \text{on } (-1,1), \end{aligned}$$

with

$$f(u) = \frac{u^2}{u^2 + \frac{1}{2}(1-u)^2}$$

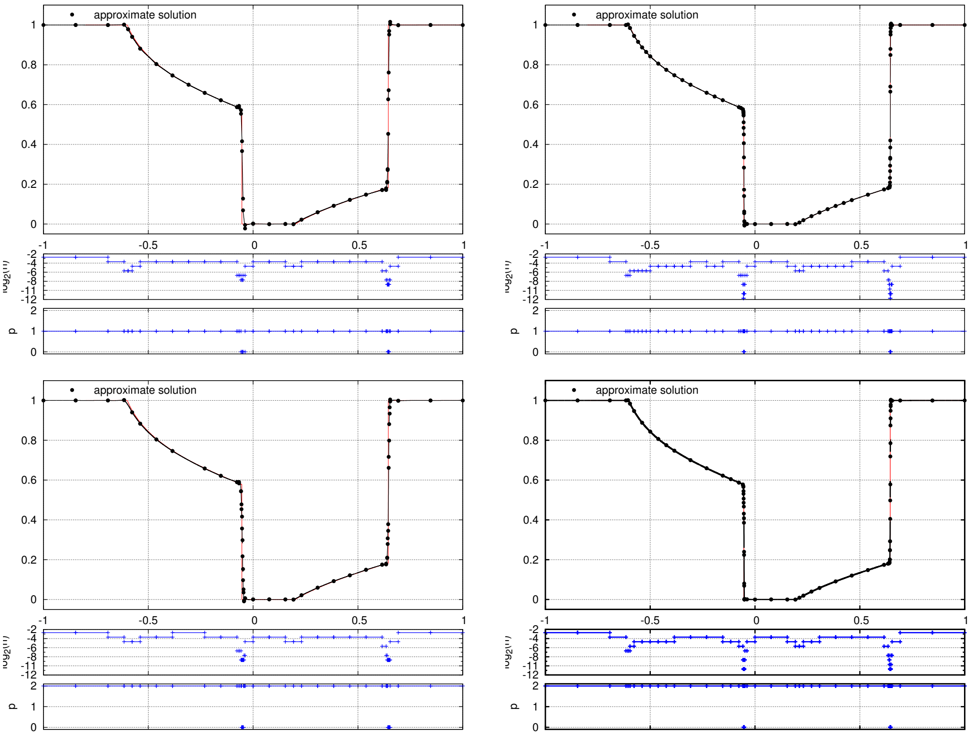
and initial data

$$u_0(x) := \begin{cases} 1, & \text{for } x < -0.6, \\ 0, & \text{for } -0.6 \le x < 0.2, \\ 1, & \text{for } 0.2 \le x. \end{cases}$$

TOL = 0.25 TOL = 0.125

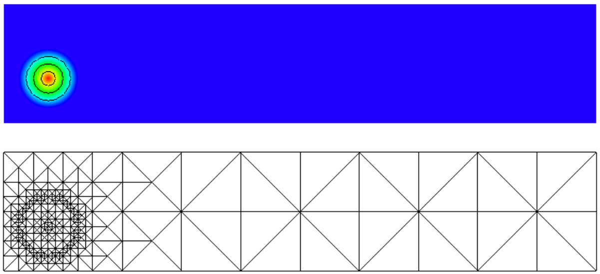
p=1

p=2

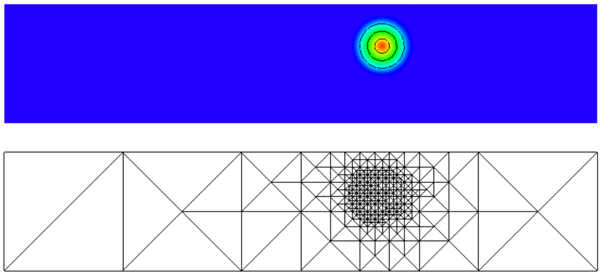


Results in 2D: Linear advection
[Dedner, Ohlberger '06]

$$\begin{aligned} \partial_t c + \nabla \cdot (\mathbf{b} c) &= 0, \\ c(\cdot, 0) &= c_0. \end{aligned}$$

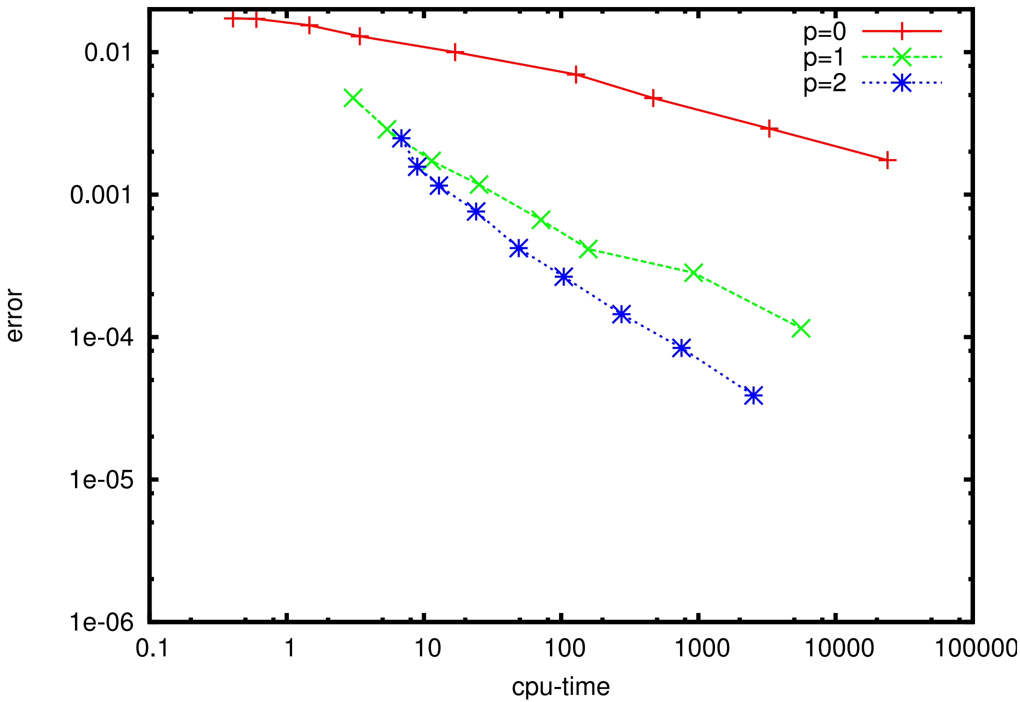


$\mathbf{t = 0}$



$\mathbf{t = T}$

Results in 2D: Linear advection

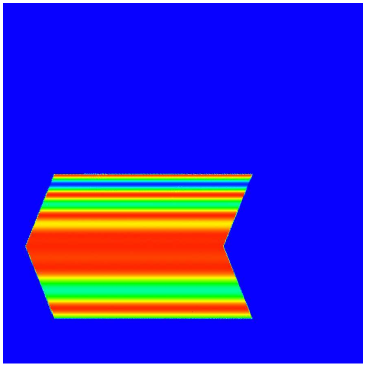


Results in 2D: Buckley–Leverett with advection

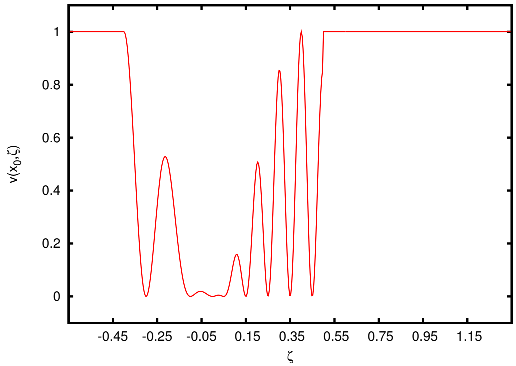
$$\begin{aligned} \partial_t u + \partial_x f(u) + a\partial_y u &= 0, \\ u(\cdot, 0) &= u_0, \end{aligned}$$

with $f(u) = \frac{u^2}{u^2+0.5(1-u)^2}$, $a = 1.3$.

Initial data:



c_0



$c_0(x = 0, \cdot)$

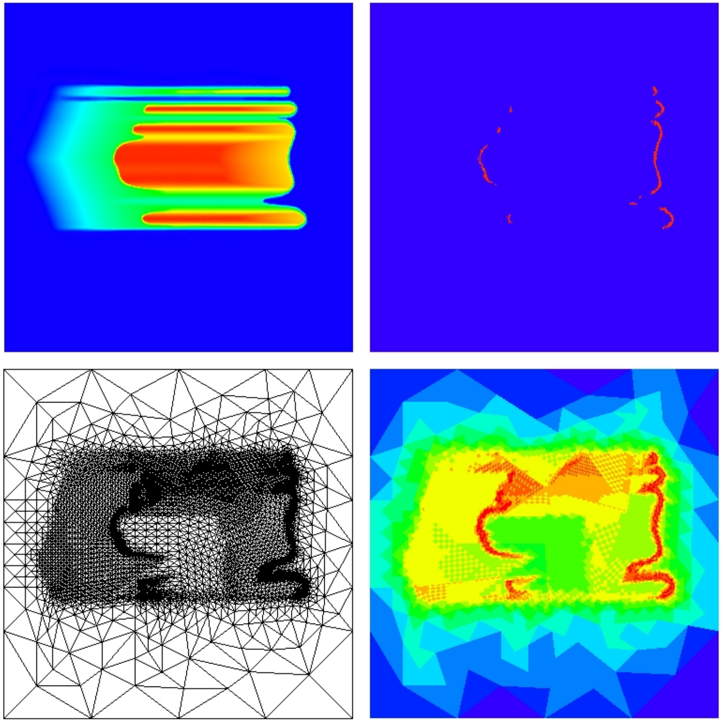
Results in 2D: Buckley–Leverett with advection

solution u_h at $t = 0.35$

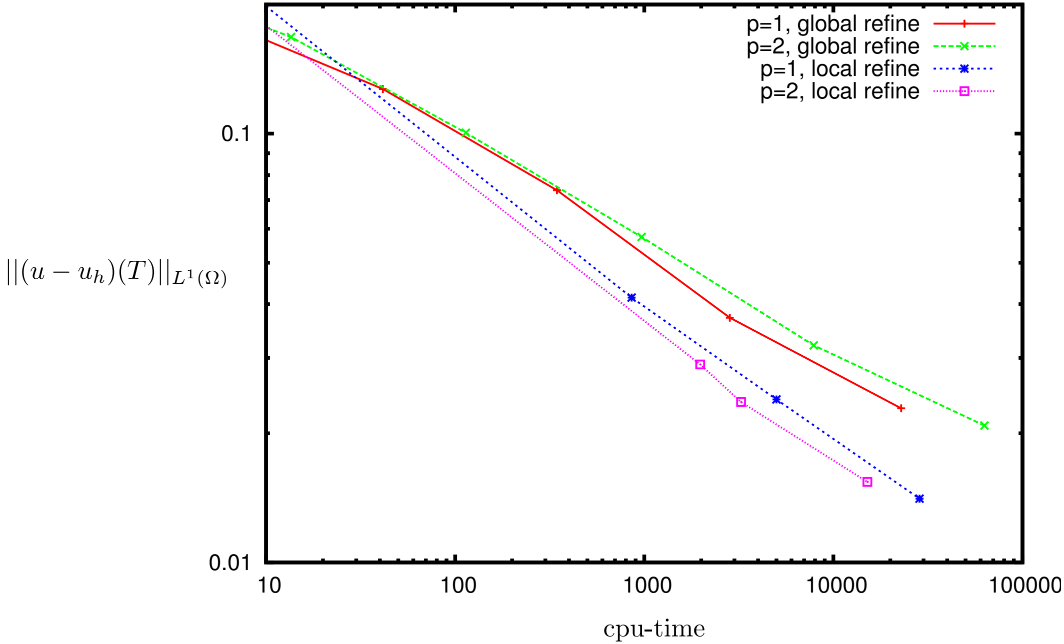
polynomial degree

grid

level of refinement

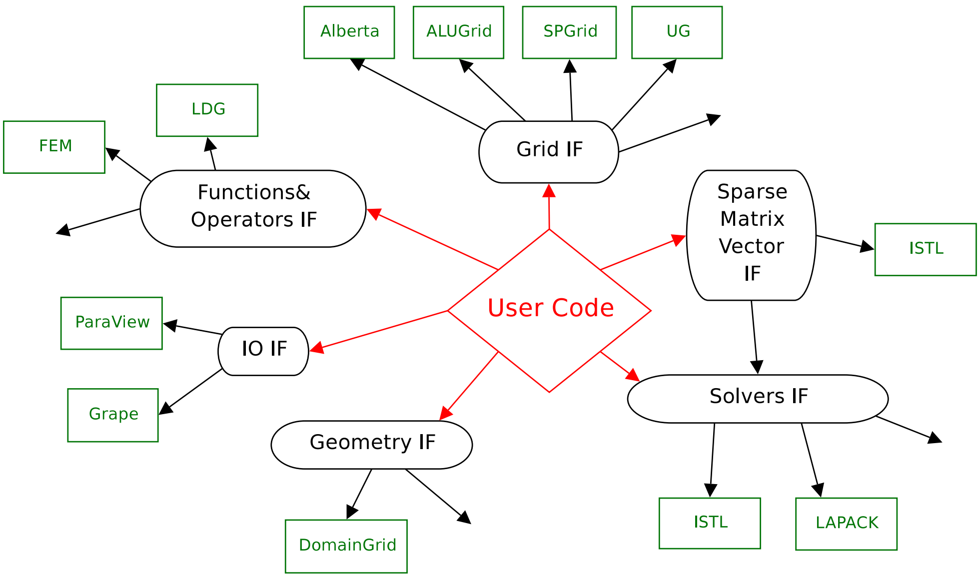


Results in 2D: Buckley–Leverett with advection



DUNE–Distributed and Unified Numerics Environment

- **Interface concept**



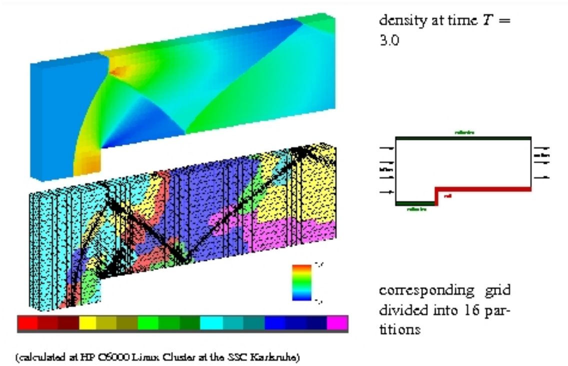
$\Longrightarrow$ **efficient reuse of existing software**

DUNE–Distributed and Unified Numerics Environment

- **Design principles**
  - Modularity through use of interfaces
  - Dimension and data structure independend programming
  - Efficiency through static polymorphism
- **Developer**
  - P. Bastian, M. Blatt, C. Engwer (Stuttgart), O. Sander (Berlin)
  - A. Dedner, R. Klöfkorn (Freiburg), M. Ohlberger (Münster)
- **Availability**
  - Homepage http://www.dune-project.org/
  - Programming language C++
  - Portability via ISO standard conformmity
  - Open source software 'GNU with linking exceptions'

DUNE – Parallel efficiency test with loaad balancing

- **Benchmark: 3D Forward Facing Step** [Burri, Dedner, Klöfkorn, Ohlberger '05]



- **Speedup and efficiency comparison**

original code

K	CPU time	$S_{4\to K}$	$E_{4\to K}$
4	0.0089062		
8	0.0046045	1.93424	0.96712
16	0.0023943	3.71978	0.92995
32	0.0012710	7.00712	0.87589

DUNE

K	CPU time	$S_{4\to K}$	$E_{4\to K}$
4	0.0101473		
8	0.0052195	1.94411	0.97205
16	0.0026859	3.77795	0.94449
32	0.0013971	7.26325	0.90791

Application: Simulation of PEM - fuel cells

Joint BMBF–project with:

- Robert Klöfkorn, Dietmar Kröner, AAM, Freiburg
- Jürgen Schumacher, Fraunhofer ISE, Freiburg
- Willy Jäger, Heidelberg
- Ben Schweizer, Basel
- Proton Motor Fuel Cell GmbH, Starnberg
- Freudenberg FCCT OHG, Weinheim

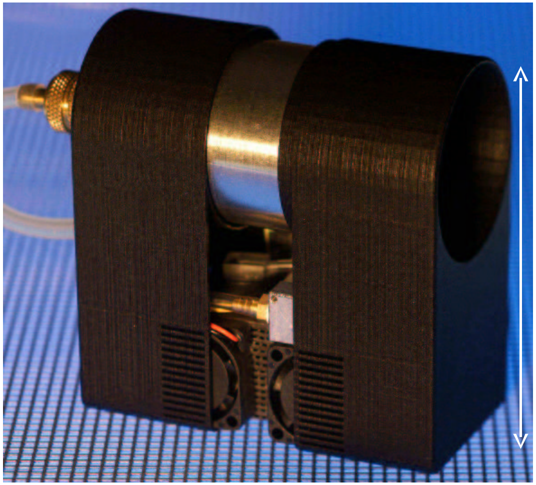
PEM fuel cells for small electronic devices

Prototype Fuel Cell System
Powering a Camcorder

9 W max. power

Same size and same energy than largest rechargeable battery pack

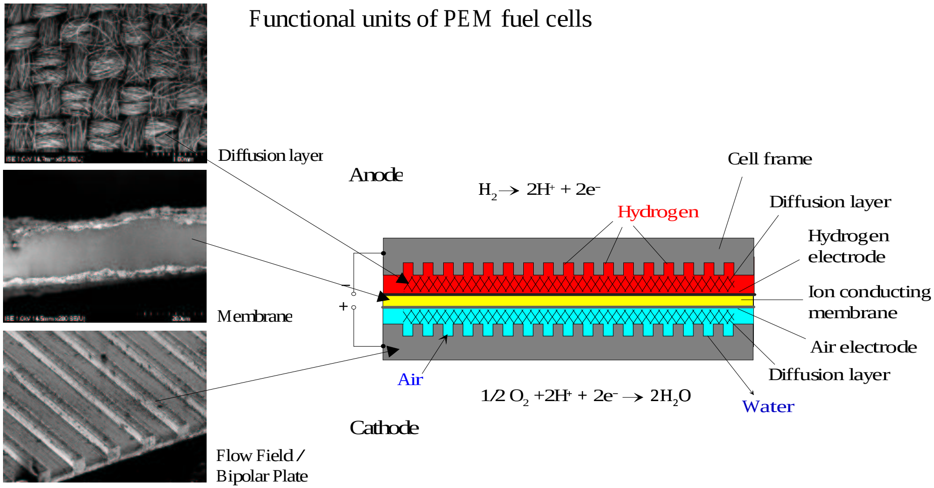
Increase of power and energy density by miniaturization of functional units



7 cm

Fraunhofer Institut Solare Energiesysteme

PEM fuel cell components



Fraunhofer Institut Solare Energiesysteme

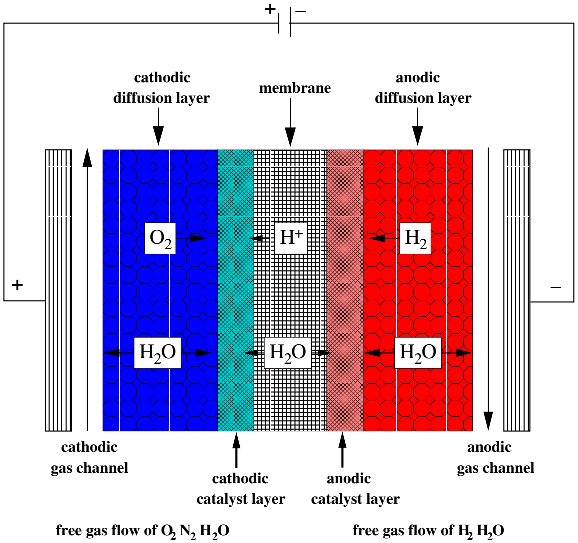
Sketch of a fuel cell

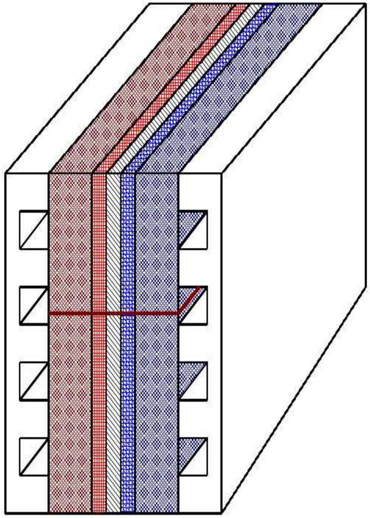
cathodic catalyst layer:

$$O_2 + 4e^- + 4H^+ \;\rightarrow\; 2H_2O$$

anodic catalyst layer:

$$2H_2 \;\rightarrow\; 4H^+ + 4e^-$$





Modeling concept:

Porous layers:

- Two phase - multi component flow in porous media with phase transition
- Potential flow of electrons and protons

Gas channels:

- Free multi component gas flow

Coupling through interface conditions

For details see:

- [Kühn, Ohlberger, Schumacher, Ziegler, Klöfkorn '03]
- [Steinkamp, Schumacher, Goldsmith, Ohlberger, Ziegler '07]

Reduced model problem in three space dimensions

[Klöfkorn, Kröner, Ohlberger '07]

Two phase flow in global pressure formulation (solve for p and $\mathbf{u}$)

$$\begin{aligned} -\nabla \cdot (K\lambda(s_w)\nabla p) &= 0, \\ \mathbf{u} &= -K\lambda(s_w)\nabla p, \end{aligned}$$

Balance equation for liquid water saturation (solve for s_w)

$$\partial_t(ns_w) + \nabla \cdot (f_w(s_w)(\mathbf{u} + \lambda_g(s_w)K\nabla p_c(s_w))) = r_{\text{phase}}.$$

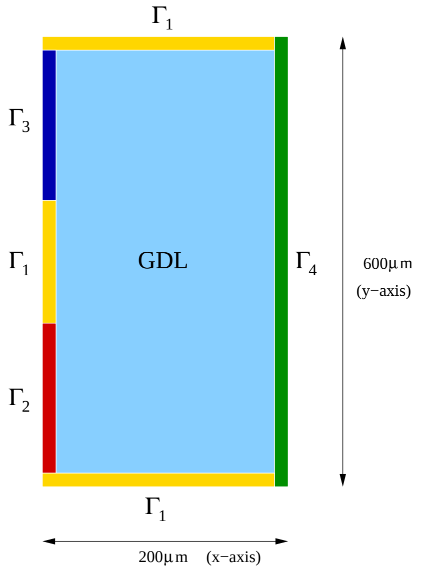
Transport of species in the gas phase (solve for $\mathbf{c} = (c_{H_2O}, c_{O_2})^T$)

$$\partial_t(ns_g\mathbf{c}) + \nabla \cdot (\mathbf{v}_g\mathbf{c}) - \nabla \cdot (ns_gD_g\nabla\mathbf{c}) = q_g.$$

Initial values and boundary conditions

Initial Values $s_w(.,0) = 0.1, \quad c_{O_2}(.,0) = 0.8, \quad c_{H_2O}(.,0) = 0.2$

Boundary Values



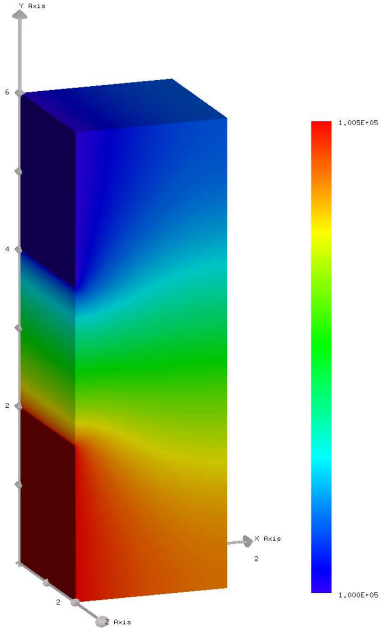
Γ_1 no flow

Γ_2 $p_w = 100500$ Pa, $s_w = 0$, $c_{O_2} = 0.8$, $c_{H_2O} = 0.2$

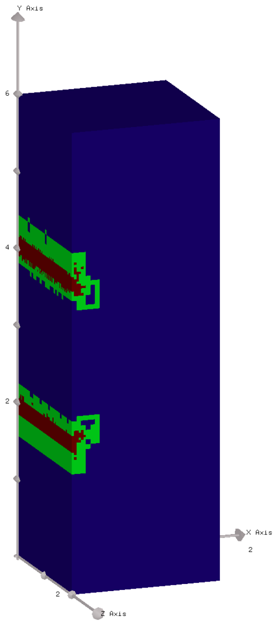
Γ_3 $p_w = 100000$ Pa, $s_w = 0$, $c_{O_2} = 0.8$, $c_{H_2O} = 0.2$

Γ_4 p_w no flow, $s_w = 1$, c_{O_2}, c_{H_2O} no flow

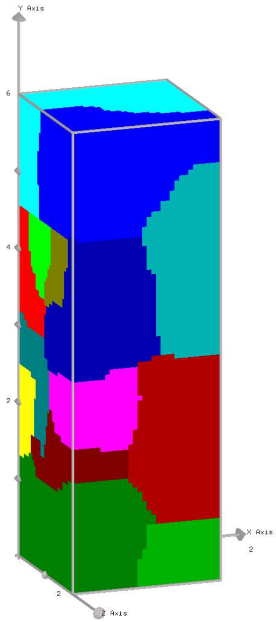
Numerical results
Pressure, adaptation level and partitioning



p_w

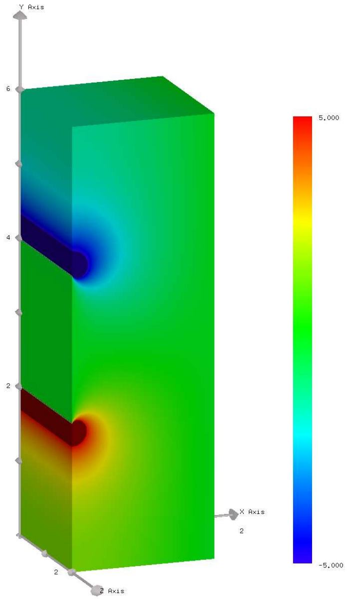


grid level

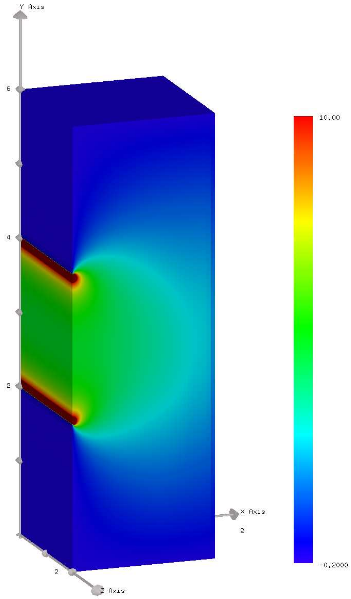


partitions

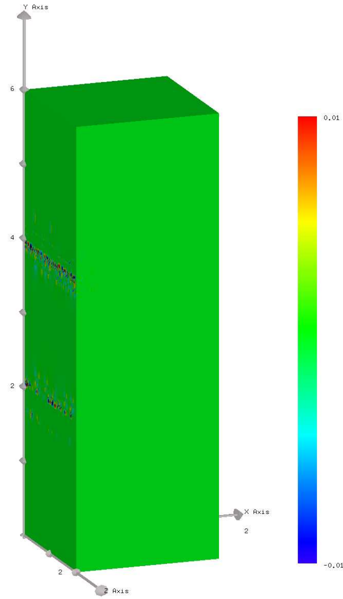
Velocity components



$\mathbf{u}_x$

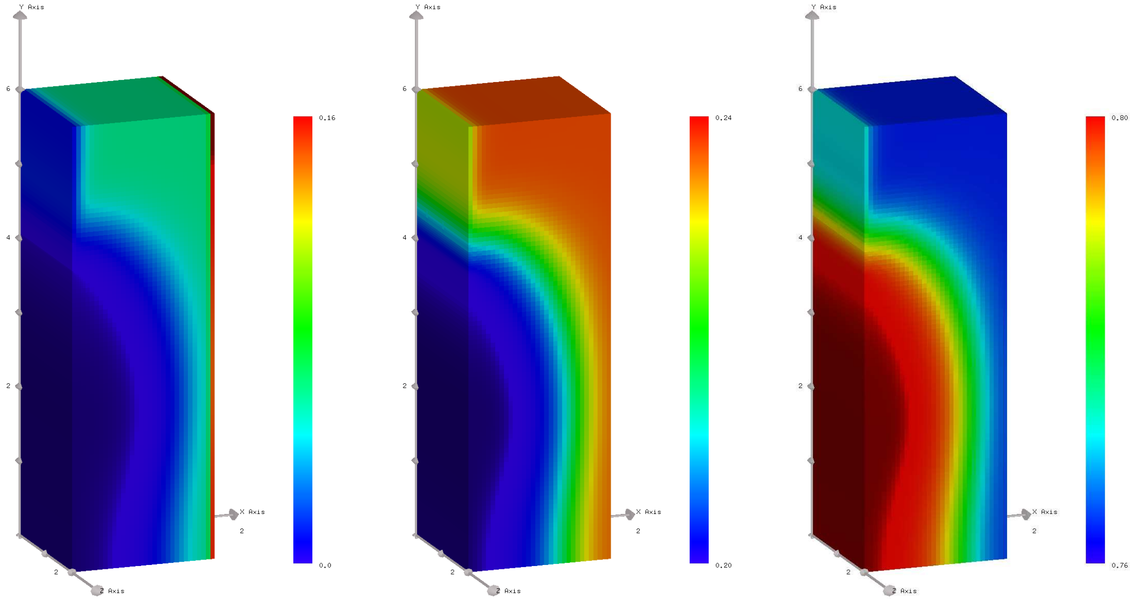


$\mathbf{u}_y$



$\mathbf{u}_z$

Saturation, and mass concentrations



s_w c_{H_2O} c_{O_2}

Complexity of the DG discretization

- **Complexity per time step:**
  - Number of elements: $\sim$ 150.000
  - Degrees of freedom per element: 43
  - Degrees of freedom per time step: $\sim$ 6.450.000
  - Computational time per time step: $\sim$ 22 seconds
  - Linear solver: preconditioned BiCG-stab from the dune-istl library [Blatt, Bastian '06]

- **Overall complexity:**
  - Number of time steps: $\sim$ 15.000
  - Computational time: $\sim$ 3 days on 16 processors (XC4000 linux cluster)
  - Time evolution with first order ODE solver

Thank you for your attention!

www.uni-muenster.de/math/u/ohlberger

Software:

DUNE	`www.dune-project.org`
ALUGrid	`www.mathematik.uni-freiburg.de/IAM/Research/alugrid`
GRAPE	`www.iam.uni-bonn.de/grape`