AMDiS

A user friendly general finite element toolbox for HPC

Thomas Witkowski
TU Dresden - Institute of Scientific Computing

6th December 2010

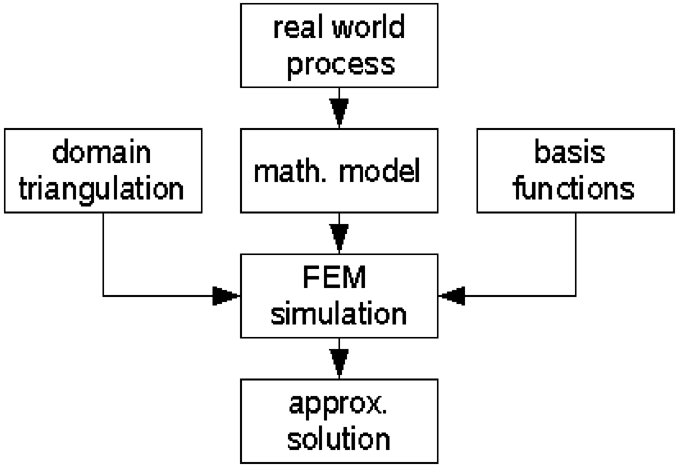
1. AMDiS

2. Parallelization
3. Applications

What is AMDiS?

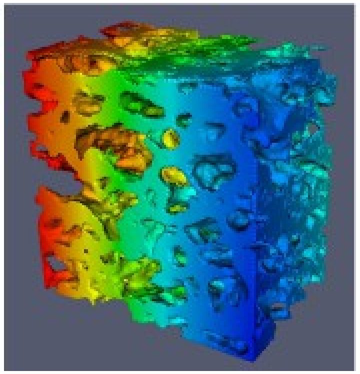
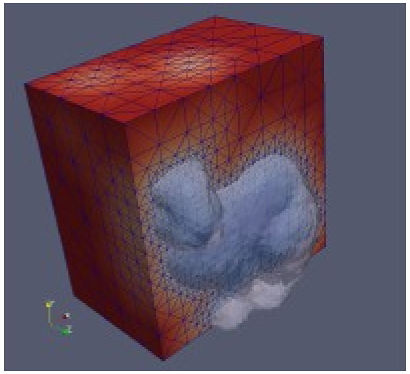
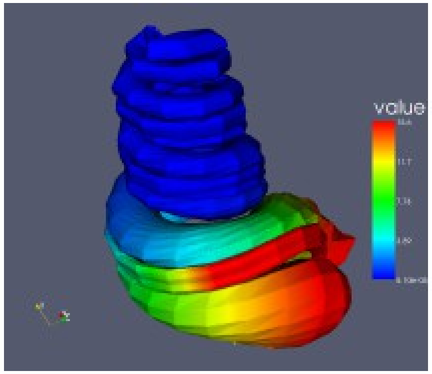
AMDiS (Adaptive MultiDimensional Simulations)

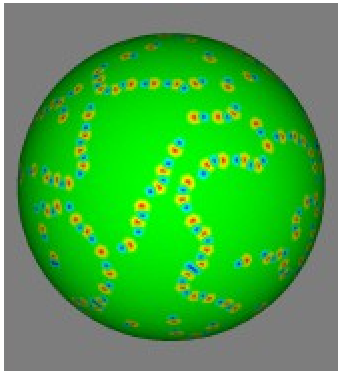
- C++ library for the numerical solution of systems of PDEs.
- Implements the finite element method (FEM).
- Design goals:
  - High level of abstraction
  - Generality
  - Extensibility
- Dimensenless problem definition: enables computation in 1D, 2D and 3D
- Hide parallelization for the user



AMDiS Features

- Solve general systems of linear 2nd order elliptic PDEs
- For time dependent equations: discretize time first and solve sequence of elliptic PDEs
- Nonlinear equations must be linearized
- Adaptivity in space and time
- Arbitrary basis functions possible
- Lagrange functions up to 4th order implemented
- Mixed finite element
- Usage of different meshes for different solution variables possible
- Arbitrary direct and iterative solvers possible
- Parallelization using nonoverlapping domain decomposition



AMDiS: Solver and performance

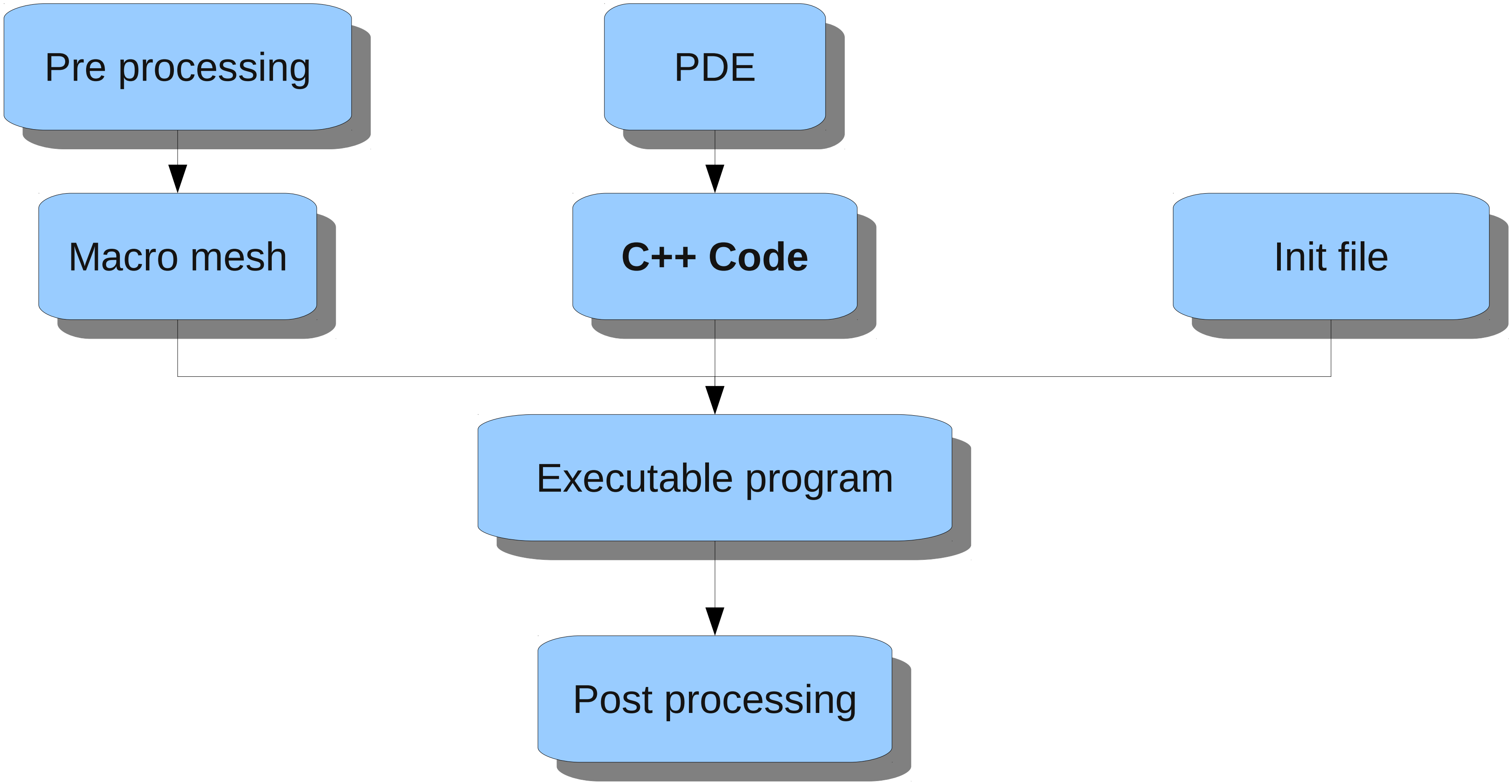
Embedded solvers:

- Iterative solvers: MTL4 (Matrix Template Library)
- Direct solvers: UMFPACK, Intel Pardiso (Part of Intel MKL 10.0)
- Parallel solvers: Petsc

Performance: How many DOFs can be solved per second?

- Assembler, solver, error estimator and adaption
- Office PC, no parallelization
- 2D: 10.000 - 25.000 DOFs
- 3D: 5.000 DOFs

Working with AMDiS



Working with AMDiS: Laplace equation

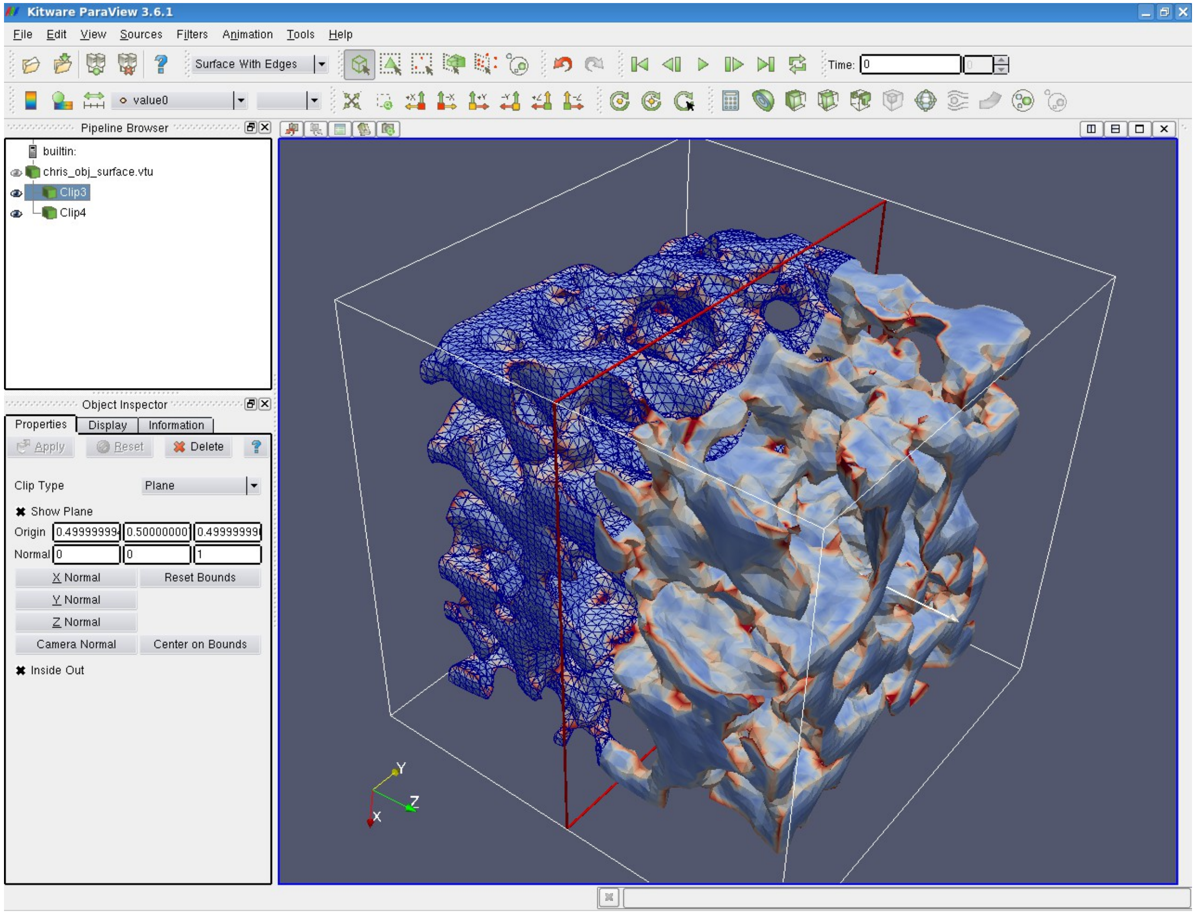
$$\Delta u = f \quad auf \quad \Omega$$
$$u = g \quad auf \quad \partial\Omega$$

```
#include "AMDiS.h"
using namespace AMDiS;

class G : public AbstractFunction<double, WorldVector<double> > { ... };
class F : public AbstractFunction<double, WorldVector<double> > { ... };

int main(int argc, char* argv[])
{
  ProblemScal ellipt("ellipt");
  ellipt.initialize(INIT_ALL);

  AdaptInfo adaptInfo("ellipt->adapt");
  AdaptStationary adapt("ellipt->adapt", ellipt, adaptInfo);

  Operator matrixOperator(ellipt.getFeSpace());
  matrixOperator.addSecondOrderTerm(new Laplace_SOT);
  ellipt.addMatrixOperator(&matrixOperator);

  Operator rhsOperator(ellipt.getFeSpace());
  rhsOperator.addZeroOrderTerm(new CoordsAtQP_ZOT(new F);
  ellipt.addVectorOperator(&rhsOperator);

  ellipt.addDirichletBC(1, new G);

  adapt.adapt();
}
```

Working with AMDiS: Post processing

ParaView (www.paraview.org)

- Open source, available for all platforms
- Parallel visualization and post processing of large data sets possible
- Open and flexible user interface
- Script language based on Python



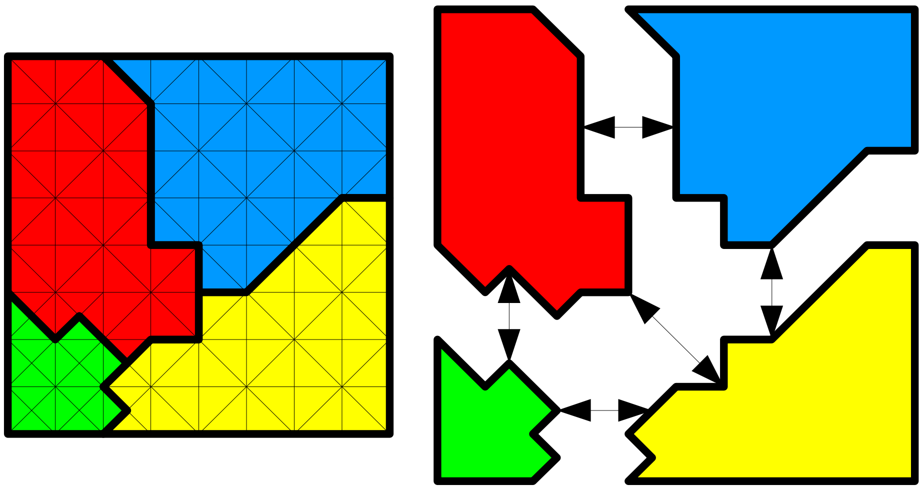
1. AMDiS

2. Parallelization

3. Applications

Thomas Witkowski (thomas.witkowski@tu-dresden.de)

AMDiS: MPI based parallelization



Parallelization:

- Non overlapping domain decomposition
- Mesh partitioning and distribution on macro level
- Use ParMetis for load balancing
- PETSc for solving the overall system
- Global matrix solver
- Schur complement approach

Main features:

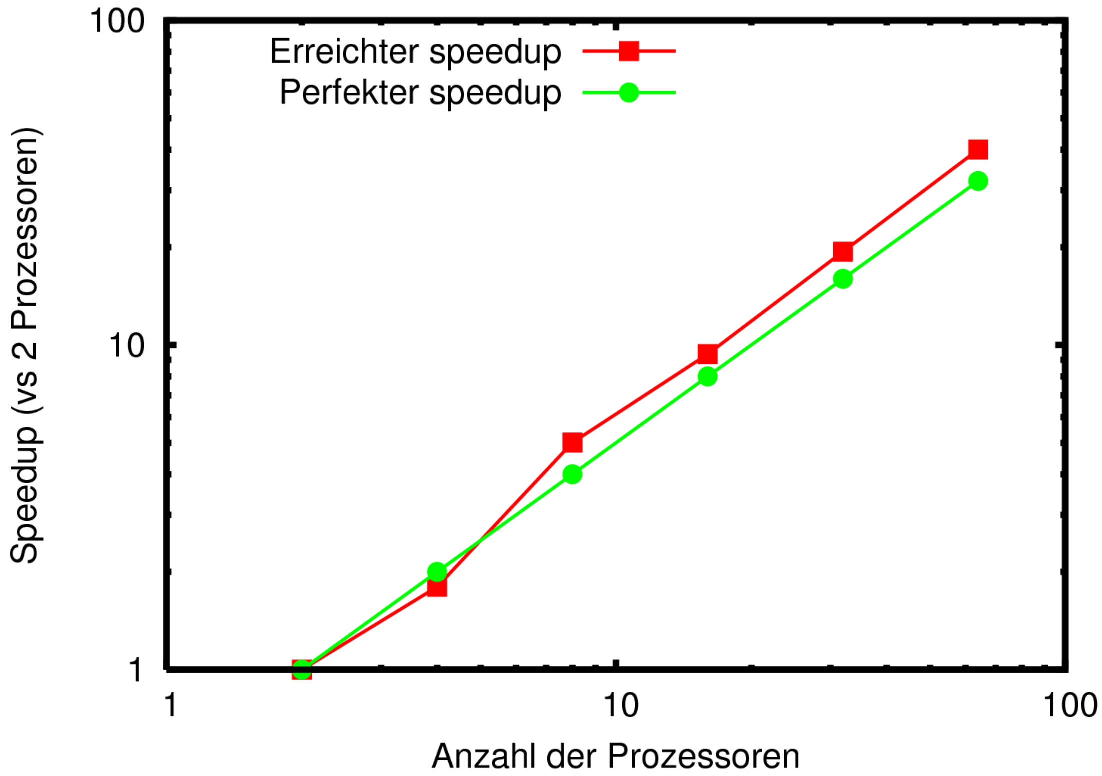
- 2D and 3D
- Adaptivity with dynamical repartitioning
- Arbitrary basis functions possible
- Periodic boundary conditions
- Support of HPC systems at TU Dresden (Batch System, Automatic restarting)
- Parallel visualization

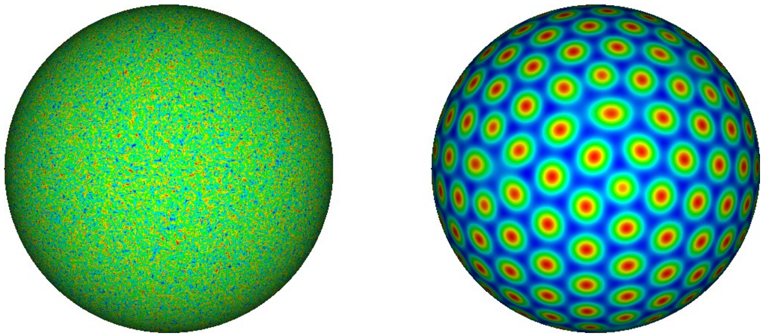
In work:

- Parallel preconditioners for the Schur complement approach
- Geometric multigrid for globally refined meshes (using PETSc's multigrid solvers)

AMDiS: MPI based parallelization

Numerical experiment: PFC on a sphere





Phase-field crystal (PFC) equation:

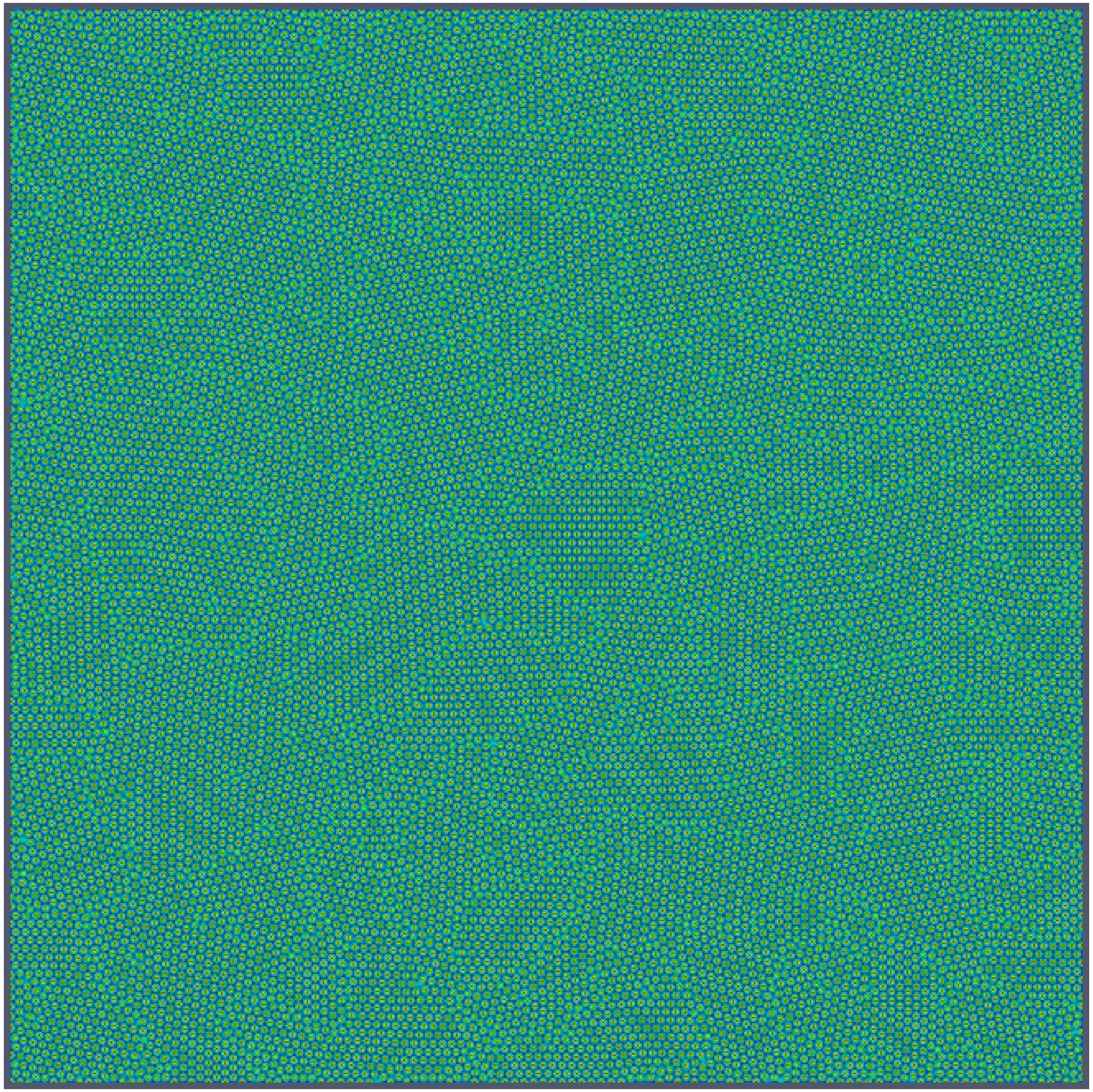
$$\begin{aligned} \partial_t \rho &= \Delta_\Gamma u \\ u &= 2\Delta_\Gamma v + v + f'(\rho) \\ v &= \Delta_\Gamma \rho \end{aligned}$$

Thomas Witkowski (thomas.witkowski@tu-dresden.de)

AMDiS: MPI based parallelization

AMDiS on 256 processors:

- PFC in 2D, globally refined mesh
- 4th order Lagrange basis functions
- 4.3 millon DOFs per variable
- 700 millions of non zero entries
- PETSc: 55 seconds to solve the system with TFQMR and block jacobi preconditioner



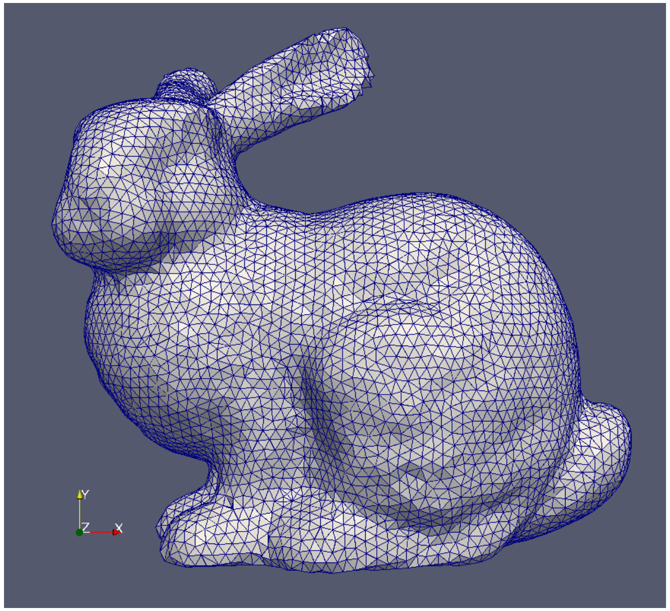
1. AMDiS
2. Parallelization
3. Applications

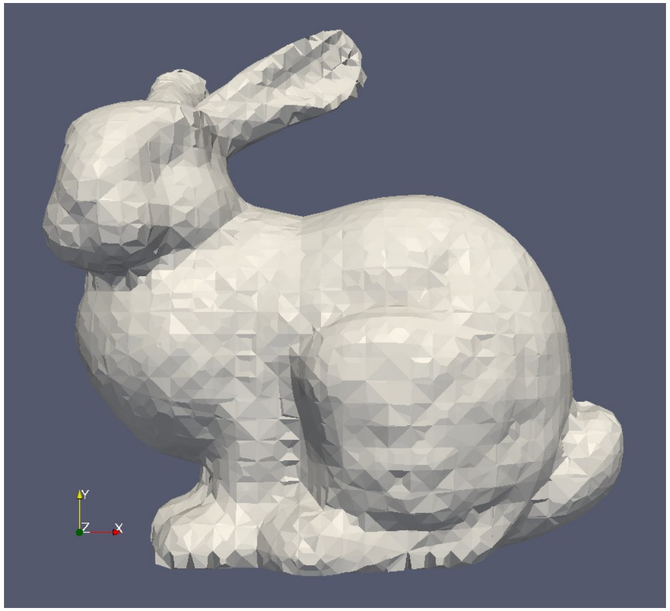
Thomas Witkowski (thomas.witkowski@tu-dresden.de)

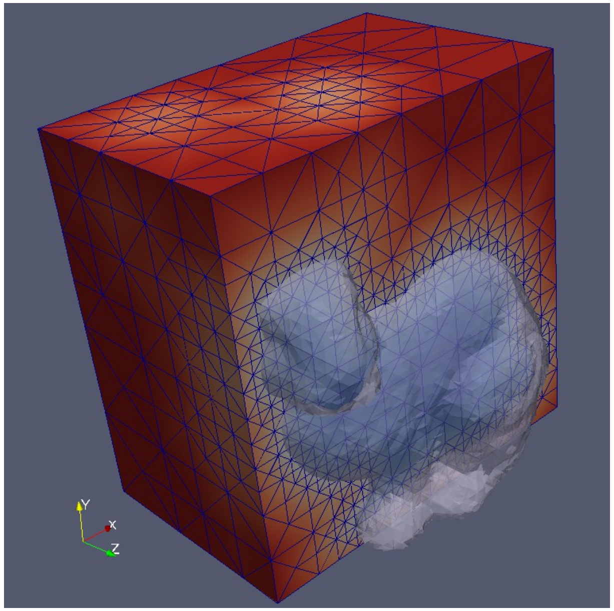
Complex geometries

Implicit representation of complex geometries

- Signed distance or Levelset function
- Function is defined within a square (2D) or cuboid (3D)
- Adaptive refinement on the 0- or 0.5-levelset







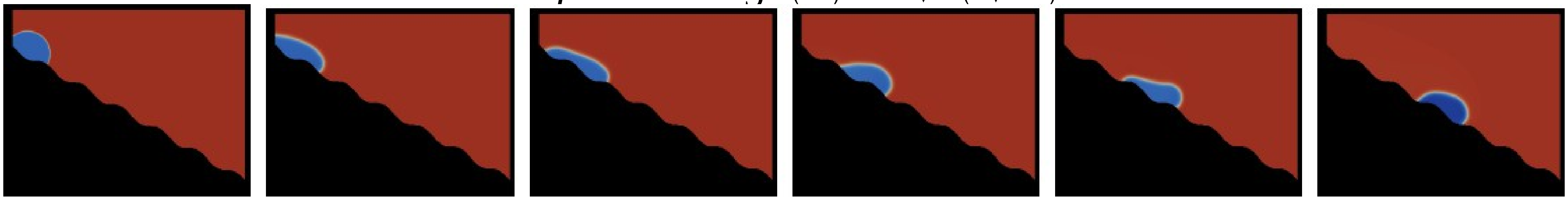
Rätz, Voigt; Nonlin. (2007), Rätz, Voigt; Comm. Math. Sci. (2006)

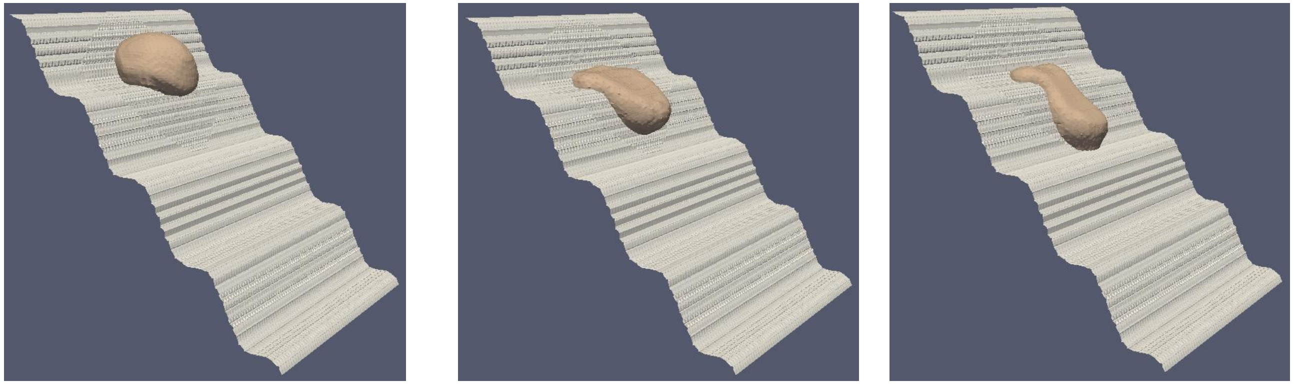
Thomas Witkowski (thomas.witkowski@tu-dresden.de)

Sliding droplets on nano structures

- Applications: Simulation of the Lotus effect
- Two phase flow
- Coupling of the Navier-Stokes and the Cahn-Hilliard equations

$$
\begin{aligned}
\rho(c)(u_t + (u \cdot \nabla) u) &= -\nabla \rho + \nabla \cdot (\nu(c) D) - \lambda c \nabla \mu \\
\nabla \cdot u &= 0 \\
c_t + u \cdot \nabla c &= \nabla \cdot (M(c) \nabla \mu) \\
\mu &= \epsilon^{-1} g'(c) - \epsilon \nabla \cdot (\nabla c)
\end{aligned}
$$

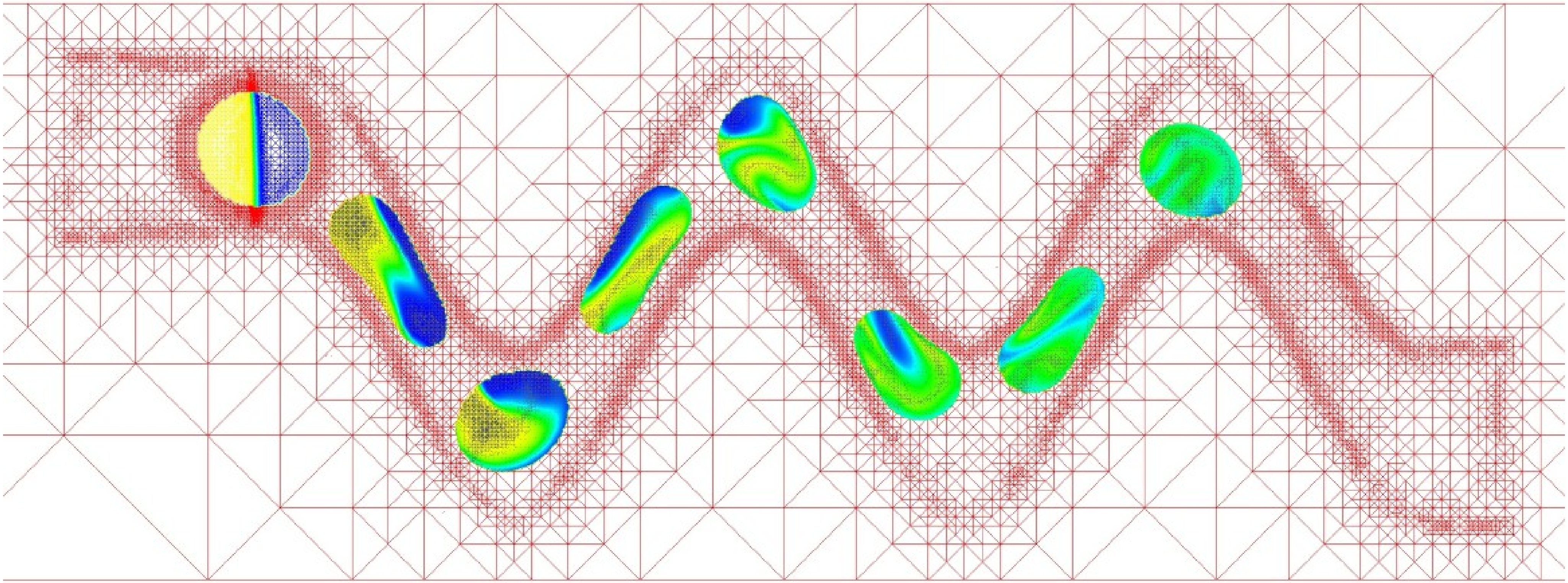




Thomas Witkowski (thomas.witkowski@tu-dresden.de)

Chaotic mixing of viscous fluids

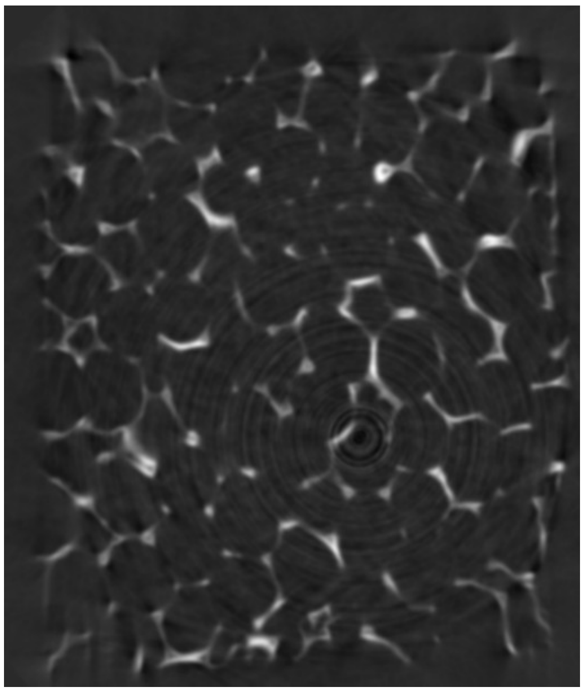
Mixing two fluids by advection inside droplets traveling through the confined micro-channels:

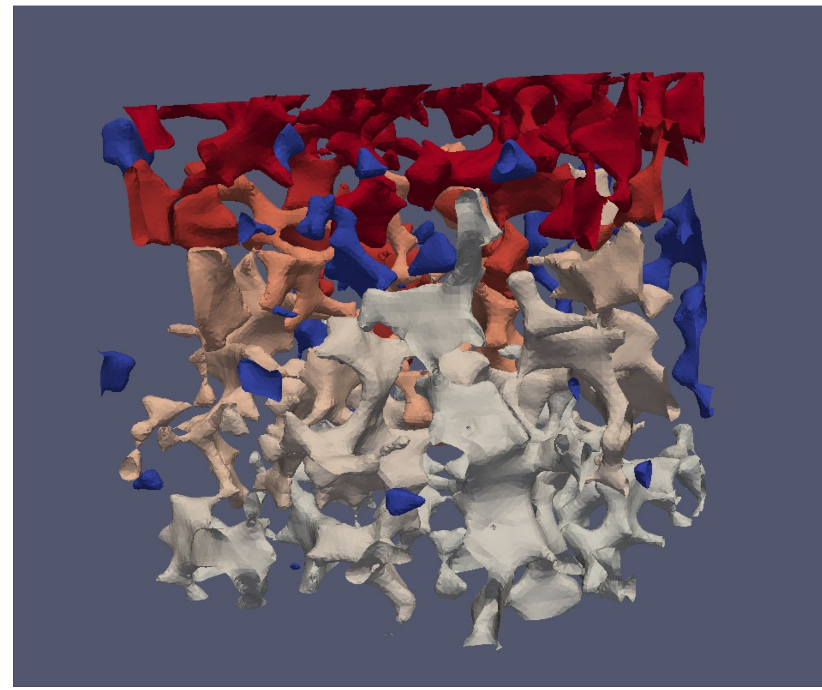


Aland, Lowengrub, Voigt; CMES 57(1), 2010

Metal foam

- Data source: tomograph images
- Using MeshConv to convert images into implicitly defined volume mesh
- Computing heat equation on this volume mesh



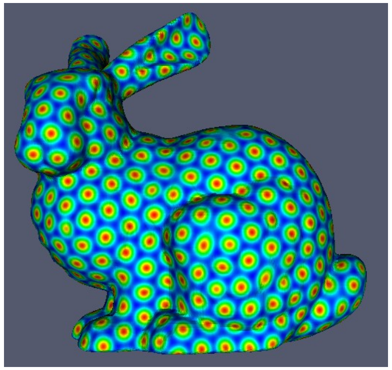


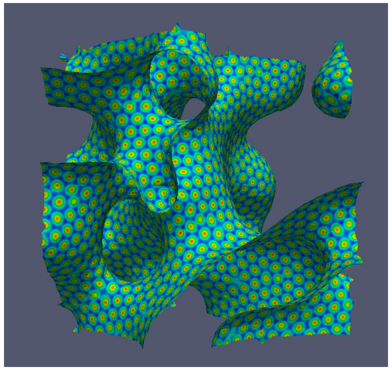
MeshConv: Florian Stenger, TU Dresden

In cooperation with Robert Müller (TU Dresden - ZIH)

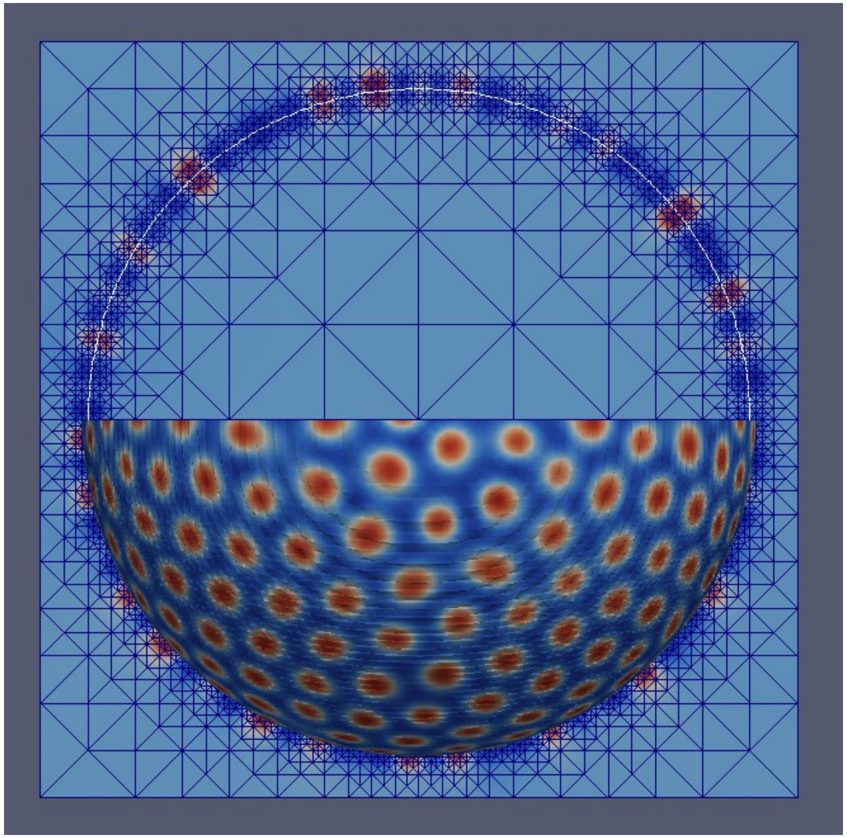
PFC on arbitrary surfaces

Explicit surface:

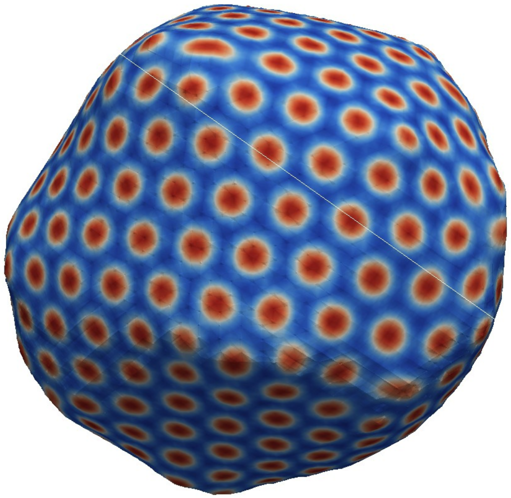




Implicit surface:



Virology:



Backofen, Gräf, Potts, Praetorius, Voigt, Witkowski; SIAM MMS (accepted), 2010 www.amdis-fem.com

Thank you for your attention!