deal.II Users and Developers Training

March 21 – 24 2016

Timo Heister (heister@clemson.edu)
Luca Heltai (luca.heltai@sissa.it)

Goals

- How to use deal.II for Finite Element computations
- Refresh numerical PDE knowledge
- Also:
  - Software best practices
  - C++, Debugging, IDEs, Visualization
  - Parallel computations with MPI

Schedule

	Monday	Tuesday	Wednesday	Thursday
9:30 – 10:45	Introduction	Dimension Independence	Embedded Surfaces	Parallel Computing
11:15 – 12:30	First Steps	Computing Errors	Adaptive Refinement	MPI demo
14:00 – 15:15	Basics of FEM	Higher Order Mappings	Systems of PDEs?	Exercises
15:45 – 17:00	Solving Poisson's eqn	Exercises	Exercises	Exercises

- See http://indico.ictp.it/event/7751/other-view?view=ictptimetable Green = in lecture room (tentative)
- Today:
  - What is deal.II?
  - Compiling, using an IDE
  - Overview about FEM
  - Basic tutorials (create mesh, solve Poisson's equation, visualization)
- Tuesday:
  - Finite Element Analysis (refinement, computing errors)
- Wednesday/Thursday:
  - Advanced topics
  - Time for projects

The plan

- Slides, some lectures on blackboard
- Many live demonstrations
- Exercises:
  - Work in groups of two!
  - Ask questions!
- Projects:
  - Required for MHPC students
  - Groups of two

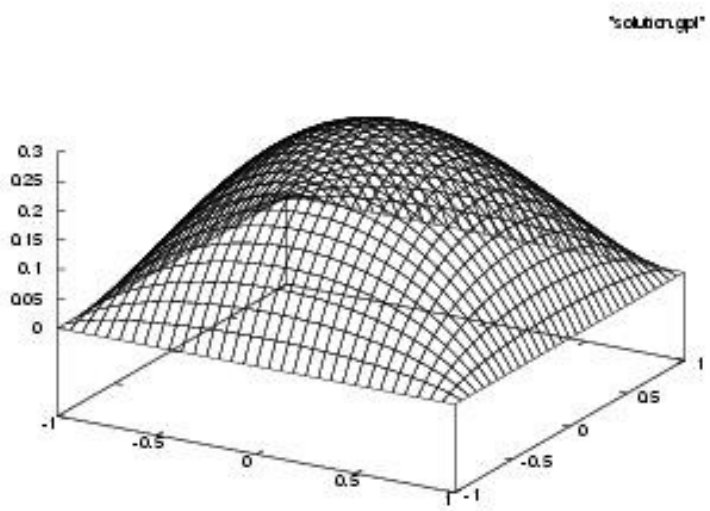
Resources

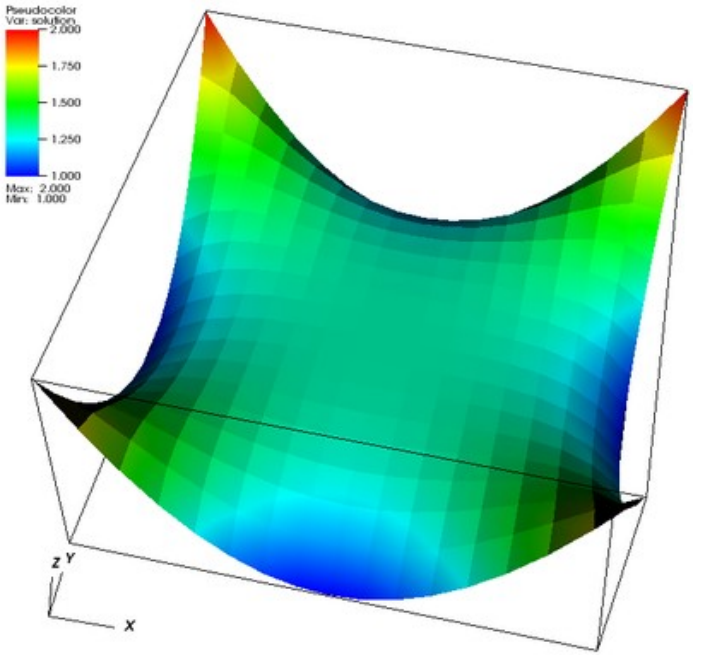
- http://indico.ictp.it/event/7751/overview
  - Schedule, Rooms, etc.
- http://dealii.org
  - Manual
  - Tutorial steps
  - Tutorial videos
- On your machine: folder /scratch/smr1909/
  - Slides
  - Example programs
  - Exercises
  - Other files

Finite Element Method

- Solve partial differential equations numerically
- Example:

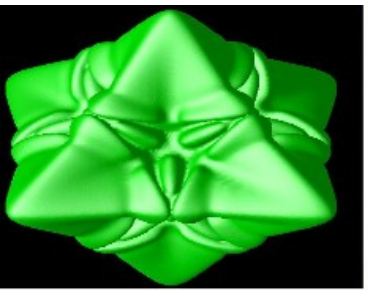
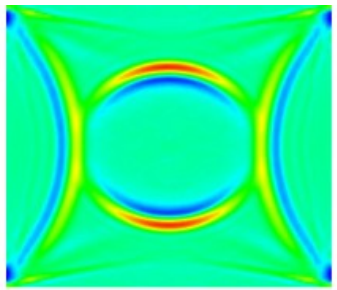
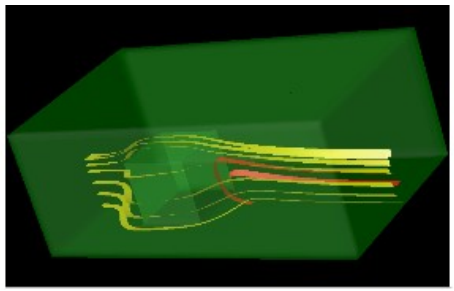
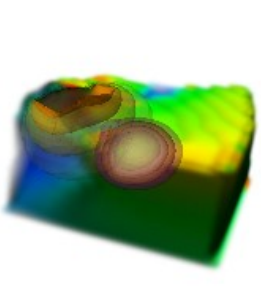
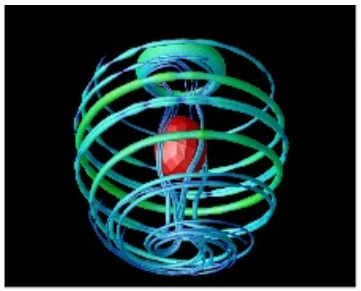
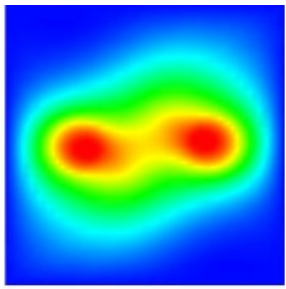
$$-\Delta u = f \quad \text{in } \Omega,$$
$$u = 0 \quad \text{on } \partial\Omega.$$





deal.II

- “A Finite Element **D**ifferential **E**quations **A**nalysis **L**ibrary”
- Open source, c++ library
- I am one of the four maintainers
- One of the most widely used libraries:
  - 800+ papers using and citing deal.II
  - ~600 downloads/month
  - 100+ people have contributed in the past 15 years
  - ~600,000 lines of code
  - 10,000+ pages of documentation
- Website: www.dealii.org

Features

- 1d, 2d, 3d computations, adaptive mesh refinement (on quads/hexes only)
- Finite element types:
  - Continuous and DG Lagrangian elements
  - Higher order elements, hp adaptivity
  - Raviart-Thomas, Nedelec, …
  - And arbitrary combinations
- PDEs on surfaces embedded in higher dimensions

Features, part II

- Linear Algebra
  - Own sparse and dense library
  - Interfaces to PETSc, Trilinos, UMFPACK, BLAS, ..
- Parallelization
  - Laptop to supercomputers
  - Multi-threading on multi-core machines
  - MPI: 64,000+ processors
- Output in many visualization file formats

Development of deal.II

- Professional-level development style
- Development in the open, repository on github.com
- Mailing lists for users and developers
- Test suite with 6,000+ tests after every change
- Platform support:
  - Linux/Unix
  - Mac
  - Windows
- Hope to see you on github.com or the mailing list!

Lab Setup

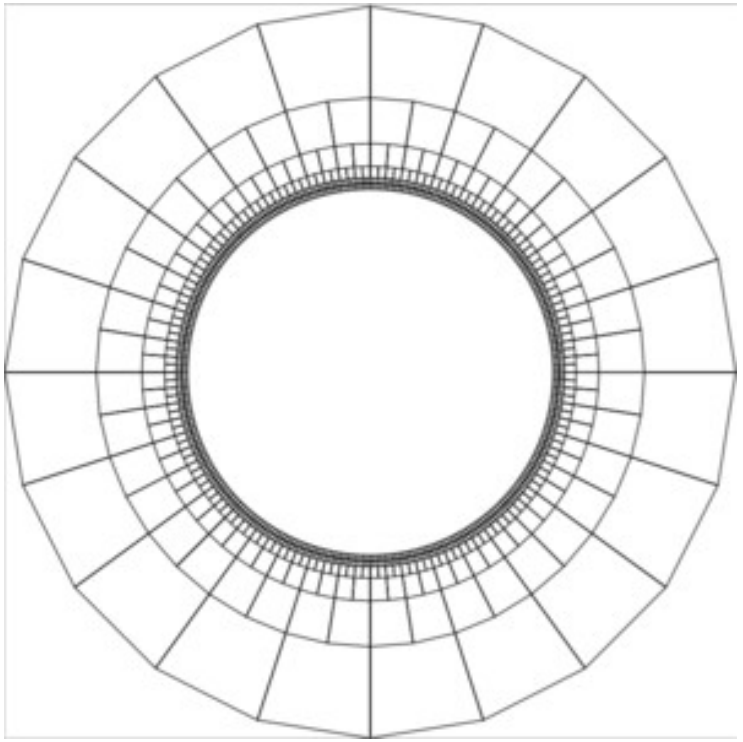
- deal.II and all required dependencies are already installed at

  /scratch/smr2909/

- (Demo, show lab01.pdf and run included step-1)

Lab 1 (step-1)

- See lab01.pdf
- Topic: creating meshes



Running examples

- In short:

```
cd examples/step-1
cmake .
make run
```

- cmake:
  - Detect configuration, only needs to be run once!
  - Input: CMakeLists.txt
  - Output: Makefile, (other files like CMakeCache.txt)
- make:
  - Code compilation
  - Tool to execute commands in Makefile, do every time you change your code
  - Input: step-1.cc, Makefile
  - Output: step-1 (the binary executable file)

- Run your program with

```
./step-1
```

- Or (compile and run):

```
make run
```

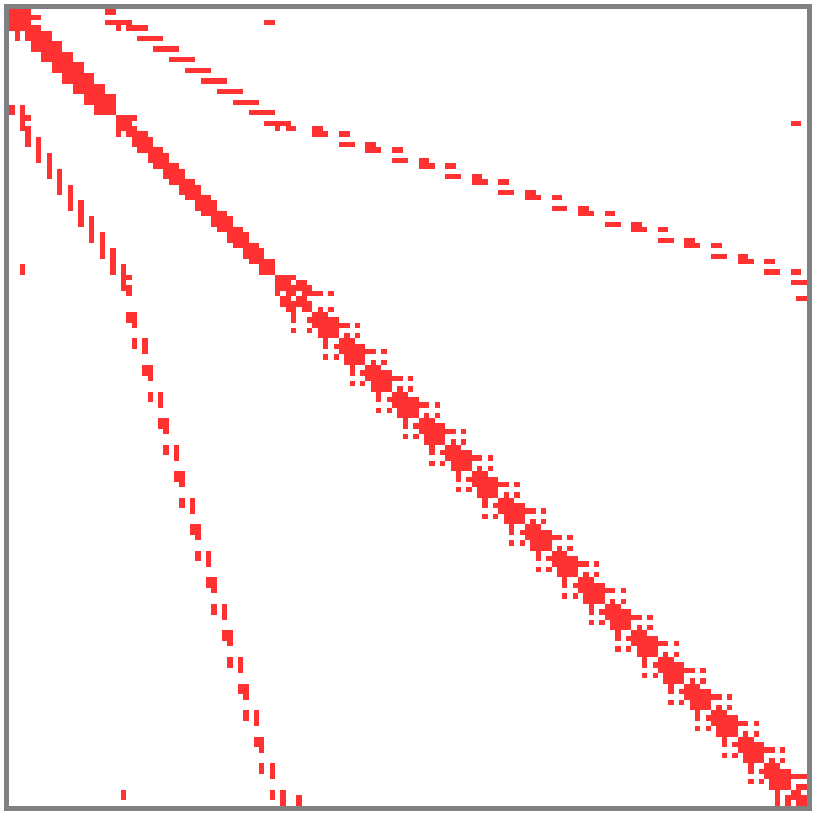
- Open in qtcreator IDE:

```
qtcreator .
```

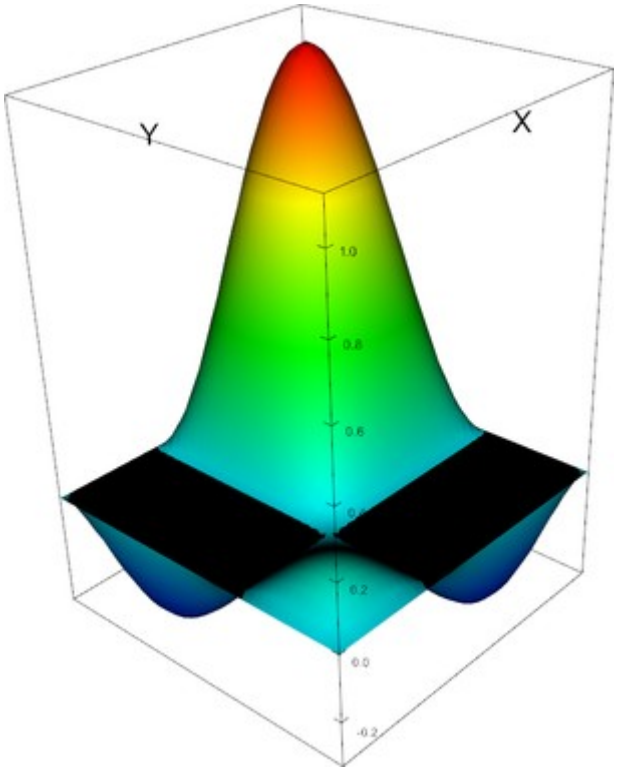
- Learn to use an IDE! (not vim, emacs, …)
- Copy into home directory required!
- [Demo the above and open project in qtcreator]

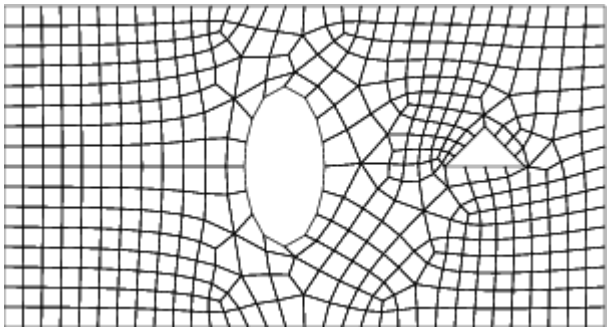
Lab 2 (step-2)

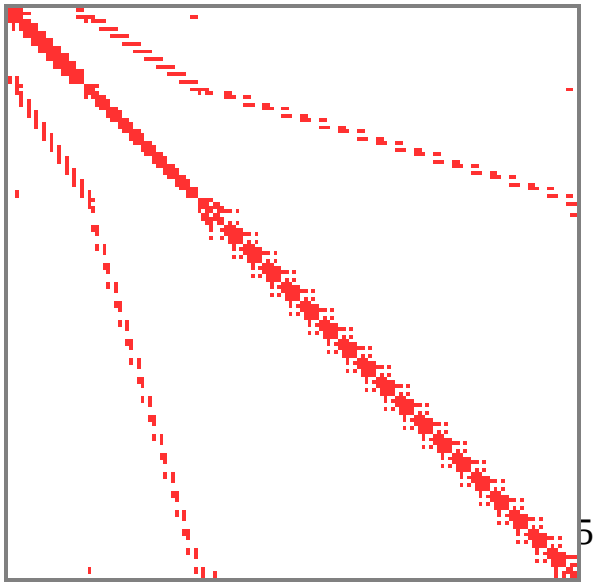
- See lab02.pdf
- Topic: sparsity patterns of matrices



Finite Element Assembly







$$A_{ij} = (\nabla\phi_i, \nabla\phi_j)$$

$$A_{ij} \approx \sum_K \sum_q J_K^{-1}(x_q)\nabla\phi_i(x_q) \cdot J_K^{-1}(x_q)\nabla\phi_j(x_q) \cdot |detJ(x_q)|w_q$$

in pseudo-code:

```
for i=0,...,N-1:
 for j=0,...,N-1:
  for all K:
    A_ij += \sum_q grad_phi(i,q) grad_phi(j,q) JxW(q)
```

But most of these contribution are zero. So we switch the order of the loops to get

```
for all K:
 for i = 0,...,N-1:
  for j = 0,...,N-1:
    A_ij += \sum_q grad_phi(i,q) grad_phi(j,q) JxW(q)
```

which I can simplify to only look at non-zero basis functions:

```
for all K:
 a = 0
 for alpha = 0,...,n_local_dofs:
  for beta = 0,...,n_local_dofs:
   for q:
    a_{alpha,beta} += grad_phi(alpha,q) grad_phi(beta,q) JxW(q)
 A_ij += a
```

Lab 3 (step-3)

- See lab03.pdf
- Topic: solving Poisson's equation