MATH 676

–

Finite element methods in scientific computing

Wolfgang Bangerth, Texas A&M University

Lecture 10:

A third example:

The *step-3* tutorial program
–
A first Laplace solver

step-3

Step-3 shows:

- How to set up a linear system
- How to assemble the linear system from the bilinear form:
  - The loop over all cells
  - The *FEValues* class
- Solving linear systems
- Visualizing the solution

step-3

Recall:

- For the Laplace equation, the bilinear form is written as a sum over all cells:

$$
\begin{aligned}
A_{ij} &= (\nabla \phi_i, \nabla \phi_j) \\
&= \sum_K \int_K \nabla \phi_i(x) \cdot \nabla \phi_j(x)
\end{aligned}
$$

step-3

Recall:

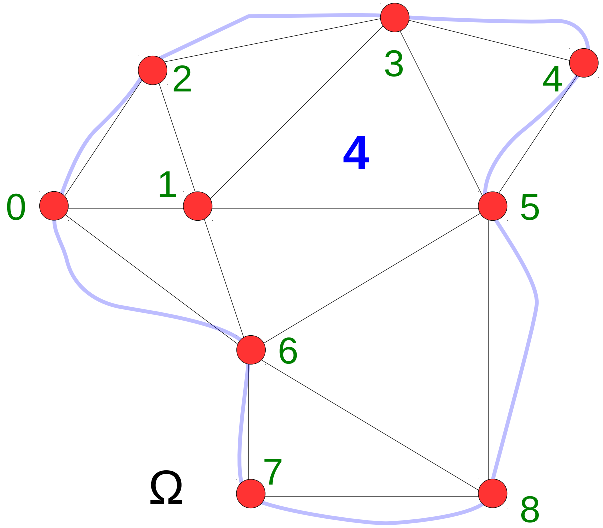
- For the Laplace equation, the bilinear form is written as a sum over all cells:

$$A_{ij} = (\nabla\phi_i, \nabla\phi_j) = \sum_K \int_K \nabla\phi_i(x)\cdot\nabla\phi_j(x)$$

- But on each cell, only few shape functions are nonzero!
- For Q_1, only $16=4^2$ matrix entries are nonzero per cell
- Only compute this (dense) sub-matrix, then “distribute” it to the global *A*
- Similar for the right hand side vector.

step-3

Example:



- On cell 4, only shape functions 1, 3, 5 are nonzero.
- We get a dense sub-matrix composed of rows and columns 1,3,5 of *A*.

step-3

Recall:

- We use quadrature

$$
\begin{aligned}
A_{ij}^K &= \int_K \nabla \phi_i(x) \cdot \nabla \phi_j \, dx \\
&\approx \sum_{q=1}^{Q} J_K^{-1}(\hat{x}_q) \hat{\nabla} \hat{\phi}_i(\hat{x}_q) \cdot J_K^{-1}(\hat{x}_q) \hat{\nabla} \hat{\phi}_j(\hat{x}_q) \underbrace{|\det J_K(\hat{x}_q)| \; w_q}_{=:\,\mathrm{JxW}}
\end{aligned}
$$

- We really only have to evaluate shape functions, Jacobians, etc., at quadrature points – not as functions
- All evaluations happen on the reference cell

step-3

Read through the commented program at

http://www.dealii.org/7.1.0/doxygen/deal.II/step_3.html

Then play with the program:

cd examples/step-3

cmake -DDEAL_II_DIR=/a/b/c . ; make run

This will run the program and generate output files:

ls -l

Then run *visit* to visualize the output

visit

Next step: Play by following the suggestions in the results section. This is the best way to learn!

MATH 676

–

Finite element methods in scientific computing

Wolfgang Bangerth, Texas A&M University