Finite element methods in scientific computing

Wolfgang Bangerth, Colorado State 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:

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

step-3

Recall:

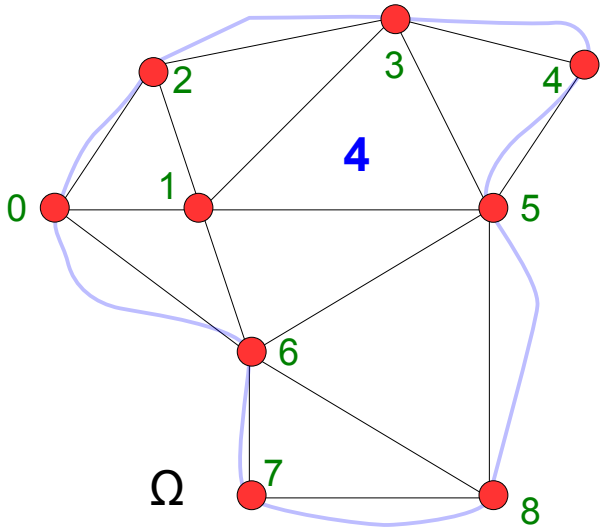
- For the Laplace equation, the bilinear form is written as a sum over all cells:

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

- 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 3x3 sub-matrix composed of rows and columns 1,3,5 of *A*.

step-3

Recall:

- We use quadrature (Lecture 3.97)

$$A_{ij}^K = \int_K \nabla \varphi_i(x) \cdot \nabla \varphi_j \, dx$$
$$\approx \sum_{q=1}^{Q} \underbrace{J_K^{-1}(\hat{x}_q)\hat{\nabla}\hat{\varphi}_i(\hat{x}_q)}_{=\nabla\varphi_i(x_q)} \cdot \underbrace{J_K^{-1}(\hat{x}_q)\hat{\nabla}\hat{\varphi}_j(\hat{x}_q)}_{=\nabla\varphi_j(x_q)} \underbrace{|\det J_K(\hat{x}_q)|}_{=:\mathrm{JxW}} w_q$$

- Only have to evaluate shape functions, Jacobians, etc., *at individual points*
- All evaluations happen on the reference cell

step-3

We need to compute

$$A = \sum_K A_{ij}^K$$

$$A_{ij}^K = \sum_{q=1}^{Q} \underbrace{J_K^{-1}(\hat{x}_q)\hat{\nabla}\hat{\varphi}_i(\hat{x}_q)}_{=\nabla\varphi_i(x_q)} \cdot \underbrace{J_K^{-1}(\hat{x}_q)\hat{\nabla}\hat{\varphi}_j(\hat{x}_q)}_{=\nabla\varphi_j(x_q)} \underbrace{|\det J_K(\hat{x}_q)| \; w_q}_{=:\mathrm{JxW}}$$

This suggests the following loop structure:

```
for (K : cells of the DoFHandler)
{
  A^K = 0
  for (q : quadrature points on cell K)
    for (i : 0...DoFs per cell)
      for (j : 0...DoFs per cell)
        A^K_ij += (integrand at x_q) * (JxW for point q)

  Copy the entries of A^K into the global matrix A
}
```

step-3

Read through the commented program at
http://dealii.org/9.5.0/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!

Finite element methods in scientific computing

Wolfgang Bangerth, Colorado State University