Finite element methods in scientific computing

Wolfgang Bangerth, Colorado State University

Lecture 20:

A seventh example:

The *step-20* tutorial program (part 1)
–
The mixed Laplace, a vector-valued problem

step-20

Step-20 shows:

- How to discretize a problem with more than one solution variable

- Building complicated solvers and preconditioners from simpler ones (next lecture)

Step-20 solves the mixed form of the Laplace equation:

$$K^{-1}\boldsymbol{u} + \nabla p = 0$$
$$-\nabla \cdot \boldsymbol{u} = -f$$

The associated weak form is

$$\left(\boldsymbol{\phi_u}, K^{-1}\boldsymbol{u}\right) - \left(\nabla \cdot \boldsymbol{\phi_u}, p\right) - \left(\phi_p, \nabla \cdot \boldsymbol{u}\right) = \left(\phi_p, -f\right)$$

step-20

From the weak form

$$(\boldsymbol{\phi_u}, K^{-1}\boldsymbol{u}) - (\nabla \cdot \boldsymbol{\phi_u}, p) - (\phi_p, \nabla \cdot \boldsymbol{u}) = (\phi_p, -f)$$

we get the matrix entries

$$A_{ij} = (\boldsymbol{\phi_{i,u}}, K^{-1}\boldsymbol{\phi_{j,u}}) - (\nabla \cdot \boldsymbol{\phi_{i,u}}, \phi_{j,p}) - (\phi_{i,p}, \nabla \cdot \boldsymbol{\phi_{j,u}})$$

which we transcribe into code using these substitutions:

$$\boldsymbol{\phi_{i,u}}(x_q) = \text{fe\_values[velocities].value(i,q)}$$
$$\phi_{i,p}(x_q) = \text{fe\_values[pressure].value(i,q)}$$
$$\nabla \cdot \boldsymbol{\phi_{i,u}}(x_q) = \text{fe\_values[velocities].divergence(i,q)}$$

step-20

To implement this problem, we need to:

- Define the appropriate finite element
  - Here, use the Raviart-Thomas element for *u*, DG for *p*
  - Use *FESystem* to combine it all

- Translate the weak form into code

- Make sure output looks as expected

step-20

Read through the commented program at
http://dealii.org/9.5.0/doxygen/deal.II/step_20.html

Then play with the program:
cd examples/step-20
cmake -DDEAL_II_DIR=/a/b/c . ; make run

This will run the program and generate output files:
ls -l

Then visualize the solutions.

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