Master in High Performance Computing

Theory and Practice of Finite Element Methods

Convergence plots: Bramble-Hilbert in action

Luca Heltai <luca.heltai@sissa.it>

International School for Advanced Studies (www.sissa.it)
Mathematical Analysis, Modeling, and Applications (math.sissa.it)
Master in High Performance Computing (www.mhpc.it)
SISSA mathLab (mathlab.sissa.it)

How to measure the Error?

- Method of Manufactured Solutions
  - Take the “u” you want as a solution, plug in the equations, get the boundary conditions and the right hand side that force the given “u”
  - Integrate (with a fine quadrature formula) the difference between the exact solution and the computed one (VectorTools::integrate_difference, or helper classes)
  - Possibly integrate the difference between the gradients of the exact and computed solutions

Error Estimates

Local Estimate:

$$\|u - \Pi u\|_{s,T_m} \lesssim \rho_m^{-s} h_m^{k+1} \, | \, u \, |_{k+1,T_m}$$

Global Estimate (for quasi uniform triangulations):

$$\sum_m \left(\|u - \Pi u\|_{s,T_m} \right) \lesssim h^{k+1-s} \, | \, u \, |_{k+1,\Omega}$$

Error Estimates

Local Estimate:

$$\|u - \Pi u\|_{s,T_m} \lesssim \rho_m^{-s} h_m^{k+1} \, | \, u \, |_{k+1,T_m}$$

If $V_h \subset H^s(\Omega)$

$$\|u - \Pi u\|_{s,\Omega} \lesssim h^{k+1-s} \, | \, u \, |_{k+1,\Omega}$$

To Reduce the Error:

- Globally, the error is dominated by *largest* element of the mesh and the $H^{k+1}(\Omega)$ norm of the exact solution
  - Reduce the overall size of the mesh h (**global refinement**), when we don't know the $H^{k+1}(\Omega)$ norm of the exact solution
  - Reduce the size of the elements where the solution has large $H^{k+1}(\Omega)$ norm, or where we estimate that $H^{k+1}(\Omega)$ norm of the solution would be large (**local refinement**)

Estimate the rate of convergence

- Once you have computed the error, how do we measure if we get the correct *convergence ratio?*

- *Consider Poisson Problem.* $V := H^1(\Omega)$

$$\| u - u_h \|_1 \lesssim \| u - \Pi u \|_1 \lesssim h^1 | u |_{2,\Omega}$$

$$\| u - \Pi u \|_0 \lesssim h^2 | u |_{2,\Omega}$$

We still need to prove that we can use u_h in the last estimate!

Estimate the rate of convergence

- Compute two successive solutions, on half the size of the mesh (i.e., after one global refinement):

$$\| u - u_{2h} \| \sim \tilde{C}(2h)^p$$

$$\| u - u_h \| \sim \tilde{C}(h)^p$$

$$\frac{\| u - u_{2h} \|}{\| u - u_h \|} \sim 2^p$$

$$p \sim \log_2 \left(\frac{\| u - u_{2h} \|}{\| u - u_h \|} \right)$$

Back to C++

- Today's program:
  - Poisson for general coefficients, boundary data, and rhs
  - Work on successively refined grids
  - Estimate $L^2(\Omega)$ and $H^1(\Omega)$ errors