MATH 676

–

Finite element methods in scientific computing

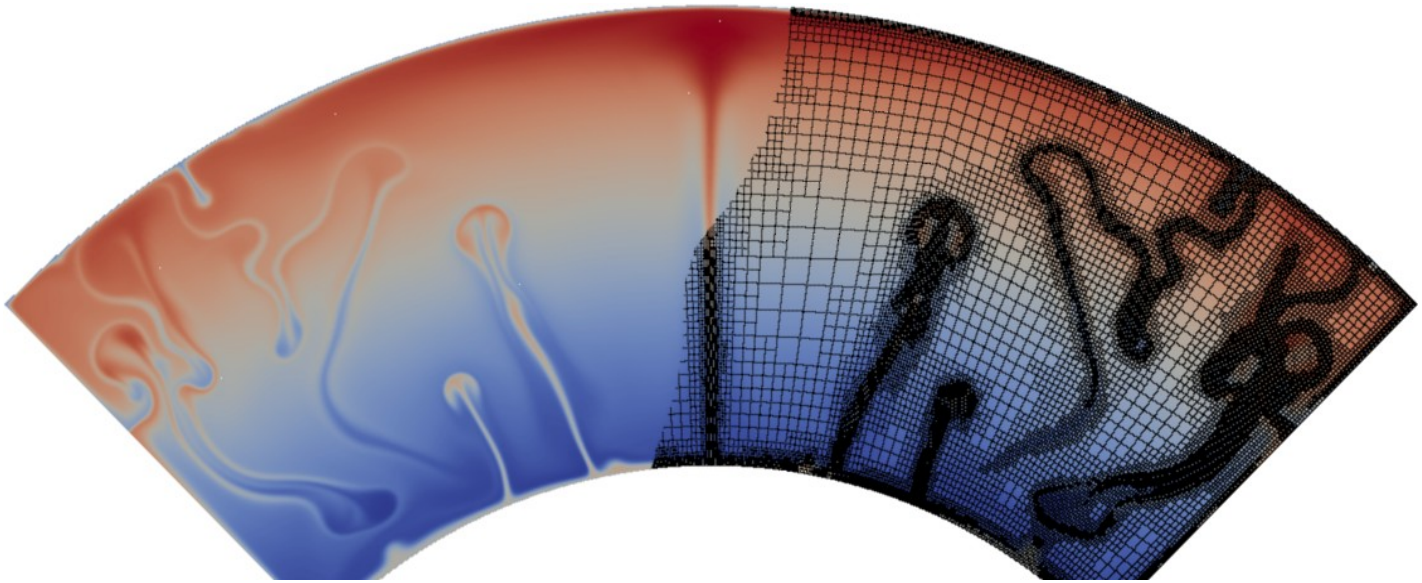
Wolfgang Bangerth, Texas A&M University

Lecture 17.25:

Generating adaptively refined meshes: Simple refinement indicators

Adaptive mesh refinement (AMR)

Example:



Intuitive goal: Use a fine mesh only where “something is happening”.

Question 1: Why?
Question 2: How?

Why adaptive mesh refinement (AMR)?

Recall from lecture 16: For many equations, the error has a general structure similar to this:

$$\|e\|^2_{H^1(\Omega)} \leq C^2 \sum_K h_K^2 |u|^2_{H^2(K)} \leq C^2 h^2 |u|^2_{H^2(\Omega)}$$

In particular, this is true for elliptic ("diffusion-dominated") second order PDEs.

Adaptive mesh refinement (AMR)

Approach: The optimal strategy to minimize the error while keeping the problem as small as possible is to *equilibrate* the local contributions

$$e_K \;=\; C\, h_K |u|_{H^2(K)}$$

That is, we want to choose

$$h_K \;\propto\; \frac{1}{|u|_{H^2(K)}}$$

Why adaptive mesh refinement (AMR)?

Recall from lecture 16: For many equations, the error has a general structure similar to this:

$$\|e\|_{H^1}^2 \leq C^2 \sum_K h_K^2 |u|_{H^2(K)}^2$$

Then choose the mesh size as:

$$h_K \propto \frac{1}{|u|_{H^2(K)}}$$

In other words: To reduce the error, we *only* need to make the mesh fine where the local H^2 norm is large!

Why adaptive mesh refinement (AMR)?

Recall from lecture 16: For many equations, the error has a general structure similar to this:

$$\|e\|_{H^1}^2 \leq C^2 \sum_K h_K^2 |u|_{H^2(K)}^2$$

Recall: The H^2 (semi-)norm is defined as

$$\|u\|_{H^2(K)}^2 = \int_K |u|^2 + |\nabla u|^2 + |\nabla^2 u|^2$$
$$|u|_{H^2(K)}^2 = \int_K |\nabla^2 u|^2$$

In other words: We only need to refine where the *second derivative* is large (= “where something is going on”).

A brief derivation

Why is this so: Consider the Laplace equation

$$-\Delta u = f \qquad u|_{\partial\Omega} = 0$$

and its weak form: *find* $u \in V := H_0^1$ *so that*

$$(\nabla u, \nabla v) = (f, v) \qquad \forall v \in V$$

Discretization: Let V_h be a finite dimensional (finite element) sub-space of V. Then the discrete problem reads:

Find $u_h \in V_h \subset V = H_0^1$ *so that*

$$(\nabla u_h, \nabla v_h) = (f, v_h) \qquad \forall v_h \in V_h \subset V$$

A brief derivation

From the two problems

$$(\nabla u, \nabla v) = (f, v) \qquad \forall\, v \in V$$

$$(\nabla u_h, \nabla v_h) = (f, v_h) \qquad \forall\, v_h \in V_h \subset V$$

we can deduce “Galerkin orthogonality”:

$$(\nabla (\underbrace{u - u_h}_{=:e}), \nabla v_h) = 0 \qquad \forall\, v_h \in V_h \subset V$$

Aside – why this is called “Galerkin orthogonality”:
The bilinear form

$$(\nabla f, \nabla g) = \int \nabla f(x) \cdot \nabla g(x)\, dx \;=:\; \langle f, g \rangle$$

defines a “scalar product” between vectors *f(x)*, *g(x)* in H^1_0.

A brief derivation

Next, consider the “energy norm error”:

$$\|\nabla(\underbrace{u-u_h}_{=:e})\|^2 = (\nabla(u-u_h),\nabla(u-u_h))$$

Galerkin orthogonality allows us to add a zero:

$$\begin{aligned}\|\nabla(\underbrace{u-u_h}_{=:e})\|^2 &= (\nabla(u-u_h),\nabla(u-u_h))+\underbrace{(\nabla(u-u_h),\nabla v_h)}_{=0}\\ &= (\nabla(u-u_h),\nabla(u-u_h+v_h))\end{aligned}$$

This is true for any choice of finite element function v_h!

In particular, let us choose $v_h=u_h-I_h u$

A brief derivation

Consider the “energy norm error”:

$$\|\nabla(u-u_h)\|^2 = (\nabla(u-u_h), \nabla(u-I_h u))$$

Next, recall the Cauchy-Schwarz inequality:

$$(f,g) \le \|f\| \, \|g\| \qquad \forall f,g \in L_2$$

Consequently:

$$\|\nabla(u-u_h)\|^2 \le \|\nabla(u-u_h)\| \, \|\nabla(u-I_h u)\|$$

$$\|\nabla(u-u_h)\| \le \|\nabla(u-I_h u)\|$$

A brief derivation

Consider the “energy norm error”:

$$\|\nabla(u-u_h)\| \leq \|\nabla(u-I_h u)\|$$

This is often called the “best-approximation property”.

Interpretation: Intuitively, this means that the finite element error is no larger than the *interpolation error*.

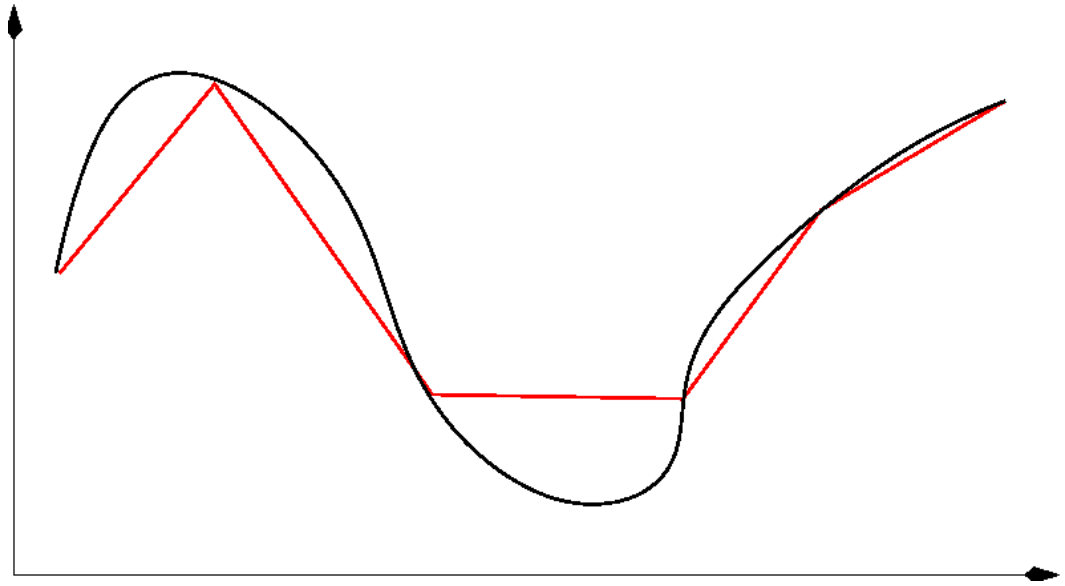
But:
– We can't compute the interpolant without the exact solution
– We *can* compute the finite element approximant

A brief derivation

Properties of the interpolant: Consider

$$\|\nabla(u-I_h u)\| = \left(\int_\Omega |\nabla(u-I_h u)|^2\right)^{1/2} = \left(\sum_K \int_K |\nabla(u-I_h u)|^2\right)^{1/2}$$

The interpolant is defined on each cell individually:



Black: $u(x)$
Red: $I_h u(x)$

Note: Error is large where the second derivative is large!

A brief derivation

Properties of the interpolant: Consider

$$\|\nabla(u-I_h u)\| = \left(\sum_K \int_K |\nabla(u-I_h u)|^2\right)^{1/2} = \left(\sum_K \|\nabla(u-I_h u)\|_K^2\right)^{1/2}$$

The “Bramble-Hilbert Lemma” provides the following for piecewise linear elements:

$$\|\nabla(u-I_h u)\|_K = \left(\int_K |\nabla(u-I_h u)|^2\right)^{1/2} \leq C h_K \|\nabla^2 u\|_K$$

$$\|\nabla(u-I_h u)\|_\Omega^2 = \sum_K \|\nabla(u-I_h u)\|_K^2 \leq C \sum_K h_K^2 \|\nabla^2 u\|_K^2$$

Or, for general elements of polynomial degree p:

$$\|\nabla(u-I_h u)\|_\Omega^2 \leq C \sum_K h_K^{p+1} \|\nabla^{p+1} u\|_K^2 = C \sum_K h_K^{2p} |u|_{H^{p+1}(K)}^2$$

A brief derivation

Taken all together: For the Laplace equation, using linear elements, the error satisfies

$$\|\nabla(u-u_h)\|_\Omega^2 \leq C\sum_K h_K^2 \|\nabla^2 u\|_K^2$$

This is called an "a priori" error estimate:

- We can say this about the error "up front"
- Right hand side does not involve computed solution u_h
- Not useful in itself because we don't know u

What to do with this?

Taken all together: For the Laplace equation, using linear elements, the error satisfies

$$\|\nabla(u-u_h)\|_\Omega^2 \leq C\sum_K e_K^2$$

$$e_K := h_K\|\nabla^2 u\|_K$$

How can we use this in practice:

- The e_K are called “cell-wise error estimators”
- We want to have a mesh that “equilibrates” the error estimators, i.e.,

$$e_K \approx \text{const} \quad \rightarrow \quad h_K \propto \frac{1}{\|\nabla^2 u\|_K}$$

What to do with this?

Taken all together: For the Laplace equation, using linear elements, the error satisfies

$$\|\nabla(u-u_h)\|_\Omega^2 \le C\sum_K e_K^2$$

$$e_K := h_K\|\nabla^2 u\|_K$$

How can we use this in practice:

- Can't evaluate e_K because we don't know u
- But maybe we can approximate/estimate

$$e_K = h_K\|\nabla^2 u\|_K \approx h_K\|\nabla_h^2 u_h\|_K =: \eta_K$$

using the computed solution u_h?

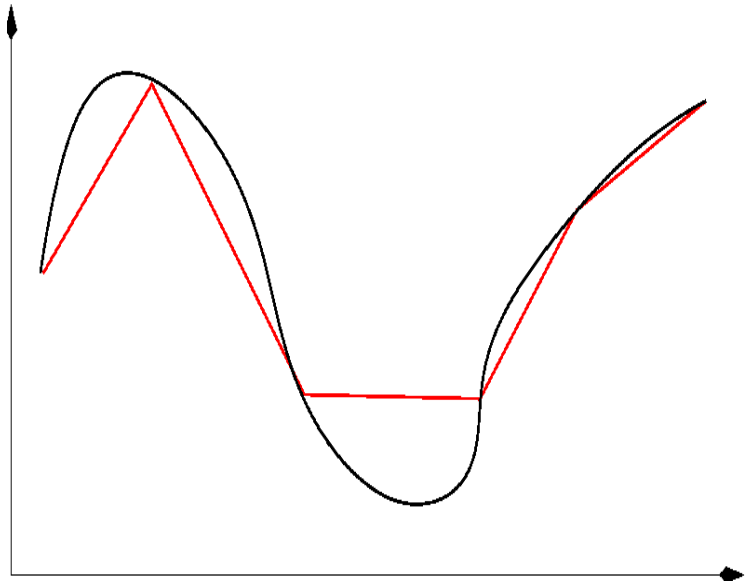
What to do with this?

Idea 1: Just approximate

$$\nabla^2 u \approx \nabla^2 u_h$$

This does not work:

- u_h is piecewise linear
- Second derivatives are zero inside cells
- Second derivatives are infinite at cell interfaces



What to do with this?

Idea 2: Try a finite difference approximation:

$$\nabla^2 u \approx \frac{\nabla u_h(x^+) - \nabla u_h(x^-)}{h} = \frac{[\nabla u_h]_i}{h}$$

Where the “jump in gradient” is defined as

$$[\nabla u_h]_i := \lim_{\varepsilon \to 0} \nabla u_h(x_i + \varepsilon) - \nabla u_h(x_i - \varepsilon)$$

This *does* work:

- Size of the jump in gradient is an indicator of the second derivative
- Can generalize to

$$\|\nabla^2 u\|_K^2 = \int_K |\nabla^2 u|^2 \approx \sum_{i \in \partial K} \frac{[\nabla u_h]_i^2}{h}$$

What to do with this?

Summary: We needed to approximate the cell-wise error indicator

$$\|\nabla(u-u_h)\|_\Omega^2 \leq C\sum_K e_K^2$$

$$e_K := h_K\|\nabla^2 u\|_K$$

We can do this in 1d using

$$\eta_K := h_K\left(\sum_{i\in\partial K}\frac{[\nabla u_h]_i^2}{h_K}\right)^{1/2}$$

and in 2d/3d using

$$\eta_K := h_K^{1/2}\left(\int_{\partial K}|[\nabla u_h]|^2\right)^{1/2}$$

What to do with this?

Aside: Why the power of *h*?

Consider the physical units in 1d:

$$e_K := h_K \|\nabla^2 u\|_K = \underbrace{h_K}_{L} \Big(\int_K |\underbrace{\nabla^2 u}_{1/L^2}|^2 \underbrace{dx}_{L} \Big)^{1/2} \quad \rightarrow \quad L^{-1/2}$$

Same for the approximation:

$$\eta_K := \underbrace{h_K}_{L} \Big(\sum_{i \in \partial K} \underbrace{\frac{1}{h_K}}_{L^{-1}} [\underbrace{\nabla u_h}_{1/L}]_i^2 \Big)^{1/2} \quad \rightarrow \quad L^{-1/2}$$

What to do with this?

Aside: Why the power of *h*?

Consider the physical units in 2d:

$$e_K := h_K \|\nabla^2 u\|_K = \underbrace{h_K}_{L} \Big(\int_K |\underbrace{\nabla^2 u}_{1/L^2}|^2 \underbrace{dx}_{L^2} \Big)^{1/2} \quad \rightarrow \quad 1$$

Same for the approximation:

$$\eta_K := \underbrace{h_K^{1/2}}_{L^{1/2}} \Big(\int_{\partial K} |[\underbrace{\nabla u_h}_{L^{-1}}]|^2 \underbrace{dx}_{L} \Big)^{1/2} \quad \rightarrow \quad 1$$

What to do with this?

Conclusions: If you are solving an equation for which:

- the best-approximation property holds:

$$\|\nabla(u-u_h)\| \leq C\|\nabla(u-I_h u)\|$$

- you are using linear elements (Q_1 or P_1)

Then: The indicator

$$\eta_K := h_K^{1/2}\left(\int_{\partial K}|[\nabla u_h]|^2\right)^{1/2}$$

is a reasonable approximation to the true error on cell K.

The “Kelly” error estimator

Kelly, de Gago, Zienkiewicz, Babuska, 1983:
For the Laplace equation, the following is indeed true:

$$\|\nabla(u-u_h)\|^2 \leq C\sum_K \eta_K^2$$

$$\eta_K = h_K^{1/2}\left(\int_{\partial K} |[\nabla u_h]|^2\right)^{1/2}$$

In other words: For the Laplace equation, we can even *prove* that our approximation leads to a correct estimate of the error!

Because of this paper, η_K is typically called the “Kelly error estimator”.
In deal.II, it is implemented in the KellyErrorEstimator class.

The “Kelly” error estimator

Observation:
While the “Kelly” error estimator

$$\eta_K = h_K^{1/2} \left(\int_{\partial K} |[\nabla u_h]|^2 \right)^{1/2}$$

only estimates the error for the Laplace equation with linear elements, in practice it also yields a good criterion to refine the meshes

- for higher order elements
- for many (most?) other equations

It is therefore widely used.

MATH 676

–

Finite element methods in scientific computing

Wolfgang Bangerth, Texas A&M University