TMA205 Differential Equations and Scientific Computing for Kb, part A

Notes on Adaptive Algorithms

October 11, 2002

1 A posteriori error estimates

An a posteriori error estimate provides an estimate of the error in terms of the computed finite element solution U, instead of the unknown exact solution u. Often a posteriori error estimates are used to investigate the accuracy of the finite element approximation and to design adaptive algorithms. An adaptive algorithm uses information extracted from earlier computations to modify the mesh in order to obtain a more efficient approximation. We return to these issues below.

1.1 1D case

For the model problem

$$-u_{xx} = f \quad \text{in } (0,1), \tag{1}$$
$$u(0) = u(1) = 0, \tag{2}$$

the following a posteriori error estimate holds

$$\boxed{\|(u-U)_x\| \leq c\left(\sum_{K\in T} R_K^2(U)\right)^{1/2}} \tag{3}$$

Here $T = \{K\}$ is the partition of $(0,1)$ into intervals K of size h_K and the element residual $R_K(U)$ is defined by

$$\boxed{R_K(U) = h_K\|f + U_{xx}\|_{L^2(K)}} \tag{4}$$

Note that for piecewise linear approximation $U_{xx} = 0$ so that the residual simplifies to

$$R_K(U) = h_K\|f\|_{L^2(K)}. \tag{5}$$

1.2 2D case

For the model problem

$$-\Delta u = f \quad \text{in } \Omega, \tag{6}$$
$$u = 0 \quad \text{on } \partial\Omega, \tag{7}$$

the following a posteriori error estimate holds

$$\boxed{\|\nabla(u-U)\| \leq c\left(\sum_{K\in T} R_K^2(U)\right)^{1/2}} \tag{8}$$

Here $T = \{K\}$ is the partition of Ω into triangles K of size h_K and the element residual $R_K(U)$ is defined by

$$\boxed{R_K(U) = h_K\|f + \Delta U\|_{L^2(K)} + h_K^{1/2}\|[n\cdot\nabla U]\|_{L^2(\partial K\setminus\partial\Omega)}/2} \tag{9}$$

where $[n\cdot\nabla U]$ denotes the jump in the normal derivative of U at an interior edge $\partial K_1 \cap \partial K_2$, i.e.,

$$[n\cdot\nabla U]|_{\partial K_1\cap\partial K_2} = n_1\cdot\nabla U_1 + n_2\cdot\nabla U_2 \tag{10}$$

with $U_i = U|_{K_i}$ and n_i the exterior unit normal of K_i. Note that for piecewise linear approximation $\Delta U = 0$ so that we get a simplification of the interior part of the residual.

2 Implementation of an adaptive algorithm

Based on the a posteriori error estimate we may construct an adaptive algorithm, which consists of four main components:

- Computation of the element residuals of a computed solution.
- Selection of elements to be refined.
- A refinement algorithm.
- A stopping criterion.

We now present the implementation of these four steps.

2.1 Computation of the element residuals

In practice, we calculate expressions (4) and (9) using numerical quadrature, for instance two point Gauss quadrature in one dimension. It is convenient to store the element residuals in a vector.

2.2 Refinement criteria

There are different possibilities for selecting the elements to be refined based on the element residuals. Here we shall consider the following basic criterion: refine an element K if

$$R_K(U) > \kappa \max_{K \in T} R_K(U), \tag{11}$$

with κ a parameter in $(0, 1)$ provided by the user. Note that $\kappa = 0$ gives uniform refinement and $\kappa = 1$ gives no refinement at all.

2.3 A refinement algorithm

The refinement algorithm takes a list of elements to be refined and calculates a new mesh such that all the elements in the list are refined.

2.3.1 1D case

In the one dimensional case the refinement algorithm simply consists of inserting an extra node at the center of each element which is refined. In other words, if we are refining $[x_i, x_{i+1}]$ we replace it by $[x_i, (x_i + x_{i+1})/2] \cup [(x_i + x_{i+1})/2, x_{i+1}]$, and finally we reorder the nodes. This is easily implemented by simply adding the coordinate of the midpoint of all the elements to be refined to the vector of nodes and then sort in increasing order.

2.3.2 2D case

In two dimensions the computation of a locally refined mesh is more complex compared to one dimension. There are two important issues to consider when constructing a refinement algorithm: 1. invalid triangles are not allowed and we wish to refine as few elements as possible which are not in the list of elements to be refined and 2. it is important that the minimal angle in the triangulation is kept as large as possible.

The Rivara algorithm A basic algorithm for creating such refinements is the Rivara algorithm. In the Rivara algorithm an element is always refined by inserting an additional edge from the midpoint of the longest side to the opposite corner, see Figure 1. Typically, such a refinement manufactures invalid triangles, i.e., triangles containing an edge with a hanging node.

The algorithm first refines all elements which are to be refined and then refines all invalid triangles created, next the last step is repeated until there are no invalid triangles left.

One can prove that the Rivara algorithm terminates and that the minimal angle in a refined mesh is never less than half the minimal angle in the original mesh.

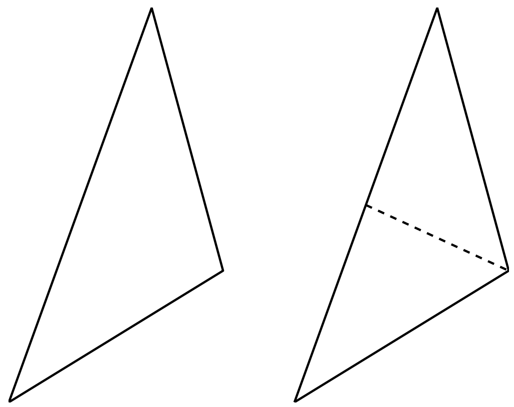

Figure 1: Refinement of triangle by inserting an edge from the midpoint of the longest edge to the opposite corner.

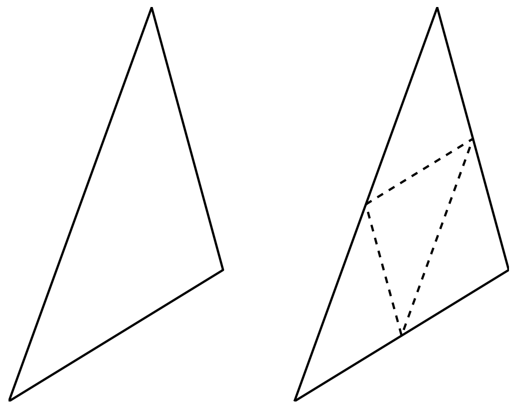

Figure 2: Refinement of triangle by inserting edges between the midpoints of all three edges.

A variant of the Rivara algorithm To obtain a stronger and more uniform refinement one may first divide all triangles which are to be refined in four equal sub-triangles, see Figure 2, and then invoke the Rivara refinement algorithm as above. Again the algorithm terminates and the minimal angle in a refined mesh is never less than half the minimal angle in the original mesh.

2.4 The stopping criterion

The stopping criterion determines when the adaptive algorithm should stop. For instance the stopping criterion may be: a maximum bound on the number of refinement levels, number of degrees of freedom in the approximation, the memory usage, the time of the computation, the total size of the residual, or a combination of these.

2.5 An adaptive algorithm

A basic adaptive algorithm takes the form. Given a mesh T and a computed solution U:

1. Determine if the stopping criterion is satisfied. If it is satisfied then stop the computation, if not continue.
2. Compute the vector of element residuals of a computed solution U.
3. Select the elements to be refined.
4. Calculate the refined mesh.
5. Compute the solution on the refined mesh and go to 1.

If the stopping criterion depends on the residual then we simply switch the order of 1 and 2.

3 Derivation of the a posteriori error estimate

3.1 1D case

We derive the a posteriori error estimate as follows

$$
\begin{aligned}
\|e_x\|^2 &= \int_\Omega e_x^2 dx && (12)\\
&= \int_\Omega e_x (e - \pi_h e)_x dx && (13)\\
&= \sum_K \int_K e_x (e - \pi_h e)_x dx && (14)\\
&= \sum_K [e_x (e - \pi_h e)]_{x_{K,l}}^{x_{K,r}} - \int_K e_{xx} (e - \pi_h e) dx && (15)\\
&= \sum_K \int_K (f + U_{xx})(e - \pi_h e) dx && (16)\\
&\le \sum_K \|f + U_{xx}\|_{L^2(K)} \|e - \pi_h e\|_{L^2(K)} && (17)\\
&\le \sum_K c h_K \|f + U_{xx}\|_{L^2(K)} \|e_x\|_{L^2(K)} && (18)\\
&\le c \left(\sum_K h_K^2 \|f + U_{xx}\|_{L^2(K)}^2 \right)^{1/2} \left(\sum_K \|e_x\|_{L^2(K)}^2 \right)^{1/2} && (19)\\
&= c \left(\sum_K h_K^2 \|f + U_{xx}\|_{L^2(K)}^2 \right)^{1/2} \|e_x\|. && (20)
\end{aligned}
$$

Here we used: the definition of the L^2-norm in (12), the Galerkin orthogonality in (13); splitting of the integral into element contributions in (14); element-wise partial integration in (15) together with the notation $x_{K,l}$ and $x_{K,r}$ for the left and right node of K; the facts that $e(x_i) - \pi_h e(x_i) = 0$ in the nodes x_i and that $-e_{xx} = -u_{xx} + U_{xx} = f + U_{xx}$ in (16); Cauchy-Schwarz inequality on each term in (17); interpolation error estimate on each element in (18); Cauchy-Schwarz inequality on the sum in (19); and finally we identify $\|e_x\|$ in (20).

Now the a posteriori error estimate follows immediately.

3.2 2D case

The proof in the 2D case is similar. Using Green's formula we get

$$\|\nabla e\|^2 = \sum_K \int_{\partial K} n \cdot \nabla e(e - \pi_h e)ds - \int_K \Delta e\,(e - \pi_h e)dx. \tag{21}$$

The second term on the right-hand side can now be estimated in the same way as above. Unfortunately the first term on the right-hand side does not vanish as in one dimension, since $e - \pi_h e$ is zero only at the nodes not on the edges. Instead we note that there are two contributions for each interior edge $\partial K_1 \cap \partial K_2$, one from triangle K_1 and one from triangle K_2. Summing these contributions we get

$$\int_{\partial K_1 \cap \partial K_2} (n_1 \cdot \nabla e_1(e_1 - \pi_h e_1) + n_2 \cdot \nabla e_2(e_2 - \pi_h e_2))ds, \tag{22}$$

where $e_i = e|_{K_i}$ and n_i is the exterior unit normal of K_i for $i = 1, 2$. Using the facts that the exact solution has a continuous normal derivative and that the error and its interpolant are continuous we get

$$\int_{\partial K_1 \cap \partial K_2} [n \cdot \nabla U](e - \pi_h e)ds. \tag{23}$$

This is the term which gives rise to the edge contribution in the element residual in two dimensions.