CE394M: FEM errors

Krishna Kumar

University of Texas at Austin

krishnak@utexas.edu

March 11, 2020

Overview

1. Error estimates
   - A priori estimates
   - Posterior error estimation

A priori estimates

- A priori error estimation is the analysis of the errors in a finite element simulation before actually performing an analysis.
- It is an abstract concept since it cannot quantify the error in a computed simulation, but analysis can tell something about whether the finite element solution will converge to the exact solution as the mesh is refined, and how quickly it will converge if the element type is changed or the mesh is refined.

Functional norms

Before considering the error, we need to decide how it should be measured. The question is how to measure a function? The answer to this is to use function norms. A common norm is known as the L^2 norm. For a function on a domain Ω, the L^2 norm is defined as:

$$\|u\|_0 = \left(\int_\Omega u^2 d\Omega \right)^{1/2}$$

This norm provides a measure of 'how big' a function is. A measure of the difference between two functions u and v is provided by:

$$\|u - v\|_0 = \left(\int_\Omega (u - v)^2 d\Omega \right)^{1/2}$$

It is only possible that $\|u - v\|_0 = 0$ if $u = v$.

Functional norms

The L^2 norm measures the 'magnitude' of a function, but does not reflect any details of the derivatives. Other norms are the H^1 norm:

$$\|u\|_0 = \left(\int_\Omega u^2 + \nabla u \cdot \nabla u d\Omega \right)^{1/2}$$

We will use function norms to measure the difference between the exact solution and the finite element solution. For example, we can define a particular measure of the error e as:

$$\|u - u_h\|_0 = \left(\int_\Omega (u - u_h)^2 d\Omega \right)^{1/2}$$

where u is the exact solution and u_h is the finite element solution.

How good is the FE solution?

Inevitably, the finite element solution is not exact. But how do we know that it bears any relation to the exact solution? If the mesh is refined, will the error reduce? We may say intuitively yes, but this is not very satisfactory. This is where a priori analysis helps. A priory error estimate:

$$\|u - u_h\|_s \leq Ch^\beta \|u\|_{k+1}$$

- s is the norm of interest,
- k is the polynomial order of the shape functions (complete polynomials),
- $\beta = min(k+1-s, 2(k+1-m))$,
- m is the highest order derivative appearing in the weak form
- C is an unknown constant that does not depend on h or the exact solution u.

What is of key interest is the exponent β.

Error in the temperature/displacement field

For a steady heat conduction or elasticity problem, for the error in the temperature/displacement field:

- $s = 0$
- $m = 1$

Therefore, for $k \geq 1$,

$$\|u - u_h\|_s \leq Ch^{k+1} \|u\|_{k+1}$$

The solution converges with order $O(h^{k+1})$. Using quadratic shape functions, the error converges with order $O(h^3)$. So, if the element size is halved, the error will be reduced by a factor of eight.

Error in the strain/flux field

For a steady heat conduction or elasticity problem, for the error in the strain/flux field we have:

- $s = 1$
- $m = 1$

Therefore, for $k \geq 1$,

$$\|u - u_h\|_s \leq Ch^k \|u\|_{k+1}$$

For a given element, the order of convergence in the first derivative is one order less than for the unknown field itself. Hence, the stresses in a finite element simulation are usually '*less*' accurate than the displacements.

Posterior error estimation and adaptivity

A *posterior error* estimation involves quantification of the error after a simulation has been preformed.

If we have an a posterior error estimator for a particular problem, a simulation can been performed, the error estimated in various part of the domain and the mesh and finite element type can be adjusted in order to reduce the error to a prescribed value. This is known as *adaptivity*.

The two obvious strategies are known as:

- *h-adaptivity*: this involves modify the mesh.
- *p-adaptivity*: this involves changing the finite element type.

Posterior error estimation and adaptivity

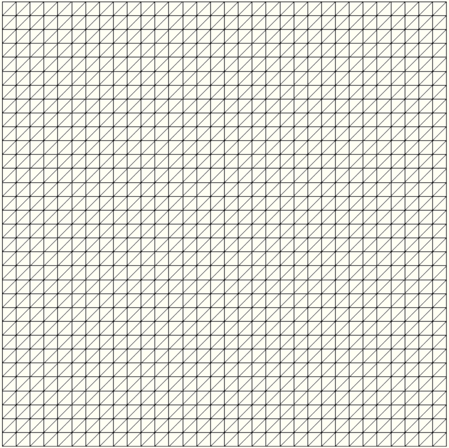
A *posterior error* estimation involves quantification of the error after a simulation has been preformed.
If we have an a posterior error estimator for a particular problem, a simulation can been performed, the error estimated in various part of the domain and the mesh and finite element type can be adjusted in order to reduce the error to a prescribed value. This is known as *adaptivity*.
The two obvious strategies are known as:

- *h-adaptivity*: this involves modify the mesh.
- *p-adaptivity*: this involves changing the finite element type.

A *posterior error* estimation involves quantification of the error after a simulation has been preformed. With an estimate of the error, a judgement can be made as to the reliability of the result. It is then also possible to adapt the finite element mesh to reduce the error. The goal of adaptivity is minimise the error for a given computational effort. It is a rich area of activity in applied and computational mathematics.

Errors, compute-time, and memory

An elasticity problem is solved on a square domain using a uniform structured mesh of linear triangular elements with n_1 vertices in each direction (n_1 is very large).



Errors, compute-time, and memory

1. Based on a priori estimates, what reduction in the error could you expect? If the total number of elements is doubled and the element order is raised to quadratic,
2. Provide an estimate of the increase in the required computational time if using LU decomposition.
3. How much more computer memory is required to store the stiffness matrix? (Give the factor increase.)

Decrease in error

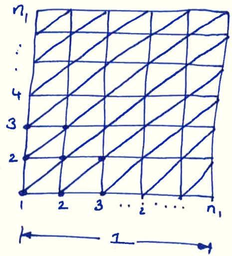
of vertices in each direction $= n_1$

For a large mesh:

# of vertices (nodes) $= n_1^2$

# of elements $\approx 2 n_1^2$ (large mesh)

Element Size $h_1 = \dfrac{1}{n_1}$



If #. of elements is doubled

$$2(2 n_1^2) = 2 n_2^2$$

n_2 – New #. of nodes.

$$n_2 = \sqrt{2}\, n_1$$

New element Size $h_2 = \dfrac{1}{n_2} = \dfrac{1}{\sqrt{2}\, n_1}$

Linear elements exhibit $O(h^2)$ convergence in the displacement norm. $(1/\sqrt{2})^2 = 1/2$

$\therefore$ Error will be halved.

Computational time

b.

Total # Nodes = # vertices + # faces nodes

vertics $= n_2^2$

for each vertex nodes ⇒ 3 facesnodes

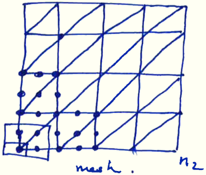
face nodes $= 3n_2^2$

Nodes $= n_2^2 + 3n_2^2$

$\Rightarrow 4n_2^2 . = 4(\sqrt{2}n_1)^2 = 8n_1^2$

For LU decomposition, time scale $O(n^3)$

$\therefore$ Computational time increases by a factor $(8^3) = 512$



Non-zero elements in the sparse stiffness matrix

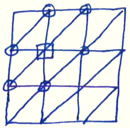
of Non zero terms in stiffness matrix is equal to

$$\sum_{i=1}^{\#\text{Nodes}} C_i \quad , \; C_i = \#\text{ of nodes, node '}i\text{' is connected to}$$

For linear elements

$$C_i = 7$$

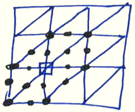
$\therefore$ Non zero entries $= 7n_1^2$



For Quadratic elements:

Vertex nodes: $C_i = 19$

Face/edge nodes $C_i = 9$



Memory increase

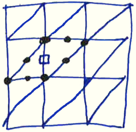
Non zero entries

$$\Rightarrow 19 * \underbrace{n_2^2}_{\#\text{vertex}} + 9 \times \underbrace{(3 * n_2^2)}_{\#\text{facenodes}}$$

$$\Rightarrow 46 n_2^2 = 46(\sqrt{2}\, n_1)^2$$

$$= 92 n_1^2$$

$$\therefore \frac{92}{7} \text{ more memory.}$$



Face