

# Courses/FEM/modules/error

---

## From Icarus

---

< Courses | FEM

### Contents

---

- 1 Error estimation
  - 1.1 Precondition
  - 1.2 Theory
    - 1.2.1 Introduction
    - 1.2.2 Error estimation
    - 1.2.3 A priori estimation in energy norm
    - 1.2.4 A posteriori estimation in energy norm
    - 1.2.5 General quantity a posteriori error estimation / Duality
      - 1.2.5.1 Quantity of interest
      - 1.2.5.2 Duality
      - 1.2.5.3 A priori estimation
      - 1.2.5.4 A posteriori estimation
  - 1.3 Software
  - 1.4 Postcondition
  - 1.5 Exercises
  - 1.6 Examination
  - 1.7 [TODO]

## Error estimation

---

### Precondition

---

- Function approximation
- Galerkin's method

### Theory

---

#### Introduction

When conducting scientific experiments in a laboratory or building a suspension

bridge, for example, there is always a lot of worry about the errors in the process. In fact, if we were to summarize the philosophy behind the scientific revolution, a main component would be the modern emphasis on the quantitative analysis of error in measurements during experiments and the reporting of the errors along with the results. The same issue comes up in computational mathematical modeling: whenever we make a computation on a practical problem, we must be concerned with the accuracy of the results and the related issue of how to compute efficiently. These issues naturally fit into a wider framework which also addresses how well the differential equation models the underlying physical situation and what effect errors in data and the model have on the conclusions we can draw from the results.

We address these issues by deriving two kinds of error estimates for the error  $\mathbf{e} = \mathbf{u} - \mathbf{U}$  of the finite element approximation. First we prove an *a priori* error estimate which shows that the Galerkin finite element method for Poisson's equation produces the best possible approximation of the solution  $\mathbf{u}$  in  $V_h$  in a certain sense. If  $\mathbf{u}$  has continuous second derivatives, then we know that  $V_h$  contains good approximations of  $\mathbf{u}$ , for example the piecewise linear interpolant. So the *a priori* estimate implies that the error of the finite element approximation can be made arbitrarily small by refining the mesh provided that the solution  $\mathbf{u}$  is sufficiently smooth to allow the interpolation error to go to zero as the mesh is refined. This kind of result is called an *a priori* error estimate because the error bound does not depend on the approximate solution to be computed. On the other, it does require knowledge about the derivatives of the (unknown) exact solution.

After that, we prove an *a posteriori* error bound that bounds the error of the finite element approximation in terms of its residual error. This error bound can be evaluated once the finite element solution has been computed and used to estimate the error. Through the *a posteriori* error estimate, it is possible to estimate and adaptively control the finite element error to a desired tolerance level by suitably refining the mesh.

## Error estimation

Examples of structure of error estimates:

*A priori* example (order of convergence):

$$\|\mathbf{e}\|_{\mathbf{E}} \leq C \|h \mathbf{u}''\|$$

*A posteriori* example (actual computable bound on error):

$$\|\mathbf{e}\|_{\mathbf{E}} \leq C \|h \mathbf{R}(\mathbf{U})\|$$

For a linear PDE we observe:

$$\begin{aligned} a(u, v) - L(v) &= 0 \\ a(U, v) - L(v) &= 0 \Rightarrow \\ a(u - U, v) &= 0, \quad \forall v \in V_h \end{aligned}$$

If  $a$  is symmetric we can define the "energy" inner product/norm:

$$(f, g)_E = a(f, g) \Rightarrow \|f\|_E = \sqrt{a(f, f)}$$

For the equation:

$$\begin{aligned} R(u) = u'' - f &= 0, \quad x \in (0, 1) \Rightarrow \\ (f, g)_E &= \int_0^1 f' g' dx \Rightarrow \|f\|_E = \int_0^1 (f')^2 dx = \|f'\| \end{aligned}$$

where we use the notation:

Energy norm

$$\|f\|_E$$

L2 norm

$$\|f\| = \|f\|_{L_2}$$

## A priori estimation in energy norm

(Recall estimate for L2 projection)

$$\begin{aligned} \|e\|_E^2 &= (e, e)_E = (u - U, u - U)_E = \\ &= (u - U, u - U)_E + (u - U, v - v)_E = \\ &= (u - U, u - v)_E + (u - U, v - U)_E = \text{This proves that there is no better} \\ &= (u - U, u - v)_E \leq \|e\|_E \|u - v\|_E \Rightarrow \end{aligned}$$

$$\|e\|_E \leq \|u - v\|_E, \quad \forall v \in V_h$$

approximation than  $U$  in  $V_h$  in the energy norm (if we can define the energy norm).

Continuing, remembering that interpolant  $\pi u \in V_h$  and using interpolation estimate  $\|u - \pi u\|_E \leq Ch \|u'\|_E$ :

$$\|e\|_E \leq \|u - v\|_E, \quad \forall v \in V_h \Rightarrow$$

$$\|e\|_E \leq \|u - \pi u\|_E \leq Ch \|u'\|_E$$

Which means that the energy norm (in this case derivative) of the error converges to zero with first order rate.

## A posteriori estimation in energy norm

We want to extract  $R(U)$  from expression with  $e$ .

Observe that  $\int_0^1 e'w' dx = \int_0^1 U'w' - fw dx$ .

Galerkin orthogonality:  $\int_0^1 U'v' - fv dx = 0, \quad \forall v \in V_h$  Note that  $\pi e \in V_h$

$$\begin{aligned} \|e\|_E^2 &= \int_0^1 e'e' dx = \int_0^1 (U'e' - fe) dx = \\ &\int_0^1 (U'e' - fe - (U'\pi e' - f\pi e)) dx = \end{aligned}$$

Continuing, using integration by parts on each cell/interval

$K_i = [a_i, b_i], 0 < a_i < b_i < 1$  and that the interpolation error is zero in the nodes:  $(e - \pi e)(x_j) = 0$ .

$$\begin{aligned} \sum_{i=1}^M \int_{a_i}^{b_i} U'(e - \pi e)' dx - \int_0^1 f(e - \pi e) dx = \\ \sum_{i=1}^M \int_{a_i}^{b_i} (-U''(e - \pi e) - f(e - \pi e)) dx + [U'(e - \pi e)]_{a_i}^{b_i} = \end{aligned}$$

Clean up, defining discontinuous  $\hat{R}(U) = -U'' - f$

$$\|e\|_E^2 = \int_0^1 \hat{R}(U)(e - \pi e) dx$$

Continuing using Cauchy-Schwartz and interpolation estimate

$$\|e - \pi e\| \leq Ch \|e'\| = Ch \|e\|_E$$

$$\|e\|_E^2 = \int_0^1 \hat{R}(U)(e - \pi e) dx \leq \|\hat{R}(U)\| \|e - \pi e\| \leq \|\hat{R}(U)\| Ch \|e\|_E$$

Which gives the final estimate/bound:

$$\|e\|_E \leq C \|h\hat{R}(U)\|$$

Note that the right hand side is computable given a discrete solution  $U$ .

## General quantity a posteriori error estimation / Duality

## Quantity of interest

We've seen error estimates in energy norm: allows a posteriori error control of a fixed global norm, i.e.  $\|\epsilon\|_E < TOL$ .

We are typically interested in a specific quantity of interest:

$$M(\epsilon) = (\epsilon, \psi) = \int_{\Omega} \epsilon \psi dx$$

We can also call  $M$  a functional of the error (takes a function and gives a scalar).

Examples:

Integral  $\psi = 1$

Gives the integral of the error over the whole domain:  $(\epsilon, \psi) = \int_{\Omega} \epsilon dx$

Average  $\psi = \chi_{\omega}/|\omega|$ ,  $\omega \subset \Omega$ ,  $\chi_{\omega} = 1, x \in \omega$ ,  $\chi_{\omega} = 0, x \notin \omega$

Gives average error in subdomain  $\omega$  (close to an object for example)

$L_2$  norm  $\psi = \epsilon$

Gives (square of the)  $L_2$  norm of the error in the whole

domain:  $(\epsilon, \psi) = (\epsilon, \epsilon) = \|\epsilon\|^2$ . Obviously we don't know  $\epsilon$ , but we may be able to use further estimates to avoid this (like for Poisson's equation).

More..

We can define  $\psi$  to evaluate the error at a point, or a derivative of the error, and other options.

## Duality

We have *primal equation*:

$$-u'' - f = 0$$

We introduce *dual equation* with the sought-after quantity  $\psi$  as source/data:

$$-\phi'' - \psi = 0$$

where we assume homogenous Dirichlet boundary conditions for simplicity.

The dual differential operator  $A^*w = -w''$  is defined by:

$$(Av, w) = (v, A^*w)$$

where  $A$  is the differential operator for the primal equation (in this case they

turn out to be the same). The dual operator  $\mathbf{A}^*$  can be constructed by repeated use of integration by parts.

### A priori estimation

(See the CDE book for steps of proof)

$$\|e\| \leq Ch^2 S \|u''\|$$

### A posteriori estimation

We are looking for an a posteriori estimate of the form:

$$|(e, \psi)| \leq Ch^q \|\hat{R}(U)\| \dots$$

- We compute solution  $U$  by Galerkin's method (which gives us the Galerkin Orthogonality):

$$\int_0^1 U'v' - fvd x = 0, \quad \forall v \in V_h, \quad U \in V_h$$

- We observe that the error  $e = u - U$  satisfies:

$$\int_0^1 e'w' dx = \int_0^1 -U'w' + fw dx$$

We want to bound quantity  $(e, \psi)$ :

$$\begin{aligned} (e, \psi) &= \int_0^1 e\psi dx = \int_0^1 e(-\phi'') dx = \\ &= \int_0^1 e'\phi' dx + [e\phi']_0^1 = \int_0^1 -U'\phi' + f\phi dx = \end{aligned}$$

We use the Galerkin orthogonality to add the interpolant of  $\phi$  :  $\pi\phi \in V_h$ , in other words adding zero.

$$= \int_0^1 -U'(\phi - \pi\phi)' + f(\phi - \pi\phi) dx =$$

We now want to move away the derivative on  $(\phi - \pi\phi)'$  with integration by parts. We split up the integral on each cell (interval in 1D)  $K_i$  and thus get boundary contributions on the cell interval boundary (end points  $a_i$  and  $b_i$ ), which again are zero.

$$= \sum_{K_i} \int_{a_i}^{b_i} (U'' + f)(\phi - \pi\phi) dx + [U'(\phi - \pi\phi)]_{a_i}^{b_i} = \sum_{K_i} \int_{a_i}^{b_i} \hat{R}(U)(\phi - \pi\phi) dx$$

Use Cauchy-Schwartz as before:

$$\begin{aligned} |(e, \psi)| &\leq \|\hat{R}(U)\| \|\phi - \pi\phi\| \\ &\leq Ch^2 \|\hat{R}(U)\| \|\phi''\| \end{aligned}$$

which is our estimate and where stability factor  $S = \|\phi''\|$  gives information about how the residual grows.

We are left with two options for estimating/computing the stability factor:

- Try to estimate  $\phi''$  analytically
- Compute discretization of  $\phi$  with FEM (solve dual equation)

In our model equation (Poisson in 1D) we have  $\|\phi''\| = \|\psi\|$  where  $\psi$  is known data. In the general case we typically have to discretize  $\phi$ .

## Software

---

We can implement norms and quantities of known function in the FEniCS form language, since they are just integrals over the domain. To implement the energy norm of  $U$  for Poisson's equation we can write:

```
U = Function(element, mesh, x)
M = (dot(grad(U), grad(U)) * dx)
energy_norm = sqrt(assemble(M, mesh))
```

Uniform mesh refinement is available in the Mesh function `refine()`:

```
mesh = Mesh("mesh.xml")
mesh.refine()
```

which halves the the mesh size  $h$  by placing a new vertex on each edge, thus splitting each triangle into four new.

## Postcondition

---

You should now be familiar with:

- Definition of computational error:  $e = u - U$

- A priori error estimate of a linear differential equation in the energy norm
- A posteriori error estimate of a linear differential equation in the energy norm
- A posteriori error estimate of a linear differential equation of a quantity (linear functional)  $\psi$  of the error

## Exercises

---

CDE: 9.23, 9.24, 9.25, 15.48

## Examination

---

1.1

Consider Poisson's equation:

$$-\Delta u = f, \quad x \in \Omega$$

with homogenous Dirichlet boundary conditions on the unit square.

With  $f(x) = 32x_1(1 - x_1) + 32x_2(1 - x_2)$  we have the exact solution  $u(x) = 16x_1(1 - x_1)x_2(1 - x_2)$ .

Investigate how the error decreases (if it does, and the rate) with uniform mesh refinement given the initial mesh <http://www.icarusmath.com/icarus/images/Square.xml> in:

- the energy norm
- the average of the error:  $(e, 1)$  on a unit size domain.
- the  $L_2$  norm

Do your results match the rates (powers of  $h$ ) from the a priori error estimates for Poisson's equation?

NB: you need to represent  $f$  and  $e$  as quadratic functions (to integrate to quadratic order) by using a quadratic element:

```

element2 = FiniteElement("Lagrange", "triangle", 2)
f = Source(element2, mesh)
exact = Exact(element2, mesh)

```

The first steps of the a posteriori estimation in 2D looks just like 1D:

$$(e, \psi) = \int_{\Omega} e \psi dx = \int_{\Omega} \nabla e \cdot \nabla \phi dx = \int_{\Omega} \nabla U \cdot \nabla \phi - f \phi dx$$

We can stop here, since  $\phi$  is computable and  $U$  and  $f$  are known.

Solve the dual problem with  $\psi = 1$  and verify that  $(e, \psi)$  and  $\int_{\Omega} \nabla U \cdot \nabla \phi - f \phi dx$  are (almost) equal (we can check since we have the exact solution in this specific case).

You have to solve the dual problem with a higher order finite element. To use quadratic basis functions you can write:

```
element_quadratic = FiniteElement("Lagrange", "triangle", 2)
```

## 1.2

Consider the stationary convection-diffusion equation (primal equation):

$$-\epsilon \Delta u + \beta \cdot \nabla u + \alpha u = f, \quad x \in \Omega$$

with homogenous Neumann boundary conditions on the unit square.

The dual equation is then:

$$-\epsilon \Delta \phi - \beta \cdot \nabla \phi + \alpha \phi = \psi, \quad x \in \Omega$$

with homogenous Neumann boundary conditions on the unit square.

Compute the primal and dual solution with:

$$\alpha = 0.1$$

$$\epsilon = 0.01$$

$$f = 1, \quad |x - (0.75, 0.5)| < 0.1 \quad f = 0 \text{ otherwise}$$

$$\psi = 1, \quad |x - (0.25, 0.25)| < 0.1 \quad \psi = 0 \text{ otherwise}$$

$$\beta = (\beta_0, \beta_1) \quad \beta_0 = 5(-x_2 + 0.5) \quad \beta_1 = 5(x_1 - 0.5)$$

Plot the primal and dual solutions. Give an interpretation with respect to the error  $e$  in the subdomain given by  $\psi$ , considering the error expressed in terms of the discrete solution  $U$  and the dual solution  $\phi$  (you can think of the error as:  $(e, \psi) = (R(U), \phi)$ ). In other words: where would you expect the error contribution to be large looking at the primal and dual solutions?

You can define a vector-valued function ( $\beta$ ) like this:

```
# Convection velocity term
class Velocity(Function):
    def __init__(self, element, mesh):
        Function.__init__(self, element, mesh)
    def eval(self, values, x):
        values[0] = 5.0*(-(x[1] - 0.5))
        values[1] = 5.0*(x[0] - 0.5)
    def rank(self):
        return 1
    def dim(self, i):
        return 2

velement = VectorElement("Lagrange", "triangle", 1)
beta = Velocity(velement, mesh)
```

## **[TODO]**

---

Retrieved from "<http://www.icarusmath.com/icarus/index.php?title=Courses/FEM/modules/error>"

- This page was last modified 12:27, 3 October 2008.