Finite Element Approximation I
MATH46052|66052 *

David J. Silvester
School of Mathematics, University of Manchester

Version 1.2 | 30 January 2019

Contents

*These notes provide a summary of the essential material from the lectures. They are by no means comprehensive and should be augmented by notes taken during the lectures.

1. Smoothness of functions This section reviews some important concepts that will be needed to understand the material in subsequent sections of the notes.

(Function)
A **function** maps one or more inputs (real numbers) to a *unique* output number; for example, $y = f(x)$ or $z = f(x, y)$ or $z = f(x_1, x_2, \ldots, x_n) = f(\vec{x})$, where $\vec{x}$ is the n–dimensional vector $(x_1, x_2, \ldots, x_n)$. The *domain* of a function is the set Ω of all possible input values. For a real-valued function the mapping is conveniently expressed by writing $f : \Omega \to \mathbb{R}$.

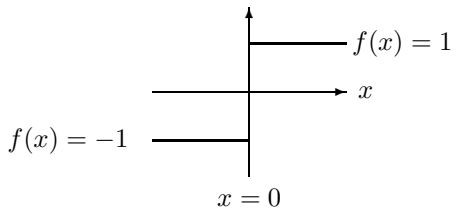
Functions are characterised by their *smoothness*. The simplest example of a smooth function of one variable is one that does *not jump* when plotted as a graph. Expressed mathematically we have the following characterization.

(Continuous function)
A univariate function f is **continuous** when

$$\lim_{\epsilon \to 0} f(x - \epsilon) = f(x) = \lim_{\epsilon \to 0} f(x + \epsilon)$$

for *all* possible values of x in Ω. In this case we write $f \in C^0(\Omega)$.

A simple example of a one-dimensional function that is not continuous ...



An alternative way of measuring the smoothness of a function is to consider their integrability. The discontinuous function above is *square integrable*.

(Square integrable function)
An real-valued function of n variables $f : \Omega \to \mathbb{R}$ is **square integrable** if and only if

$$\int_\Omega \{f(x_1, x_2, \ldots, x_n)\}^2 \, \mathrm{d}x_1 \, \mathrm{d}x_2 \ldots \mathrm{d}x_n < \infty.$$

For the example above we have $\int_\Omega f^2 = \int_{-1}^0 (-1)^2 \, \mathrm{d}x + \int_0^1 (1)^2 \, \mathrm{d}x = 2 < \infty$ and we write $f \in L^2(\Omega)$.

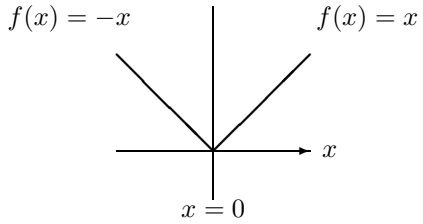
A stronger definition of a smooth univariate function arises when we insist that the two tangents at a given point x are both finite and do *not jump*. Such a function is said to be *differentiable*.

(Differentiable function)
A *continuous* function f is **differentiable** when

$$\lim_{\epsilon \to 0} \frac{f(x) - f(x-\epsilon)}{\epsilon} = f'(x) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}$$

for *all* possible values of x in Ω. If a differentiable function has a derivative function that is continuous then we write $f \in C^1(\Omega)$.

A simple example of a continuous function that is not differentiable ...



Differentiable functions with more than one input variable have more than one tangent at every point $\vec{x}$ and are associated with partial derivatives.

(Partial derivative)
A function of two variables f has a **partial derivative** whenever

$$\lim_{\epsilon \to 0} \frac{f(x,y) - f(x-\epsilon,y)}{\epsilon} = \frac{\partial f}{\partial x} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon,y) - f(x,y)}{\epsilon}$$

It represents the *rate of change* with respect to the first variable when the second variable is kept fixed. Similarly, we write

$$\lim_{\epsilon \to 0} \frac{f(x,y) - f(x,y-\epsilon)}{\epsilon} = \frac{\partial f}{\partial y} = \lim_{\epsilon \to 0} \frac{f(x,y+\epsilon) - f(x,y)}{\epsilon}$$

(Gradient)
When $\vec{x}$ is the n-component vector $(x_1, x_2, \ldots, x_n)$, the **gradient** of a multivariate function f is the *vector* of partial derivatives (usually written as a column vector) given by

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

The nondifferentiable function pictured above is smooth in the sense that it is square integrable over $[-1, 1]$ and it has a square integrable (weak) derivative.

(Square integrable derivative)
An multivariate function $f : \Omega \to \mathbb{R}$ has a **square integrable derivative** if and only if

$$\int_\Omega \nabla f \cdot \nabla f < \infty.$$

In this case we write $\nabla f \in (L^2(\Omega))^n$.

Functions that are square integrable with a square integrable derivative are members of a special function space.

(Sobolev space)
The **Sobolev space** $H^1(\Omega)$ consists of functions that are square integrable and have a square integrable derivative:

$$H^1(\Omega) = \{v | v \in L^2(\Omega), \nabla v \in (L^2(\Omega))^n\}.$$

The above definition can be extended to *higher-order* Sobolev spaces; for example, for a positive index k, the Sobolev space $H^k(0, 1)$ is the set of functions $v : (0, 1) \to \mathbb{R}$ such that v and all (weak-)derivatives of order up to and including k are square integrable:

$$u \in H^k(0,1) \iff \int_0^1 u^2 \,\mathrm{d}x < \infty,\ \int_0^1 (u')^2 \,\mathrm{d}x < \infty,\ \ldots,\ \int_0^1 (u^{(k)})^2 \,\mathrm{d}x < \infty.$$

One point that needs to be emphasised is that functions that have a square integrable derivative need not be differentiable in the classical sense. The connection between continuously differentiable functions and square integrable functions is given by Sobolev's *embedding* theorem.[1] In one dimension the functions of $H^1(\Omega)$ are continuous, whereas in dimension 2 or 3 this may not be the case. If $d = 2$ or 3, functions in the space $H^2(\Omega)$ are continuous.

The function spaces $L^2(\Omega)$ and $H^k(\Omega)$ are pretty special—they have (or more formally, they are "equipped with") an *inner product*.

(Inner product space)
An inner product space V is a vector space over the field of real numbers $\mathbb{R}$ which has a mapping $(\cdot, \cdot) : V \times V \to \mathbb{R}$ that satisfies four axioms:

① $(u, w) = (w, u) \qquad \forall\, u, w \in V$ (where $\forall$ means "for all");

[1] A detailed discussion of the underlying theory can be found in Robinson [3, Section 5.7].

② $(u,u) \geq 0 \qquad \forall\, u \in V;$

③ $(u,u) = 0 \iff u = 0 \qquad$ (where $\iff$ means "if and only if");

④ $(\alpha u + \beta v, w) = \alpha(u,w) + \beta(v,w) \qquad \forall\, \alpha, \beta \in \mathbb{R}; \quad \forall\, u, v, w \in V.$

The inner product allows us to extend the notion of *orthogonality*, a geometric property, to function spaces. The following examples of inner product spaces are important:

- $V = \mathbb{R}^2$, that is, all vectors $\boldsymbol{u} = \left[\begin{smallmatrix} u_1 \\ u_2 \end{smallmatrix}\right]$, with inner product

$$(\boldsymbol{u}, \boldsymbol{w}) = u_1 w_1 + u_2 w_2 \ = \boldsymbol{u} \cdot \boldsymbol{w}$$

- $V = L^2(\Omega)$, with inner product

$$(u,w) = \int_\Omega uw$$

- $V = H^1(\Omega)$, with inner product

$$(u,w) = \int_\Omega uw + \int_\Omega \nabla u \cdot \nabla w$$

- $V = H^2(0,1)$, with inner product

$$(u,w) = \int_0^1 uw \,\mathrm{d}x + \int_0^1 u'w' \,\mathrm{d}x + \int_0^1 u''w'' \,\mathrm{d}x$$

An inner product space is also a *normed space*.

(Normed vector space)
A normed vector space V over the field of real numbers $\mathbb{R}$ has (or more formally is "equipped with") a mapping $\|\cdot\| : V \to \mathbb{R}$ that satisfies four axioms:

① $\|u\| \geq 0 \qquad \forall\, u \in V;$

② $\|u\| = 0 \iff u = 0;$

③ $\|\alpha u\| = |\alpha| \|u\| \quad \forall\, \alpha \in \mathbb{R}$ and $\forall\, u \in V;$

④ $\|u + v\| \leq \|u\| + \|v\| \quad \forall\, u, v \in V.$

The norm extends the notion of *length*, a geometric property, to function spaces. Every inner product space V has a natural (or "energy") norm defined so that

$$\|u\| = (u,u)^{1/2}.$$

The following examples of normed spaces are particularly important:

- $V = \mathbb{R}^2$, with norms

$$\begin{aligned} \|\boldsymbol{u}\|_1 &= |u_1| + |u_2|, && \ell_1 \text{ norm} \\ \|\boldsymbol{u}\|_2 &= ({u_1}^2 + {u_2}^2)^{1/2}, && \ell_2 \text{ norm} \\ \|\boldsymbol{u}\|_\infty &= \max\{|u_1|, |u_2|\}. && \ell_\infty \text{ norm} \end{aligned}$$

- $V = \mathbb{R}^{n\times m}$, with norms

$$\begin{aligned} \|A\|_1 &= \max_{j=1,\ldots,m} \Big\{ \sum_{i=1}^{n} |a_{ij}| \Big\}, && \ell_1 \text{ matrix norm} \\ \|A\|_\infty &= \max_{i=1,\ldots,n} \Big\{ \sum_{j=1}^{m} |a_{ij}| \Big\}. && \ell_\infty \text{ matrix norm} \end{aligned}$$

- $V = C^0(\overline{\Omega})$ with norm

$$\|u\|_{L^\infty(\Omega)} = \max_{x\in\overline{\Omega}} |u(x)|. \qquad L_\infty \text{ norm}$$

- $V = L^2(\Omega)$ with norm

$$\|u\|_{L^2(\Omega)} = \left\{ \int_\Omega u^2 \right\}^{1/2}. \qquad L_2 \text{ norm}$$

Inner products and norms are related by the *Cauchy–Schwarz* inequality

(Cauchy–Schwarz inequality)

$$|(u, v)| \le \|u\| \, \|v\| \qquad \forall\, u, v \in V.$$

The following examples of the inequality are important:

- $V = \mathbb{R}^2$ gives the discrete Cauchy–Schwarz (C–S) inequality:

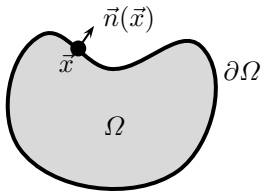
$$\boldsymbol{u} \cdot \boldsymbol{w} \le |\boldsymbol{u} \cdot \boldsymbol{w}| \le ({u_1}^2 + {u_2}^2)^{1/2} ({w_1}^2 + {w_2}^2)^{1/2}.$$

- $V = L^2(\Omega)$ gives C–S for definite integrals:

$$\int_\Omega uw \le \left| \int_\Omega uw \right| \le \left(\int_\Omega u^2 \right)^{1/2} \left(\int_\Omega w^2 \right)^{1/2}.$$

2. A model PDE problem The finite element method is a classical procedure that can be used to generate accurate numerical solutions of boundary value problems.[2] The simplest example of such a problem is associated with the Poisson equation posed over a domain $\Omega \subset \mathbb{R}^n$ where $n = 2$ or $n = 3$.

The classical problem we will study is the following: given a smooth forcing function ($f \in C^0(\Omega)$) and smooth boundary data $g \in C^0(\partial\Omega)$, compute (or approximate) the function $u : \Omega \to \mathbb{R}$ satisfying



$$-\nabla^2 u = f \quad \text{in } \Omega, \tag{2.1}$$

together with *mixed* boundary conditions

$$u = g_D \quad \text{on } \partial\Omega_D \qquad \text{(Dirichlet)}, \tag{2.2a}$$

$$\frac{\partial u}{\partial n} = g_N \quad \text{on } \partial\Omega_N \qquad \text{(Neumann)}. \tag{2.2b}$$

Note that we tacitly assume that the boundary can be broken into disjoint pieces so that $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ and $\partial\Omega_D \cap \partial\Omega_N = \emptyset$.

(Classical solution)
A classical solution u satisfying (2.1) and (2.2) is a solution that is suitably smooth. In the pure Dirichlet case ($\partial\Omega = \partial\Omega_D$) we require that $u \in C^2(\Omega)$ and that u is continuous up to the boundary (so that (2.2a) can be satisfied).

The starting point for finite element analysis is the concept of a *weak solution*. This is discussed next. The starting point is to introduce a suitable space of "test functions", say X, and to make the PDE *residual* orthogonal (in an $L^2(\Omega)$ sense) to all functions in this space, that is, we enforce

$$\int_\Omega \{\nabla^2 u + f\}\, v = 0 \quad \forall v \in X. \tag{2.3}$$

The function u satisfying (2.3) is called a *strong solution*. A neat strategy is to reduce the differentiability requirements on u by transferring derivatives onto the test function v using integration by parts:

$$\begin{aligned}\int_\Omega \nabla^2 u\, v = \int_\Omega (\nabla \cdot \nabla u)\, v &= -\int_\Omega \nabla v \cdot \nabla u + \int_{\partial\Omega} v \nabla u \cdot \vec{n}\, \mathrm{d}s \\ &= -\int_\Omega \nabla u \cdot \nabla v + \int_{\partial\Omega} v \frac{\partial u}{\partial n}\, \mathrm{d}s.\end{aligned}$$

[2] A *boundary value problem* is a PDE (or a system of PDEs) together with appropriate boundary and initial conditions.

Substituting this expression into (2.3) and integrating over the boundary sections seperately gives

$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega fv + \int_{\partial\Omega} v\frac{\partial u}{\partial n}\,\mathrm{d}s,$$
$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega fv + \int_{\partial\Omega_D} v\frac{\partial u}{\partial n}\,\mathrm{d}s + \int_{\partial\Omega_N} vg_N\,\mathrm{d}s.$$

We then simplify the formulation (remove the first boundary integral) by insisting that the test functions v be zero on the Dirchlet part of the boundary. This gives a *weak formulation* of the original boundary value problem

$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega fv + \int_{\partial\Omega_N} g_N v\,\mathrm{d}s, \tag{2.4}$$

which is required to hold for all functions v in the test space X. The specification of the weak problem is completed by identifying the largest space X for which the integrals over Ω in (2.4) remain finite. Using the Cauchy–Schwarz inequality we can show that this is precisely the space of functions that are square integrable and which have a square integrable first derivative. This immediately leads to the identification

$$X = \left\{v \,|\, v \in L^2(\Omega), \nabla v \in (L^2(\Omega))^n;\ \ v = 0 \text{ on } \partial\Omega_D\right\}. \tag{2.5}$$

(Weak solution)
A weak solution u satisfies the *essential* boundary condition (2.2a) and is the particular member of the *solution space*

$$X_E = \left\{v \,|\, v \in L^2(\Omega), \nabla v \in (L^2(\Omega))^n;\ \ v = g_D \text{ on } \partial\Omega_D\right\},$$

that satisfies the weak formulation (2.4) for all functions in the associated test space X. Note that the weak solution will satisfy the *natural* boundary condition (2.2b) automatically. Observe that X_E is not a vector space (the sum of two functions lies outside the space) if the Dirichlet boundary condition is nonhomogeneous.

What is the connection between classical and weak solutions? By construction, any function u that satisfies (2.1) and (2.2) must also be a weak solution. Showing that a smooth enough weak solution is also a classical solution requires Sobolev's *embedding* theorem and a lot of functional analysis. A clear exposition can be found in the book by Robinson [3, Chapter 6].

The uniqueness of the weak solution can be established by contradiction, making use of the Poincaré–Friedrichs inequality.[3] This inequality bounds the

[3] A detailed discussion of the P–F inequality can be found in Braess [1, pp. 30–31].

norm of a test function by the norm of its (weak) derivative and a characteristic length scale L (in two dimensions, L is the length of the side of the smallest square that contains Ω),

$$\|u\|_{L^2(\Omega)} \leq L\|\nabla u\|_{L^2(\Omega)}, \quad u \in X, \tag{2.6}$$

and involves the additional requirement that $\int_{\partial\Omega_D} \mathrm{d}s \neq 0$. (Otherwise, we have $\partial\Omega = \partial\Omega_N$, which is the so-called pure Neumann problem.)

One big advantage of working with the weak formulation (2.4) is that it is well defined (all integrals are finite) for discontinuous forcing data f and rough boundary data g. This can be seen by applying the Cauchy–Schwarz inequality to the integrals on the right-hand side of (2.4),

$$\int_\Omega fv + \int_{\partial\Omega_N} g_N v \, \mathrm{d}s \leq \|f\|_{L^2(\Omega)} \underbrace{\|v\|_{L^2(\Omega)}}_{<\infty} + \|g_N\|_{L^2(\partial\Omega_N)} \underbrace{\|v\|_{L^2(\partial\Omega_N)}}_{<\infty}, \tag{2.7}$$

where the boundedness of the second term follows from extending v by zero to the whole of the boundary and then using a *trace inequality*[4] to relate its norm to that of a smooth extension that is defined inside the domain

$$\|v\|_{L^2(\partial\Omega_N)} = \|v\|_{L^2(\partial\Omega)} \leq C \underbrace{\|v\|_{H^1(\Omega)}}_{<\infty}.$$

The implication of (2.7) is that the weak formulation (2.4) is valid for (discontinuous) forcing data $f \in L^2(\Omega)$ and for boundary data $g_N \in L^2(\partial\Omega_N)$. This is a much larger class of functions than the data specification that is required when solving the classical problem.

3. Galerkin approximation The weak formulation is not tractable because the test space X is infinite-dimensional. A good way of generating a computational approximation to the weak solution is to find the best approximation to u from a finite-dimensional subspace of the solution space X_E. The resulting function u_h is called the *Galerkin solution*.

(Galerkin solution)
A Galerkin solution (denoted u_h) satisfies the *essential* boundary condition (2.2a) exactly. It is the particular member of a *finite-dimensional* approximation space $X_E^h \subset X_E$ that satisfies the weak formulation (2.4) for all functions in the associated approximation test space $X^h \subset X$. Note that the Galerkin solution will not satisfy the *natural* boundary condition (2.2b) in general.

[4]Relevant trace theorems are stated (and proved) in Braess [1, pp. 44–51].

Stated formally, the Galerkin solution is the function $u_h \in X_E^h \subset X_E$ that satisfies a finite-dimensional version of (2.4):

$$\int_\Omega \nabla u_h \cdot \nabla v_h = \int_\Omega f v_h + \int_{\partial\Omega_N} g_N v_h \, \mathrm{d}s \quad \forall\, v_h \in X^h. \tag{3.8}$$

To show that u_h is the *best* approximation of u, we pick an arbitrary test function $v = z_h$ in (2.4) and subtract (3.8) with $v_h = z_h$ to give $\int_\Omega \nabla u \cdot \nabla z_h - \int_\Omega \nabla u_h \cdot \nabla z_h = 0$. That is,

$$\int_\Omega \nabla(u - u_h) \cdot \nabla z_h = 0 \quad \forall\, z_h \in X^h. \tag{3.9}$$

This result is usually referred to as *Galerkin orthogonality*: the error function $e = u - u_h$ is H^1 orthogonal to the test subspace X^h. We can exploit this orthogonality by considering a specific measure (norm) of the error, namely

$$\|u - u_h\|_E = \|\nabla u - \nabla u_h\|_{L^2(\Omega)}. \tag{3.10}$$

In particular,

$$\begin{aligned}
\|u - u_h\|_E^2 = \|\nabla(u - u_h)\|_{L^2(\Omega)}^2 &= \int_\Omega \nabla(u - u_h) \cdot \nabla(u - u_h) \\
&= \int_\Omega \nabla(u - u_h) \cdot \nabla(u - v_h + v_h - u_h) \qquad (\text{for any } v_h \in X_E^h) \\
&= \int_\Omega \nabla(u - u_h) \cdot \nabla(u - v_h) + \int_\Omega \nabla(u - u_h) \cdot \nabla(v_h - u_h) \\
&= \int_\Omega \nabla(u - u_h) \cdot \nabla(u - v_h) \qquad (\text{from (3.9)}, \, v_h - u_h \in X^h).
\end{aligned}$$

Applying Cauchy–Schwarz to the right-hand side then gives

$$\|u - u_h\|_E^2 \le \|\nabla(u - u_h)\|_{L^2(\Omega)} \, \|\nabla(u - v_h)\|_{L^2(\Omega)} \le \|u - u_h\|_E \, \|u - v_h\|_E.$$

Thus, if $\|u - u_h\|_E \neq 0$, the Galerkin solution $u_h \in X_E^h$ satisfying (3.8) is the *best approximation*[5] to the weak solution u when the error is measured in the *energy* norm (3.10),

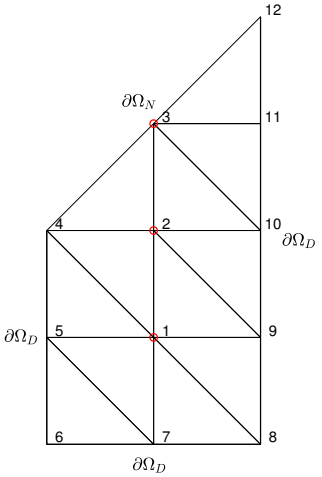
$$\|u - u_h\|_E \le \|u - v_h\|_E \quad \forall\, v_h \in X_E^h. \tag{3.11}$$

A key question needs to be addressed at this point: How easy is it to construct the solution space X_E^h and test space X^h? What make this difficult is the need to construct basis functions that satisfy the essential boundary condition. Globally continuous *piecewise polynomial* approximations are the answer: they provide a simple mechanism for constructing a sequence of increasingly refined approximation spaces $X_E^{h_1} \subset X_E^{h_2} \subset \cdots \subset X_E$ in a computational setting.

[5]Note that we also have a best approximation (3.11) in the case $\|u - u_h\|_E = 0$.

(Finite element approximation)
A finite element approximation (also denoted u_h) to the weak solution u satisfying (2.4) is a Galerkin solution computed using approximation spaces $X_E^h \subset H^1(\Omega)$, $X^h \subset H^1(\Omega)$ that are spanned by C^0 piecewise polynomials of low degree (typically linear or quadratic). The finite element approximation is thus a *best approximation*.

Specific examples of two-dimensional finite element approximations will be presented after discussion of some linear algebra aspects. To enable this, let us consider the generic case of a polygonal domain Ω partitioned into n_k *elements* (for example, triangles in $\mathbb{R}^2$, bricks in $\mathbb{R}^3$) with a total of n interpolation points in the interior of $\Omega \cup \partial\Omega_N$, together with an additional n_∂ interpolation points lying on the Dirichlet boundary $\partial\Omega_D$. As an example, for the mesh illustrated, if the interpolation points are taken to be vertices then we have $n = 3$ (the marked vertices 1, 2 and 3 are *interior*) and $n_\partial = 9$ (the vertices $4, 5, \ldots, 12$ are on $\partial\Omega_D$).



In this example the solution approximation space will be spanned by 12 linear *hat functions*,

$$X_h^E = \text{span}\,\{\phi_j(x,y)\}_{j=1}^{n+n_\partial}\,,$$

satisfying the interpolation conditions

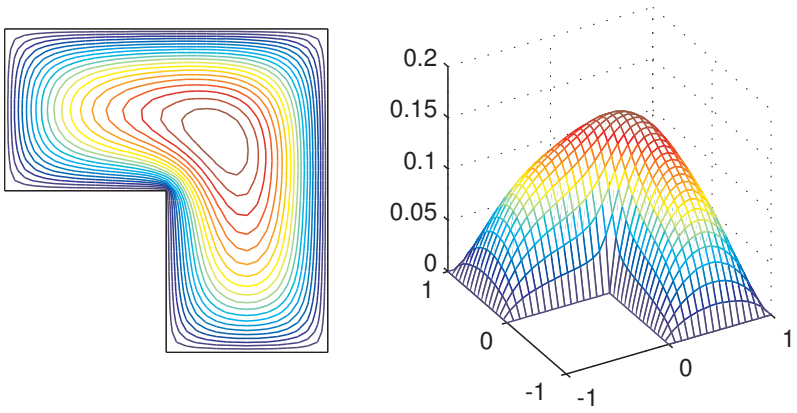
$$\phi_j = \begin{cases} 1 & \text{at vertex } j, \\ 0 & \text{at vertex } i \neq j, \end{cases}$$

and the Galerkin solution is constructed so that

$$u_h(x,y) = \underbrace{\sum_{j=1}^{n} u_j\,\phi_j(x,y)}_{=0 \text{ on } \partial\Omega_D} + \underbrace{\sum_{j=n+1}^{n+n_\partial} \overline{u}_j\,\phi_j(x,y)}_{=g_D \text{ on } \partial\Omega_D}\,. \tag{3.12}$$

In this representation, $u_1, u_2, \ldots, u_n$ are unknown coefficients and $\overline{u}_j$ are fixed values (they represent the value of $g_D(x,y)$ at the jth boundary vertex). An important point is that the function g_D will *not be represented exactly* if it is not itself a piecewise linear function. Otherwise the approximation is *nonconforming*.[6] The quality of the approximation is not compromised in this

[6]The resulting approximation error is referred to a *variational crime* in the book by Strang & Fix [4, Chapter 4].



situation—the essential boundary condition will be approximated increasingly accurately whenever a refinement to the mesh is made. An important observation is that the interior functions $\phi_1, \phi_2, \ldots, \phi_n$ form a basis for the approximation test space X^h (all these basis functions are zero on $\partial\Omega_D$). This means that the Galerkin formulation (3.8) can be written as a system of n equations

$$\int_\Omega \nabla u_h \cdot \nabla \phi_i = \int_\Omega f\phi_i + \int_{\partial\Omega_N} g_N \phi_i \, \mathrm{d}s, \quad i = 1, 2, \ldots, n. \tag{3.13}$$

Substituting (3.12) into (3.13) gives an $n \times n$ linear algebra system

$$A\boldsymbol{u} = \boldsymbol{f}, \tag{3.14}$$

where $\boldsymbol{u}$ is the vector of all the unknown coefficients. We will refer to (3.14) as the *Galerkin system* and A as the *stiffness matrix* $A_{i,j} = \int_\Omega \nabla\phi_j \cdot \nabla\phi_i$. The right-hand-side $\boldsymbol{f}$ is a vector of known values

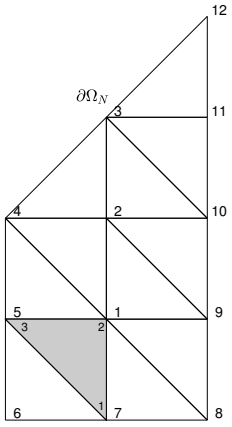
$$\boldsymbol{f}_i = \int_\Omega f\phi_i + \int_{\partial\Omega_N} g_N\phi_i \, \mathrm{d}s - \sum_{j=n+1}^{n+n_\partial} \overline{u}_j \int_\Omega \nabla\phi_j \cdot \nabla\phi_i \tag{3.15}$$

that characterise the problem input data, that is, f, g_D and g_N.

4. A worked example The best way to get a handle on the finite element solution process is to work through a specific problem on paper. The figure above shows a finite element solution of the Poisson equation computed on an L-shaped domain with a constant forcing function ($f = 1$) and a zero Dirichlet condition $u = 0$ imposed everywhere on $\partial\Omega$. What makes the problem challenging is the reentrant corner. The contours of solution height are

increasingly closely packed near the origin. Converting to polar coordinates we find that the radial derivative of the solution u becomes unbounded in the limit $r \to 0$.

To link with the discussion in the previous section we will exploit the inherent symmetry of the solution and solve the PDE over the bottom half of the domain with a zero Neumann condition $\partial u/\partial n = 0$ on the line of symmetry ($y = x$). If we want to compute the solution on the triangular mesh introduced in the previous section then we simply need to construct the Galerkin system (3.14) corresponding to the basis functions at the three interior vertices labelled 1, 2 and 3 in the picture. Note that since we have $g_N = 0$ and $g_D = 0$, the components of the vector $\boldsymbol{f}$ have a relatively simple form: $\boldsymbol{f}_i = \int_\Omega \phi_i$ for $i = 1, 2, 3$.



In general, the finite element solution process has three component parts; element *integrations*, element *assembly* and Galerkin system *solution*. First, element integrations are needed to compute the local contributions to the global Galerkin system. The idea is to construct the 3×3 element *contribution* to the Galerkin matrix and a 3×1 element vector to the right-hand-side vector. If we focus on the specific triangle ② that is highlighted in the picture, then the three localised linear basis functions are given by the *barycentric* coordinates

$$L_1 := \phi_7(x, y)\Big|_{②}, \quad L_2 := \phi_1(x, y)\Big|_{②}, \quad L_3 := \phi_5(x, y)\Big|_{②}.$$

The element contributions are thus given by

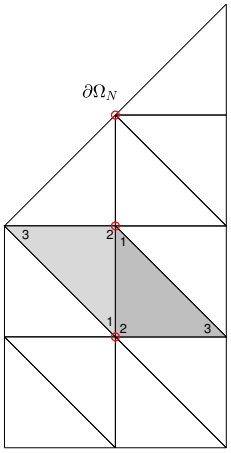
$$A^{②}_{i,j} = \int_{②} \nabla L_j \cdot \nabla L_i = \frac{1}{4|\triangle|} \{b_i b_j + c_i c_j\}, \qquad i = 1, 2, 3,\ j = 1, 2, 3,$$
$$f^{②}_i = \int_{②} L_i = \frac{|\triangle|}{3}, \qquad i = 1, 2, 3,$$

where$|\triangle| = 0.125$ is the area of triangle ② and the b_i's and c_i's are the vertex distances of the local coordinates (shown in the shaded region in the picture) given by

$$b_1 = y_2 - y_3 = 0, \qquad b_2 = y_3 - y_1 = 0.5, \qquad b_3 = y_1 - y_2 = -0.5,$$
$$c_1 = x_3 - x_2 = -0.5, \qquad c_2 = x_1 - x_3 = 0.5, \qquad c_3 = x_2 - x_1 = 0.$$

Working out the explicit matrix entries gives

$$A^{②} = \begin{bmatrix} 1/2 & -1/2 & 0 \\ -1/2 & 1 & -1/2 \\ 0 & -1/2 & 1/2 \end{bmatrix}, \quad f^{②} = \begin{bmatrix} 1/24 \\ 1/24 \\ 1/24 \end{bmatrix}.$$



A feature of our triangular subdivision is that all n_k elements are congruent. Thus, since a consistent numbering convention is adopted, all element matrices have exactly the same form $A^{①} = A^{②} = A^{③} = \cdots$ and since f is constant on the domain, so do the vectors $f^{①} = f^{②} = f^{③} = \cdots$

To illustrate the element *assembly* process we consider the Galerkin matrix entry

$$A_{1,2} = \int_\Omega \nabla\phi_2 \cdot \nabla\phi_1 = \sum_{k=1}^{12} \int_{⑥} \nabla\phi_2 \cdot \nabla\phi_1.$$

We can exclude from this sum all triangles that do not contain both of the vertices (otherwise either $\phi_1 = 0$ or $\phi_2 = 0$). (The same argument implies that the entry $A_{1,3}$ is zero. If two vertices are *not connected* then the associated Galerkin matrix entry is zero.) This leaves the two triangles ⑥ (lighter shading) and ⑦ (darker shading) shown in the picture. With the local numbering shown we deduce that

$$A_{1,2} = \int_{⑥} \nabla L_2 \cdot \nabla L_1 + \int_{⑦} \nabla L_1 \cdot \nabla L_2 = A^{⑥}_{1,2} + A^{⑦}_{2,1} = (-1/2) + (-1/2) = -1.$$

Generating the other entries in the same way gives rise to the *fully assembled* system for the three interior (red) nodes,

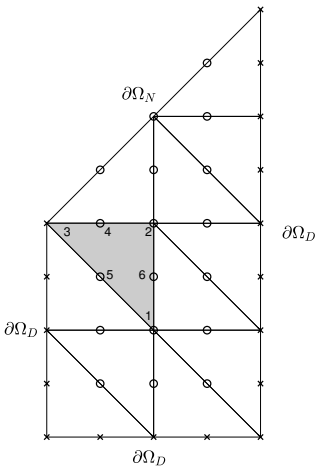
$$\begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 6/24 \\ 5/24 \\ 4/24 \end{bmatrix}. \tag{4.16}$$

This can readily be solved to give the Galerkin solution

$$u_h(x,y) = 0.08974\,\phi_1(x,y) + 0.10897\,\phi_2(x,y) + 0.13782\,\phi_3(x,y).$$

Recall that, by construction, the computed values $u_1 = 0.08974$, $u_2 = 0.10897$ and $u_3 = 0.13782$ represent the values of the solution at the three interior vertices.

There are two ways of increasing the accuracy of this solution. The simplest way is to generate a finer triangulation (*h-refinement*), for example by subdividing every element into four smaller ones by connecting the mid-edge points. An alternative approach is to increase the degree of the polynomial approximation within each triangle (*p-refinement*). To give an illustration, the picture shows what happens if the interpolation points (or *nodes*) are taken to be vertices and mid-edge points. Note that while we still have $n_k = 12$ elements, we now have $n = 18$ (interior nodes marked with a circle) and $n_\partial = 17$ (nodes on $\partial\Omega_D$ marked with a cross). Thus, in this case, the finite element solution space will be spanned by 35 piecewise *quadratic hat* functions,



$$X_h^E = \operatorname{span}\{\psi_j(x,y)\}_{j=1}^{n+n_\partial}$$

satisfying the interpolation conditions

$$\psi_j = \begin{cases} 1 & \text{at node } j, \\ 0 & \text{at node } i \neq j. \end{cases}$$

If we focus on the specific triangle ⑥ that is highlighted in the picture then the six localised quadratic basis functions are given by

$$\begin{aligned} &N_1 = 2L_1^2 - L_1, && N_2 = 2L_2^2 - L_2, && N_3 = 2L_3^2 - L_3, \\ &N_4 = 4L_2L_3, && N_5 = 4L_1L_3, && N_6 = 4L_1L_2, \end{aligned}$$

and the element contributions are given by

$$A_{i,j}^{⑥} = \int_{⑥} \nabla N_j \cdot \nabla N_i, \quad i,j = 1,2,\ldots,6; \qquad f_i^{⑥} = \int_{⑥} N_i, \quad i = 1,2,\ldots,6.$$

Working out the matrix entries explicitly gives

$$A^{⑥} = \begin{bmatrix} 1/2 & 1/6 & 0 & 0 & 0 & -2/3 \\ 1/6 & 1 & 1/6 & -2/3 & 0 & -2/3 \\ 0 & 1/6 & 1/2 & -2/3 & 0 & 0 \\ 0 & -2/3 & -2/3 & 8/3 & -4/3 & 0 \\ 0 & 0 & 0 & -4/3 & 8/3 & -4/3 \\ -2/3 & -2/3 & 0 & 0 & -4/3 & 8/3 \end{bmatrix}, \quad f^{⑥} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/24 \\ 1/24 \\ 1/24 \end{bmatrix}.$$

Assembling the 12 (n_k) element contributions gives an 18×18 Galerkin system

$$\begin{bmatrix} 4 & 1/3 & 0 & \cdots & 0 \\ 1/3 & 4 & 1/3 & \cdots & 0 \\ 0 & 1/3 & 2 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \\ 0 & 0 & 0 & \cdots & 8/3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{18} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1/24 \end{bmatrix}$$

which can be solved to give the refined Galerkin solution

$$u_h(x, y) = 0.10207\, \psi_1(x, y) + 0.13536\, \psi_2(x, y) + \cdots + 0.05901\, \psi_{18}(x, y).$$

To quantify the improvement in solution accuracy, we note that the value $u_1 = 0.10207$ is much closer to the exact value $u(0.5, -0.5) = 0.10236\ldots$ (computed on a very fine grid) than the original estimate $u_1 = 0.08974$.

(**Nonzero boundary conditions**)
The structure of the Lagrangian basis means that imposing a *nonzero* essential boundary condition $u = g_D \neq 0$ is a straightforward exercise. The key point is that the function g_D is interpolated at the n_∂ boundary nodes in the construction (3.12).

To illustrate the construction process, consider solving the worked example with $g_D = x^2 + y^2$ instead of $g_D = 0$. In this situation, we want to impose the boundary values $\overline{u}_j = x_j^2 + y_j^2$ at nodes 4–12 on the boundary of the mesh pictured at the beginning of this section. Consider node 5 as a representative example. Suppose that a preliminary version of the Galerkin system (3.14) is constructed by assembling all 12 element matrices $A^{(k)}$ and load vectors $f^{(k)}$. Then, to enforce the value $\overline{u}_5 = 0^2 + (-1/2)^2$, the given value must be included in the definition of the vector $\boldsymbol{f}$ of (3.15) by multiplying the fifth column of A by the specified boundary value $\overline{u}_5 = 1/4$ and then subtracting the result from $\boldsymbol{f}$. Correspondingly, since ϕ_5 is being removed from the space of test functions, the fifth row and the fifth column of the preliminary Galerkin matrix must then be deleted. Having done this for all 9 boundary nodes the reduced system matrix will be identical to that in (4.16).

Note that it is easiest to treat any nonzero Neumann boundary conditions in system (3.14) after the assembly process and the imposition of essential boundary conditions has been completed. At this stage, the boundary contribution in (3.15) can be assembled by running through the boundary edges on $\partial\Omega_N$ and evaluating the component edge contributions using standard (one-dimensional) Gauss quadrature.

5. Convergence analysis From a numerical analysis perspective, one would like to show that the approximation u_h is guaranteed to converge to the solution u when the grid is increasingly refined (that is, when the mesh parameter $h \to 0$).[7] Since point values of $H^1(\Omega)$ functions may not be uniquely defined when $\Omega \subset \mathbb{R}^2$ (or $\Omega \subset \mathbb{R}^3$), we will need to work in a a higher-order Sobolev space if we are to make progress. Specifically, if it is known that $u \in H^2(\Omega)$ (usually written as $\| \mathrm{D}^2 u \|_{L^2(\Omega)} < \infty$, where $\mathrm{D}^2 u$ is the square root of the sum of the squares of second weak derivatives of u) then it is assured that the weak solution is a continuous function; that is $u \in C^0(\overline{\Omega})$. The reason why this is important is that the piecewise linear spline *interpolant* of u is then a well-defined entity. The simplifying assumption is formalised in the following definition.

(H^2 **regularity**)
The weak solution $u \in X_E$ of (2.1) and (2.2) is said to be H^2 regular if there exists a constant C such that $\| \mathrm{D}^2 u \|_{L^2(\Omega)} \leq C \| f \|_{L^2(\Omega)}$ for every $f \in L^2(\Omega)$. A sufficient condition for this regularity is that the domain Ω is *convex*.

Note that the worked example discussed in Section 4 is not covered by the following analysis; the unbounded radial derivative at the corner means that $u \notin H^2(\Omega)$. In general, if we have H^2 regularity then the finite element approximation is guaranteed to converge linearly.

(Linear convergence in energy)
If the weak solution $u \in X_E$ of (2.1)–(2.2) is H^2 *regular* then the linear finite element function u_h solving (3.8) using a triangular mesh T_h satisfies the error bound

$$\| u - u_h \|_E \leq C_1 h \, \| f \|_{L^2(\Omega)}, \tag{5.17}$$

where C_1 is a constant[8] and where the mesh parameter h is the longest triangle edge in T_h. The bound (5.17) implies that $u_h \to u$ in the limit $h \to 0$.

In simple terms, if the domain Ω is *convex* then the one-dimensional optimal approximation bound generalises to the case of two (or more) dimensions. Since the task of establishing interpolation error estimation in multiple dimensions is beyond our scope, we will proceed at a high level, omitting the more technical details of the convergence proof. As in one dimension, the first step in establishing (5.17) is to introduce the linear interpolation function $\pi_h u$ defined on T_h. This function agrees with the solution u at all vertices of the triangulation and is a member of the solution approximation space

[7] Ideally, one would also like to quantify the *rate of convergence*.
[8] This constant depends only on the *shape regularity* of the triangulation.

X_E^h. Then, using the best approximation property (3.11) and breaking the integration into element contributions we see that

$$\begin{aligned} \|u - u_h\|_E^2 &\leq \|u - \pi_h u\|_E^2 \\ &= \|\nabla(u - \pi_h u)\|_{L^2(\Omega)}^2 \\ &= \sum_{\text{ⓚ} \in T_h} \|\nabla(u - \pi_h u)\|_{L^2(\triangle_k)}^2 . \end{aligned} \tag{5.18}$$

The element contribution to the interpolation error can be bounded using a *scaling argument*, which is discussed next. This argument will lead to the *local interpolation error* bound

$$\|\nabla(u - \pi_h u)\|_{L^2(\triangle_k)} \leq C \frac{h_k^3}{|\triangle_k|} \left\| \mathrm{D}^2 u \right\|_{L^2(\triangle_k)} , \tag{5.19}$$

where C is a constant, $|\triangle_k|$ is the area of triangle ⓚ and h_k is the length of the longest edge. Plugging (5.19) into (5.18) and then using the trigonometric bound $\frac{h_k^2}{4} \sin\theta_k \leq |\triangle_k|$, where θ_k is the smallest angle, gives

$$\|u - u_h\|_E^2 \leq C \sum_{\text{ⓚ} \in T_h} \frac{h_k^6}{|\triangle_k|^2} \left\| \mathrm{D}^2 u \right\|_{L^2(\triangle_k)}^2 \leq \overline{C} \sum_{\text{ⓚ} \in T_h} \frac{h_k^2}{\sin^2\theta_k} \left\| \mathrm{D}^2 u \right\|_{L^2(\triangle_k)}^2 . \tag{5.20}$$

This is the point where we need to make precise the notion of *shape regularity*.

(Minimum angle condition)
A sequence of meshes $\{T_h\}$ is said to be shape regular if there exists a *minimum angle* θ_* such that $\theta_k \geq \theta_*$ for all triangles ⓚ in T_h.

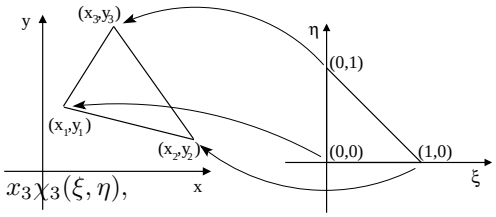
Note that since $\theta_* \leq \theta_k \leq \frac{\pi}{3}$ we have $\sin\theta_* \leq \sin\theta_k$ so that $1/\sin^2\theta_k \leq C_*$. Finally, since $h_k \leq h = \max h_k$ we can further simplify the right-hand side to give

$$\|u - u_h\|_E^2 \leq \overline{C}\, C_*\, h^2 \sum_{\text{ⓚ} \in T_h} \left\| \mathrm{D}^2 u \right\|_{L^2(\triangle_k)}^2 = \overline{C}\, C_*\, h^2 \left\| \mathrm{D}^2 u \right\|_{L^2(\Omega)}^2 ,$$

which, when combined with H^2 regularity, immediately leads to the linear convergence bound (5.17).

The interpolation error bound (5.19) is the key result. It is established by mapping the error from the physical element $\triangle_k$ to a reference element (defined in ξ, η coordinate space), bounding the error in terms of norms of higher derivatives, and then mapping the higher derivatives back to the physical element.

For straight-sided triangles the mapping is defined for all points $(x, y) \in \triangle_k$ and is given by



(5.21a)
$$x(\xi,\eta) = x_1\chi_1(\xi,\eta) + x_2\chi_2(\xi,\eta) + x_3\chi_3(\xi,\eta),$$

(5.21b)
$$y(\xi,\eta) = y_1\chi_1(\xi,\eta) + y_2\chi_2(\xi,\eta) + y_3\chi_3(\xi,\eta),$$

where

$$\chi_1(\xi,\eta) = 1 - \xi - \eta, \quad \chi_2(\xi,\eta) = \xi, \quad \chi_3(\xi,\eta) = \eta$$

are simply the *barycentric* basis functions defined on the reference element. We note in passing that elements with curved sides can be generated using an analogous mapping defined using the six quadratic basis functions introduced in the previous section. The map from the reference element $\triangle_*$ onto $\triangle_k$ is differentiable. Thus, given a differentiable function $\varphi(\xi,\eta)$, we can transform derivatives via

$$\begin{bmatrix} \dfrac{\partial\varphi}{\partial\xi} \\ \dfrac{\partial\varphi}{\partial\eta} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial x}{\partial\xi} & \dfrac{\partial y}{\partial\xi} \\ \dfrac{\partial x}{\partial\eta} & \dfrac{\partial y}{\partial\eta} \end{bmatrix} \begin{bmatrix} \dfrac{\partial\varphi}{\partial x} \\ \dfrac{\partial\varphi}{\partial y} \end{bmatrix}. \tag{5.22}$$

The *Jacobian matrix* in (5.22) can be calculated by differentiating (5.21):

$$J_k = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix} =: \begin{bmatrix} c_3 - b_3 \\ -c_2 & b_2 \end{bmatrix}. \tag{5.23}$$

Thus we see that J_k is a constant matrix over the reference element, and that the determinant

$$|J_k| = \begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{vmatrix} = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = 2|\triangle_k|$$

is simply the ratio of the area of the mapped element $\triangle_k$ to that of the reference element $\triangle*$. The fact that $|J_k(\xi,\eta)| \neq 0$ for all points $(\xi,\eta) \in \triangle*$ ensures that the inverse mapping from $\triangle_k$ onto the reference element is uniquely defined and is differentiable. This means that the derivative transformation (5.22) can be inverted to give

$$\begin{bmatrix} \dfrac{\partial\varphi}{\partial x} \\ \dfrac{\partial\varphi}{\partial y} \end{bmatrix} = \frac{1}{2|\triangle_k|} \begin{bmatrix} b_2 & b_3 \\ c_2 & c_3 \end{bmatrix} \begin{bmatrix} \dfrac{\partial\varphi}{\partial\xi} \\ \dfrac{\partial\varphi}{\partial\eta} \end{bmatrix}. \tag{5.24}$$

Returning to (5.19), defining $e_k = (u - \pi_h u_h)\big|_{\triangle_k}$ and letting $\overline{e}_k$ represent the mapped error on the reference triangle, we have

$$\begin{aligned} \|\nabla e_k\|^2_{L^2(\triangle_k)} &= \int_{\triangle_k} \left(\frac{\partial e_k}{\partial x}\right)^2 + \left(\frac{\partial e_k}{\partial y}\right)^2 \mathrm{d}x\ \mathrm{d}y \\ &= \int_{\triangle_*} \left(\left(\frac{\partial \bar{e}_k}{\partial x}\right)^2 + \left(\frac{\partial \bar{e}_k}{\partial y}\right)^2 \right) 2|\triangle_k|\ \mathrm{d}\xi\ \mathrm{d}\eta, \end{aligned} \tag{5.25}$$

where the derivatives satisfy (5.24); in particular, the first term is of the form

$$\left(\frac{\partial \bar{e}_k}{\partial x}\right)^2 = \frac{1}{4|\triangle_k|^2} \left(b_2 \frac{\partial \bar{e}_k}{\partial \xi} + b_3 \frac{\partial \bar{e}_k}{\partial \eta} \right)^2$$

with b_2 and b_3 defined by (5.23). Using the facts that $(a+b)^2 \leq 2(a^2+b^2)$ and $|b_i| \leq h_k$, then gives the bound

$$2|\triangle_k| \left(\frac{\partial \bar{e}_k}{\partial x}\right)^2 \leq \frac{h_k^2}{|\triangle_k|} \left(\left(\frac{\partial \bar{e}_k}{\partial \xi}\right)^2 + \left(\frac{\partial \bar{e}_k}{\partial \eta}\right)^2 \right).$$

The second term in (5.25) can be bounded in exactly the same way ($|c_i| \leq h_k$). Summing the two terms gives

$$\|\nabla(u - \pi_h u)\|^2_{L^2(\triangle_k)} \leq 2 \frac{h_k^2}{|\triangle_k|} \|\nabla(\bar{u} - \pi_h \bar{u})\|^2_{L^2(\triangle_*)}. \tag{5.26}$$

We will need to use a technical argument due to Bramble & Hilbert [2] at this point in the discussion. It provides a general estimate for the interpolation error measured in the appropriate Sobolev space norm.[9]

(Bramble–Hilbert lemma)
If $\pi_h^1 \bar{u}$ is the *linear* interpolant of a sufficiently smooth function $\bar{u}$ defined on a reference triangle, we have

$$\left\|\nabla(\bar{u} - \pi_h^1 \bar{u})\right\|_{L^2(\triangle_*)} \leq C \left\|\mathrm{D}^2(\bar{u} - \pi_h^1 \bar{u})\right\|_{L^2(\triangle_*)} = C \left\|\mathrm{D}^2 \bar{u}\right\|_{L^2(\triangle_*)}. \tag{5.27}$$

If $\pi_h^2 \bar{u}$ represents the *quadratic* interpolant of a sufficiently smooth function $\bar{u}$ defined on a reference triangle, then we have a refined estimate

$$\left\|\nabla(\bar{u} - \pi_h^2 \bar{u})\right\|_{L^2(\triangle_*)} \leq C \left\|\mathrm{D}^3(\bar{u} - \pi_h^2 \bar{u})\right\|_{L^2(\triangle_*)} = C \left\|\mathrm{D}^3 \bar{u}\right\|_{L^2(\triangle_*)}, \tag{5.28}$$

where $\mathrm{D}^3 u$ is the square root of the sum of squares of the third derivatives of the weak solution u.

[9] This lemma is a multidimensional extension of the error estimates that can be estalished in one dimension using Rolle's theorem. See Braess [1, pp. 77–78] for further discussion.

In simple terms, the error due to linear interpolation in a unit triangle measured in the energy norm is bounded by the L^2 norm of the second derivatives of the interpolation error. Combining (5.27) with (5.26) gives

$$\|\nabla(u - \pi_h u)\|^2_{L^2(\triangle_k)} \leq C \frac{h_k^2}{|\triangle_k|} \left\|\mathrm{D}^2 \bar{u}\right\|^2_{L^2(\triangle *)}. \tag{5.29}$$

The final step of the scaling argument is to map the second derivative terms appearing on the right-hand side of (5.29) back to the physical element. By definition,

$$\begin{aligned} \left\|\mathrm{D}^2 \bar{u}\right\|^2_{L^2(\triangle *)} &= \int_{\triangle *} \left(\frac{\partial^2 \bar{u}}{\partial \xi^2}\right)^2 + \left(\frac{\partial^2 \bar{u}}{\partial \xi \, \partial \eta}\right)^2 + \left(\frac{\partial^2 \bar{u}}{\partial \eta^2}\right)^2 \mathrm{d}\xi \, \mathrm{d}\eta \\ &= \int_{\triangle_k} \left(\left(\frac{\partial^2 u}{\partial \xi^2}\right)^2 + \left(\frac{\partial^2 u}{\partial \xi \, \partial \eta}\right)^2 + \left(\frac{\partial^2 u}{\partial \eta^2}\right)^2\right) \frac{1}{2|\triangle_k|} \, \mathrm{d}x \, \mathrm{d}y, \end{aligned} \tag{5.30}$$

where the derivatives are mapped using (5.22); in particular, the first term is of the form

$$\begin{aligned} \left(\frac{\partial}{\partial \xi}\left(\frac{\partial u}{\partial \xi}\right)\right)^2 &= \left(c_3 \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial \xi}\right) - b_3 \frac{\partial}{\partial y}\left(\frac{\partial u}{\partial \xi}\right)\right)^2 \\ &= \left(c_3^2 \frac{\partial^2 u}{\partial x^2} - 2 c_3 b_3 \frac{\partial^2 u}{\partial x \, \partial y} + b_3^2 \frac{\partial^2 u}{\partial y^2}\right)^2 \\ &\leq 3\left(c_3^4 \left(\frac{\partial^2 u}{\partial x^2}\right)^2 + 4 c_3^2 b_3^2 \left(\frac{\partial^2 u}{\partial x \, \partial y}\right)^2 + b_3^4 \left(\frac{\partial^2 u}{\partial y^2}\right)^2\right) \\ &\leq 12 h_k^4 \left(\left(\frac{\partial^2 u}{\partial x^2}\right)^2 + \left(\frac{\partial^2 u}{\partial x \, \partial y}\right)^2 + \left(\frac{\partial^2 u}{\partial y^2}\right)^2\right). \end{aligned}$$

The second and third terms in (5.30) can be bounded in exactly the same manner. Summing the three terms gives the result

$$\left\|\mathrm{D}^2 \bar{u}\right\|^2_{L^2(\triangle *)} \leq 18 h_k^2 \frac{h_k^2}{|\triangle_k|} \left\|\mathrm{D}^2 u\right\|^2_{L^2(\triangle_k)}, \tag{5.31}$$

which when combined with (5.29) gives the interpolation error bound (5.19).

It is instructive to consider how the convergence rate would be changed if we were to use piecewise quadratic finite element approximation in place of piecewise linear approximation in the worked example. Mapping the interpolation error to the reference triangle gives estimate (5.26). Next, assuming *additional smoothness* of the exact solution (third derivatives are square integrable), we can combine the refined estimate (5.28) with (5.26) to give the

higher-order bound

$$\left\|\nabla(u-\pi_h^2 u)\right\|^2_{L^2(\triangle_k)} \le C\frac{h_k^2}{|\triangle_k|}\left\|\mathrm{D}^3\bar{u}\right\|_{L^2(\triangle_*)}. \tag{5.32}$$

Thus, following the construction above and mapping the (four) third derivative terms on the right-hand side of (5.32) back to the physical element gives the estimate

$$\left\|\mathrm{D}^3\bar{u}\right\|^2_{L^2(\triangle_*)} \le 48h_k^4\frac{h_k^2}{|\triangle_k|}\left\|\mathrm{D}^3 u\right\|^2_{L^2(\triangle_k)}, \tag{5.33}$$

which when combined with (5.32) gives the improved interpolation error bound

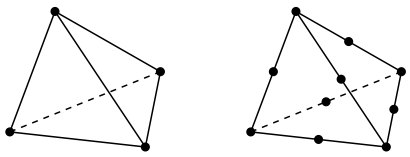
$$\left\|\nabla(u-\pi_h^2 u)\right\|_{L^2(\triangle_k)} \le C\frac{h_k^4}{|\triangle_k|}\left\|\mathrm{D}^3 u\right\|_{L^2(\triangle_k)}. \tag{5.34}$$

The bottom line: If we have an H^3 regular problem then the quadratic finite element approximation is guaranteed to converge at a higher-order rate.

(Quadratic convergence in energy)
If the weak solution $u \in X_E$ of (2.1) and (2.2) is smooth enough then the quadratic finite element function u_h solving (3.8) using a triangular mesh T_h satisfies the error bound

$$\|u-u_h\|_E \le C_2\, h^2, \tag{5.35}$$

where h is the longest triangle edge in T_h and C_2 is a constant that depends on the shape regularity and on $\|\,\mathrm{D}^3 u\|_{L^2(\Omega)}$.



While the results in this section have been established in the context of two-dimensional approximation, the convergence estimates (5.17) and (5.35) also hold when a three-dimensional Poisson problem is solved on a *tetrahedral* tessellation using piecewise linear and quadratic approximation respectively. In this case the interpolation points are the vertices and mid-edges shown in the picture.

6. Estimation of the approximation error An unanswered question at this point is: How accurate are the linear and quadratic finite element solutions computed in Section 4. Alternatively, let $e = u - u_h \in X$ be the *approximation error*. Can it be quantified?

We will describe two alternative strategies for estimating the energy error $\|e\|_E$ in the remainder of this section. The first strategy (applicable in cases where the solution is *not* H^2 regular) is to compare the energy of the discrete solution $\|u_h\|_E$ with the energy of the exact solution $\|u\|_E$. Specifically, exploiting *Galerkin orthogonality* (3.9) gives the following:

$$\begin{aligned}
\|u-u_h\|_E^2 &= \int_\Omega (\nabla u - \nabla u_h)\cdot(\nabla u - \nabla u_h) \\
&= \int_\Omega (\nabla u - \nabla u_h)\cdot \nabla u - \underbrace{\int_\Omega (\nabla u - \nabla u_h)\cdot \nabla u_h}_{=0 \quad \text{since } u_h \in X^h} \\
&= \int_\Omega (\nabla u - \nabla u_h)\cdot \nabla u + \int_\Omega (\nabla u - \nabla u_h)\cdot \nabla u_h \\
&= \int_\Omega (\nabla u - \nabla u_h)\cdot (\nabla u + \nabla u_h) \\
&= \int_\Omega \big(\nabla u\cdot\nabla u - \nabla u_h\cdot\nabla u_h\big) = \|u\|_E^2 - \|u_h\|_E^2.
\end{aligned}$$

In words, the square of the energy error is the difference between the square of the *exact* energy and the square of the *discrete* energy.[10] A key point is that if $u \notin X^h$ then $u_h \neq u$, so that $\|u-u_h\|_E > 0$, which implies that $\|u\|_E > \|u_h\|_E$.

Note that since the exact solution u is not accessible, $\|e\|_E$ needs to be estimated from a reference solution u_{ref}, typically generated by solving the problem on a highly refined grid. Thus, instead of $\|e\|_E$ one computes

$$\|e_{\text{ref}}\|_E = \big(\|u_{\text{ref}}\|_E^2 - \|u_h\|_E^2\big)^{1/2}. \tag{6.36}$$

One reason why the representation (6.36) is so useful is that having solved the Galerkin system (3.14), the discrete energy is simply the scalar product of the solution vector $\boldsymbol{u}$ and the load vector $\boldsymbol{f}$:

$$\begin{aligned}
\|u_h\|_E^2 &= \sum_j \sum_i u_j u_i \int_\Omega \nabla\phi_j\cdot\nabla\phi_i \\
&= \sum_i \sum_j u_i u_j A_{i,j} = \boldsymbol{u}^T A\boldsymbol{u} = \boldsymbol{u}\cdot\boldsymbol{f}.
\end{aligned}$$

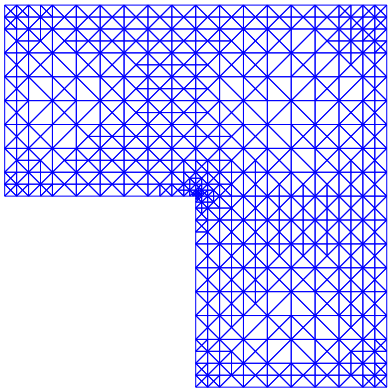
The error representation (6.36) can provide an answer to the question at the start of this section. In particular, if we (re-)solve the problem in Section 4 on a sequence of uniformly refined grids and compute the discrete energies we get the results tabulated. A reference value $\|e_{\text{ref}}\|_E = 0.46268$ that is

[10] Pythagoras in Russian: Galerkin orthogonality implies that $\|u\|_E^2 = \|u_h\|_E^2 + \|u-u_h\|_E^2$.

correct to five decimal places can also be generated by running an adaptive refinement procedure; details will be presented later.[11]

	Linear approximation		Quadratic approximation	
h	$\|u_h\|_E$	$\|e_{\text{ref}}\|_E$	$\|u_h\|_E$	$\|e_{\text{ref}}\|_E$
$1/4$	0.43796	1.492×10^{-1}	0.46116	3.746×10^{-2}
$1/8$	0.45520	8.283×10^{-2}	0.46216	2.184×10^{-2}
$1/16$	0.46038	4.605×10^{-2}	0.46248	1.343×10^{-2}
$1/32$	0.46194	2.612×10^{-2}	0.46260	8.262×10^{-3}
$1/64$	0.46243	1.512×10^{-2}	0.46265	4.924×10^{-3}
$1/128$	0.46259	8.840×10^{-3}	0.46267	2.610×10^{-3}
$1/\infty$	0.46268	0	0.46268	0

Looking at the computed values of $\|u_h\|_E$ it is clear that both columns of discrete energies converge to the reference value. Looking at the associated errors we find that the error reduction rate is slower than linear—the ratio of successive entries is always bigger than 2^{-1}. We can also observe that the rate of convergence using quadratic approximation is exactly the same as the rate using linear approximation. This behaviour is symptomatic of situations where the weak solution u is not smooth enough to satisfy the assumptions made in deriving the a priori error estimates (5.17) and (5.35). This deterioration in the convergence rate is the primary motivation for developing the error estimation strategies that are discussed next.



The question of the accuracy of the finite element approximation can be answered in much more refined way. The idea is to estimate the error e by postprocessing $u_h \in X_E^h$, so as to compute an accurate estimate of the error in any given element $\eta_k \approx \|\nabla(u - u_h)\|_{L^2(\triangle_k)}$. This process is known as a posteriori error estimation. This local error estimate can then be used to drive an *adaptive* refinement process that attempts to construct a mesh of elements so that the error is *equidistributed*, so that it is roughly the same on each element. An important requirement is that η_T must be cheap to compute—as a rule of thumb, the computational work should scale linearly as the number of elements is increased. For the singular example discussed

[11] With a specified tolerance of 0.0002 the adaptive procedure generates a nonuniform grid containing 168,671 vertices with extremely small triangles in the vicinity of the singularity.

earlier, adaptive refinement will lead to a succession of meshes like the one shown, that are selectively refined in the vicinity of the reentrant corner so as to equidistribute the error and enhance overall cost effectiveness. A key requirement is that the accuracy should be guaranteed in the sense that the estimated global error is an upper bound on the exact error,

$$\|\nabla(u-u_h)\|^2_{L^2(\Omega)} = \sum_{\Bbbk\in T_h} \|\nabla(u-u_h)\|^2_{L^2(\triangle_k)} \le C(\theta_*)\sum_{\Bbbk\in T_h}\eta_k^2, \tag{6.37}$$

with a constant C that depends only on shape regularity. If, in addition to satisfying (6.37), the estimate η_k gives a lower bound for the exact local error

$$\eta_k \le C(\theta_{\omega_k})\,\|\nabla(u-u_h)\|_{L^2(\omega_k)}\,, \tag{6.38}$$

where ω_k represents the patch of elements adjoining $\triangle_k$, then the estimator η_k is likely to be effective if it is used to drive an adaptive refinement process.

How does one compute η_k? Borrowing from linear algebra, we can compute the "backward error" by substituting the computed solution u_h into the weak formulation to give a *residual function* r that satisfies

$$\int_\Omega rv = \int_\Omega fv + \int_{\partial\Omega_N} g_N v\;\mathrm{d}s - \int_\Omega \nabla u_h\cdot\nabla v \quad \forall\, v\in X^h_*, \tag{6.39}$$

where X^h_* is a suitably chosen test space $X^h_*\subset X$ to be discussed later. We will refer to X^h_* as the *detail space*. Note that, by the definition of the weak solution,

$$0 = \int_\Omega fv + \int_{\partial\Omega_N} g_N v\;\mathrm{d}s - \int_\Omega \nabla u\cdot\nabla v \quad \forall\, v\in X^h_*, \tag{6.40}$$

so if we subtract equations we get a simple relationship between the error function $e=u-u_h\in X$ and the residual function r,

$$\int_\Omega rv = \int_\Omega \nabla e\cdot\nabla v \quad \forall\, v\in X^h_*. \tag{6.41}$$

The error characterisation (6.41) underlies our a posteriori error estimation procedure. The selection of the detail space is key. First, if the error estimate is to be computable then X^h_* needs to be finite-dimensional. Second, if $X^h_*\subseteq X^h$ then $r=0$ (from Galerkin orthogonality) so the detail space has to include functions that are not in the original approximation space X^h. Third, we would like to compute estimates of the error element by element. For this reason we consider a broken version[12] of (6.39) and integrate the rightmost

[12] In a finite element context "broken" means the breaking of an integral over the domain into the sum of integrals over individual elments.

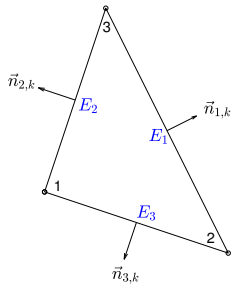
term by parts to give

$$\begin{aligned}\int_\Omega rv &= \int_{\partial\Omega_N} g_N v \,\mathrm{d}s + \int_\Omega fv - \sum_{\textcircled{k}\in T_h}\int_{\triangle_k} \nabla u_h\cdot\nabla v \\ &= \int_{\partial\Omega_N} g_N v \,\mathrm{d}s + \int_\Omega fv + \sum_{\textcircled{k}\in T_h}\int_{\triangle_k} \nabla^2 u_h\, v - \sum_{\textcircled{k}\in T_h}\int_{\partial\triangle_k} (\nabla u_h\cdot\vec{n})\, v \,\mathrm{d}s \\ &= \int_{\partial\Omega_N} g_N v \,\mathrm{d}s + \sum_{\textcircled{k}\in T_h}\int_{\triangle_k} \left\{f+\nabla^2 u_h\right\} v - \sum_{\textcircled{k}\in T_h}\int_{\partial\triangle_k} (\nabla u_h\cdot\vec{n})\, v \,\mathrm{d}s.\end{aligned} \tag{6.42}$$

There are actually three residuals embedded in (6.42). The first of these is the *interior residual* (or the PDE residual) $R_k = \left\{f+\nabla^2 u_h\right\}\big|_{\textcircled{k}}$. Note that using linear finite element approximation R_k simplifies to $f\big|_{\textcircled{k}}$.

The other two residuals are associated with the boundary terms in the error representation (6.42). If we introduce the set of triangle edges $\mathcal{E}_k = \{E_1, E_2, E_3\}$ then the element flux contributions can be written

$$\sum_{\textcircled{k}\in T_h}\sum_{E\in\mathcal{E}_k}\int_E (\nabla u_h\cdot\vec{n}_{E,k})\, v \,\mathrm{d}s.$$



The triangle edges can be classified into three types. The first type are Dirichlet boundary edges, $E\in\mathcal{E}_D$. On these edges we have $v=0$ so the contribution to $\int_\Omega rv$ is zero. The second type are Neumann boundary edges $E\in\mathcal{E}_N$. For such edges, the outward flux can be combined with the $\partial\Omega_N$ term in (6.42) to give the *boundary flux residual* $R_N = \{g_N - \nabla u_h\cdot\vec{n}\}\big|_{E\in\mathcal{E}_N}$. The third and final type of edge is *interior edges*, $E\in\mathcal{E}_h$. Every interior edge will incorporate two contributions to the residual error (one from the triangle ⓚ and the other from the adjoining triangle, ⓝ say). Combining the two contributions gives the *flux jump residual*[13] associated with edge E,

$$R_E = [\![\partial u_h/\partial n]\!]_E = \nabla u_h\cdot\vec{n}_{E,k} + \nabla u_h\cdot\vec{n}_{E,n} = \left\{\nabla u_h\big|_{\textcircled{k}} - \nabla u_h\big|_{\textcircled{n}}\right\}\cdot\vec{n}_{E,k}. \tag{6.43}$$

Note that R_E is a constant function on the edge E in the case that the finite element solution u_h is piecewise linear. Next, substituting the definitions R_k,

[13]The reason that this is a residual is that the classical solution u is a differentiable function so there are no interior flux jumps.

R_N and R_E into the error representation gives

$$\int_\Omega rv = \int_{\partial\Omega_N} R_N v \, \mathrm{d}s + \sum_{\textcircled{k} \in T_h} \int_{\triangle_k} R_k \, v - \sum_{E \in \mathcal{E}_h} \int_E R_E \, v \, \mathrm{d}s.$$

Finally, this can be written as an assembly of element contributions by *equidistributing* the flux jump to the two adjoining elements and then setting $R_E = 2\{g_N - \nabla u_h \cdot \vec{n}\}$ on Neumann boundary edges,

$$\int_\Omega rv = \sum_{\textcircled{k} \in T_h} \left\{ \int_{\triangle_k} R_k \, v - \frac{1}{2} \sum_{E \in \mathcal{E}_k} \int_E R_E \, v \, \mathrm{d}s \right\}. \tag{6.44}$$

Equating (6.41) with (6.44) suggests a mechanism for estimating the error element by element when one is given a suitable detail space X_*^h defined on a given element $\textcircled{k} \in T_h$. This leads to the following characterisation.

(Local error estimator)
A local error estimator is a function $e_k \in X_*^h$ that satisfies the local Neumann problem

$$\int_{\triangle_k} \nabla e_k \cdot \nabla v_k = \int_{\triangle_k} R_k \, v_k - \frac{1}{2} \sum_{E \in \mathcal{E}_k} \int_E R_E \, v_k \, \mathrm{d}s \tag{6.45}$$

for all test functions v_k in the *detail space* X_*^h. Here, R_k is the interior residual and R_E is the flux jump residual associated with the finite element solution $u_h \in X_E^h$. The associated (energy) error estimate is

$$\eta_k = \|\nabla e_k\|_{L^2(\triangle_k)} \approx \|\nabla(u - u_h)\|_{L^2(\triangle_k)}.$$

In a practical setting the detail space can be generated using either *h-refinement* (subdividing the triangle into four smaller ones) or *p-refinement* (adding quadratic interpolation functions at the mid-edge points). To give an illustration, we will reconsider the worked example discussed earlier. The error in the linear finite element solution can be estimated in the highlighted element ⑥ by first computing the constant interior residual

$$R_k = \{f + \nabla^2 u_h\}\big|_{\textcircled{k}} = f\big|_{\textcircled{6}}$$

and then computing the flux jumps R_E across edges E_1, E_2 and E_3 using (6.43).

Next we set up a local problem (6.45) with a detail space

$$X_*^h = \operatorname{span}\{\phi_{E_1}, \phi_{E_2}, \phi_{E_3}\},$$

consisting of the piecewise linear interpolation functions that are defined on the highlighted patch of four triangles (corresponding to an *h-refinement*). Note that the vertex interpolation functions are not included in the basis, so functions in the detail space are zero at the vertices. Computing the matrix and right-hand-side entries explicitly gives the 3×3 system

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} 0.03365 \\ 0.04087 \\ -0.01843 \end{bmatrix},$$

which can be solved to give the local error estimator

$$e_{\textcircled{b}}(x,y) = 0.04107\,\phi_{E_1}(x,y) + 0.04848\,\phi_{E_2}(x,y) + 0.01502\,\phi_{E_3}(x,y).$$

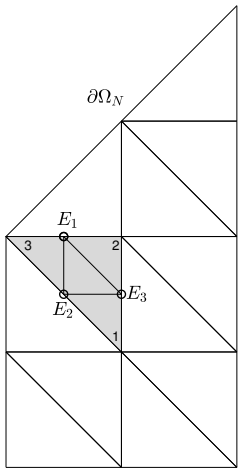
The associated element error estimate is also readily computed: $\eta_k = 0.00309$.

The error estimate is useful for two reasons. First, the global estimate

$$\eta = \Big\{ \sum_{\textcircled{k} \in T_h} \eta_k^2 \Big\}^{1/2} = 0.16265...$$

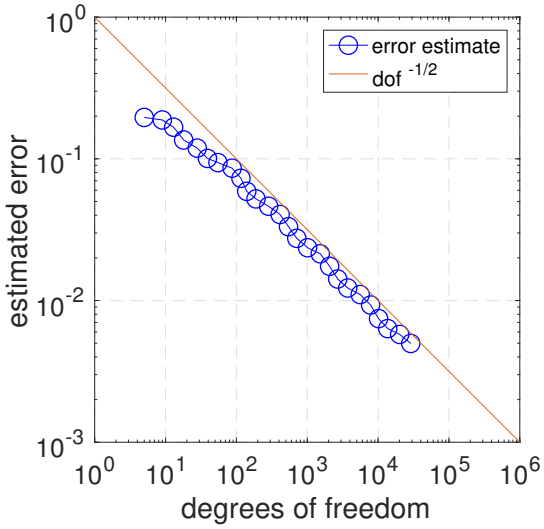
gives a reliably accurate estimate of the overall energy error. Computational testing shows that the *effectivity* of the local error estimator strategy is always close to 1, typically

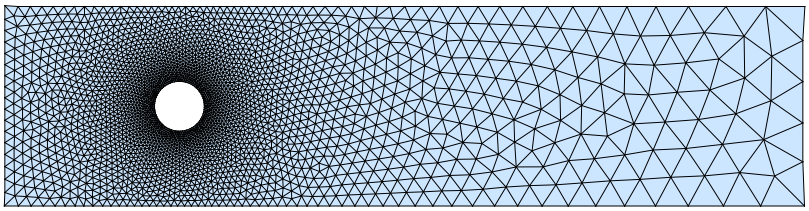
$$0.9 \leq \frac{\eta}{\|u - u_h\|_E} \leq 1.1.$$



Second, if we compare the element error estimates η_k with the total error η, we can *selectively refine* the specific elements that contribute the most to the estimated error. Thus we don't refine areas where the solution is flat so that little error reduction would be achieved. This is the process that was used to generate the highly nonisotropic subdivision illustrated earlier. When an adaptive refinement procedure is applied to the problem solved in Section 4 one is able to reduce the (estimated) error from 0.16265... to less than 0.005 in 25 refinement steps.

Plotting the estimated error against the number of degrees of freedom we see that the rate of convergence is $O(n^{-1/2})$. This rate is *optimal*. On a uniform grid, n varies like h^{-2} so $O(n^{-1/2})$ corresponds to *linear convergence*: this is the convergence rate that one would anticipate using piecewise linear approximation to solve an H^2 regular problem in two dimensions!





The computational results that are presented in this chapter were computed using the freely downloadable T-IFISS software package: http://www.manchester.ac.uk/ifiss/tifiss.html. The T-IFISS software package can be used to explore the geometric flexibility of the finite element method. To give an example, the pictured triangular mesh of a rectangular domain with a hole can be generated automatically using the DistMesh package: http://persson.berkeley.edu/distmesh/ which is incorporated in the T-IFISS package.

References

[1] Dietrich Braess. *Finite elements*. Cambridge University Press, Cambridge, 2007. Third edition, ISBN: 978-0-521-70518-9.

[2] James A. Bramble and Stephen R. Hilbert. Estimation of linear functionals on Sobolev spaces with application to Fourier transforms and spline interpolation. *SIAM Journal on Numerical Analysis*, 7:112–124, 1970. doi: 0.1137/0707006.

[3] James Robinson. *Infinite-dimensional dynamical systems*. Cambridge University Press, Cambridge, 2001. ISBN: 978-0-521-63564-0.

[4] Gilbert Strang and George Fix. *An analysis of the finite element method*. Wellesley–Cambridge, 2008. Second edition, ISBN: 978-0-980-253270-7.