76
Piecewise linears in $\mathbb{R}^2$ and $\mathbb{R}^3$

> ...usually he sat in a comfortable attitude, looking down, slightly stooped, with hands folded above his lap. He spoke quite freely, very clearly, simply and plainly: but when he wanted to emphasise a new viewpoint ... then he lifted his head, turned to one of those sitting next to him, and gazed at him with his beautiful, penetrating blue eyes during the emphatic speech. ... If he proceeded from an explanation of principles to the development of mathematical formulas, then he got up, and in a stately very upright posture he wrote on a blackboard beside him in his peculiarly beautiful handwriting: he always succeeded through economy and deliberate arrangement in making do with a rather small space. For numerical examples, on whose careful completion he placed special value, he brought along the requisite data on little slips of paper. (Dedekind about Gauss)

76.1 Introduction

In this chapter, we prepare for the application of FEM to partial differential equations by discussing approximation of functions by piecewise linear functions in in $\mathbb{R}^2$ and $\mathbb{R}^3$. We consider three main topics: (i) the construction of a mesh, or *triangulation*, for a domain in $\mathbb{R}^2$ or $\mathbb{R}^3$, (ii) the construction piecewise linear functions on a triangulation, and (iii) estimation of interpolation errors.

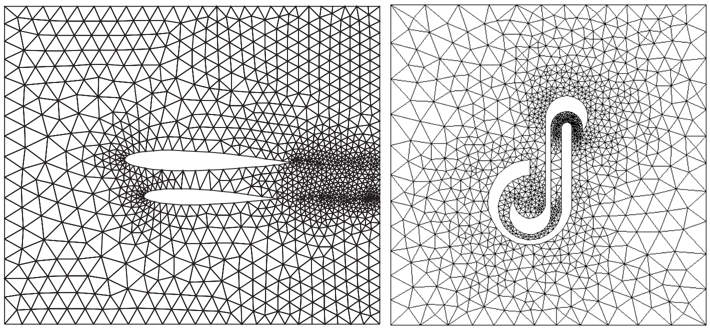

FIGURE 76.1. The mesh on the left was used in a computation of the flow of air around two airfoils. The mesh on the right was used to discretize a piece of metal punched with a fancy character. In both cases, the meshes are adapted to allow accurate computation, taking into account both the behavior of the solution and the shape of the domain.

76.2 Triangulation of a domain in $\mathbb{R}^2$

We start by considering a two-dimensional domain Ω with a polygonal boundary Γ. A *triangulation* $\mathcal{T}_h = \{K\}$ is a sub-division of Ω into a non-overlapping set of triangles, or *elements*, K constructed so that no vertex of one triangle lies on the edge of another triangle, see Fig. 76.2. We use $\mathcal{N}_h = \{N\}$ to denote the set of *nodes* N or corners of the triangles, usually numbered $N_1, N_2, ..., N_M$, where M is the total number of nodes. A triangulation is specified by a list of the coordinates of the nodes, together with a list containing the numbers of the nodes of each triangle. We may also list the set of triangle sides or *edges* $\mathcal{S}_h = \{S\}$, with each edge S specified by the node numbers of its two end-points, and a list of the nodes and edges on the boundary Γ.

We measure the size of a triangle $K \in \mathcal{T}_h$, by the length h_K of its largest side, which is called the *diameter* of the triangle. The *mesh function* $h(x)$ associated to a triangulation $\mathcal{T}_h$ is the piecewise constant function defined so $h(x) = h_K$ for $x \in K$ for each $K \in \mathcal{T}_h$. We measure the degree of *isotropy* of an element $K \in \mathcal{T}_h$ by its smallest angle τ_K. If $\tau_K \approx \pi/3$ then K is almost isosceles, while if τ_K is small then K is thin, see Fig. 76.3. We use the smallest angle among the triangles in $\mathcal{T}_h$, i.e.

$$\tau = \min_{K \in \mathcal{T}_h} \tau_K$$

as a measure of the degree of anistropy of the triangulation $\mathcal{T}_h$. We shall see below that certain interpolation errors related to approximation with piecewise linear functions on a given triangulation get larger as τ tends to zero, corresponding to allowing the triangles to very thin.

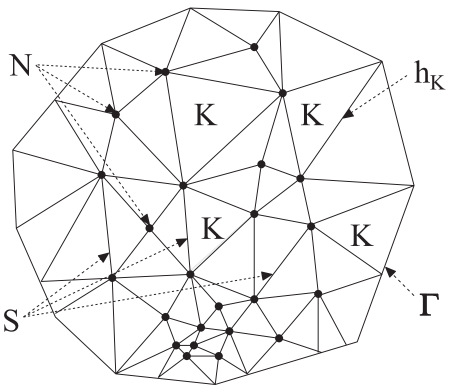

FIGURE 76.2. A triangulation of a domain Ω.

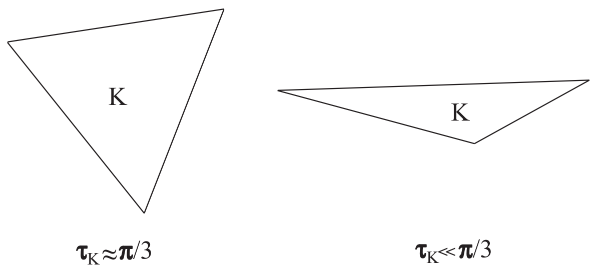

FIGURE 76.3. Measuring the isotropy of a triangle.

The basic problem of *mesh generation* is to generate a triangulation of a given domain with mesh size given by a prescribed mesh function $h(x)$. This problem arises in each step of an adaptive algorithm, where a new mesh function is computed from an approximate solution on a given mesh, and a new mesh is constructed with mesh size given by the new mesh function. The process is then repeated until a stopping criterion is satisfied. The new mesh may be constructed from scratch or by modification of the previous mesh including local refinement or coarsening.

In the *advancing front* strategy a mesh with given mesh size is constructed beginning at some point (often on the boundary) by successively adding one triangle after another, each with a mesh size determined by the mesh function. The curve dividing the domain into a part already triangulated and the remaining part is called the *front*. The front sweeps through the domain during the triangulation process. An alternative is to use a *h-refinement* strategy, where a mesh with a specified local mesh size is constructed by successively dividing elements of an initial coarse triangulation with the elements referred to as *parents*, into smaller elements, called the *children*. We illustrate the refinement and advancing front strategies in

Fig. 76.4. It is often useful to combine the two strategies using the advanc-

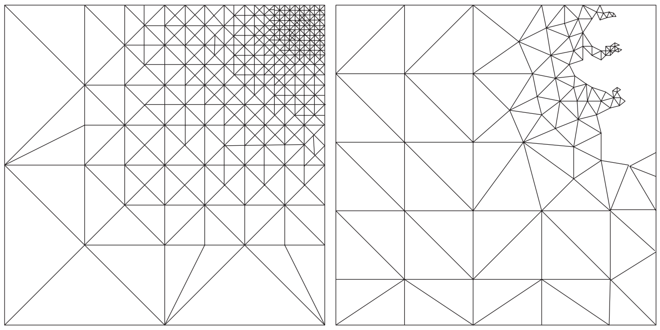

FIGURE 76.4. The mesh on the left is being constructed by successive h refinement starting from the coarse parent mesh drawn with thick lines. The mesh on the right is being constructed by an advancing front strategy. In both cases, high resolution is required near the upper right-hand corner.

ing front strategy to construct an initial mesh that represents the geometry of the domain with adequate accuracy, and use adaptive h-refinement.

There are various strategies for performing the division in an h-refinement aimed at limiting the degree of anisotropy of the elements. After the refinements are completed, the resulting mesh is fixed up by the addition of edges aimed at avoiding nodes that are located in the middle of element sides. This causes a mild "spreading" of the adapted region. We illustrate one technique for h-refinement in Fig. 76.5. In general, refining a mesh tends

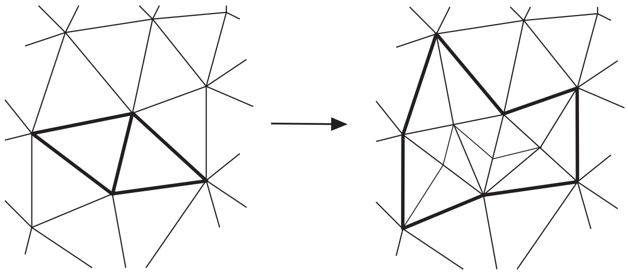

FIGURE 76.5. On the left, two elements in the mesh have been marked for refinement. The refinement uses the Rivara algorithm in which an element is divided into two pieces by inserting a side connecting the node opposite the longest side to the midpoint of the longest side. Additional sides are added to avoid having a node of one element on the side of another element. The refinement is shown in the mesh on the right along with the boundary of all the elements that had to be refined in addition to those originally marked for refinement.

to introduce elements with small angles, as can be seen in Fig. 76.5 and it is an interesting problem to construct algorithms for mesh refinement that avoid this tendency in situations where the degree of anisotropy has to be limited. On the other hand, in certain circumstances, it is important to use "stretched" meshes that have regions of thin elements aligned together to give a high degree of refinement in one direction. In these cases, we also introduce mesh functions that give the local stretching, or degree of anisotropy, and the orientation of the elements. We discuss the construction and use of such meshes in the advanced companion volume.

76.3 Mesh generation in $\mathbb{R}^3$

Mesh generation in three dimensions is analogous to that in two dimensions with the triangles being replaced by *tetrahedra*. In practice, the geometric constraints involved become more complicated and the number of elements also increases drastically. We show some examples in Fig. **??** and Fig. 76.7.

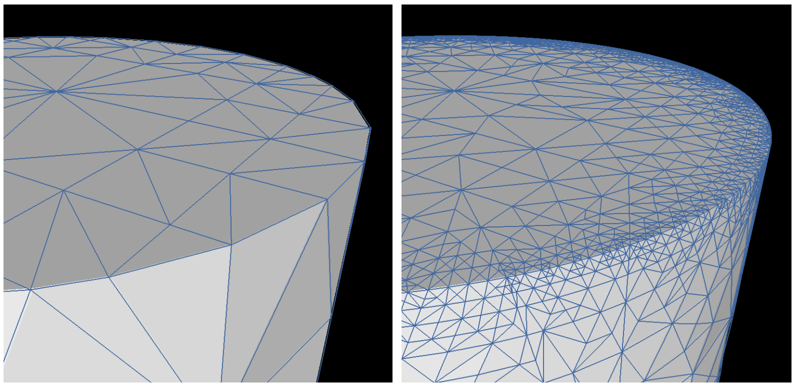

FIGURE 76.6. Initial and refined mesh of cylinder.

76.4 Piecewise linear functions

Let $\mathcal{T}_h = \{K\}$ be a triangulation of a two-dimensional domain Ω with piecewise polynomial boundary Γ, let $\mathcal{N}_h = \{N\}$ denote the nodes of $\mathcal{T}_h$ and introduce the corresponding the finite dimensional vector space V_h consisting of the continuous piecewise linear functions on $\mathcal{T}_h$. In other words,

$$V_h = \big\{v : v \text{ is continuous on } \Omega, v|_K \in \mathcal{P}(K) \text{ for } K \in \mathcal{T}_h\big\},$$

where $\mathcal{P}(K)$ denotes the set of linear functions on K, i.e., the set of functions $v(x) = v(x_1, x_2)$ of the form $v(x) = c_0 + c_1x_1 + c_2x_2$ for some constants

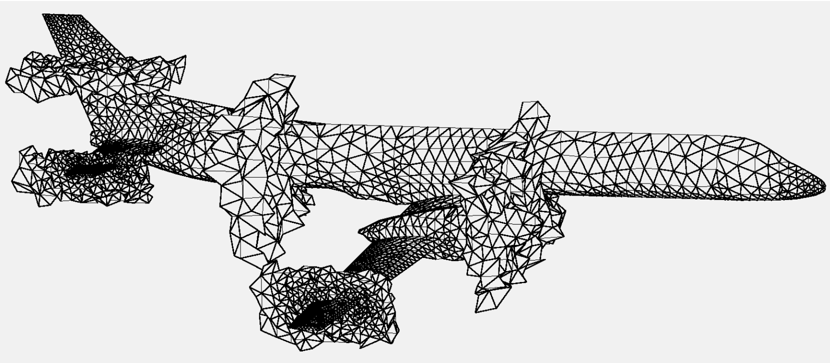

FIGURE 76.7. The surface mesh on the body, and parts of a tetrahedral mesh around a Saab 2000

c_i. We can describe a function $v(x)$ in V_h by the nodal values $v(N)$ with $N \in \mathcal{N}_h$ because of two facts. The first is that a linear function is uniquely determined by its values at three points, as long as they don't lie on a straight line. To prove this claim, let $K \in \mathcal{T}_h$ have vertices $a^i = (a_1^i, a_2^i)$,

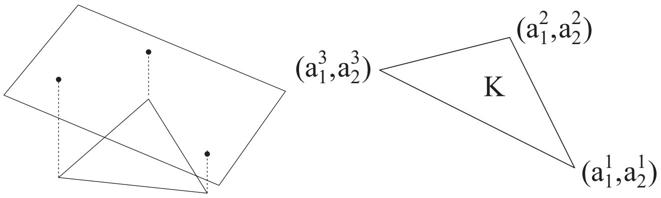

FIGURE 76.8. On the left, we show that the three nodal values on a triangle determine a linear function. On the right, we show the notation used to describe the nodes of a typical triangle.

$i = 1, 2, 3$, see Fig. 76.8. We want to show that $v \in \mathcal{P}(K)$ is determined uniquely by $\{v(a^1), v(a^2), v(a^3)\} = \{v_1, v_2, v_3\}$. A linear function v can be written $v(x_1, x_2) = c_0 + c_1x_1 + c_2x_2$ for some constants c_0, c_1, c_2. Substituting the nodal values of v into this expression yields a linear system of equations:

$$\begin{pmatrix} 1 & a_1^1 & a_2^1 \\ 1 & a_1^2 & a_2^2 \\ 1 & a_1^3 & a_1^3 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}.$$

The determinant of the coefficient matrix is equal to the determinant of the following matrix resulting from subtracting the first row from the second and third row:

$$\begin{pmatrix} 1 & a_1^1 & a_2^1 \\ 0 & a_1^2 - a_1^1 & a_2^2 - a_2^1 \\ 0 & a_1^3 - a_1^1 & a_2^2 - a_2^1 \end{pmatrix},$$

which is equal to the twice the area of the triangle K (up to the sign). The determinant of the coefficient matrix is thus non-zero, and we conclude that the system of equations(76.4) has a unique solution. We conclude that at linear function is uniquely specified by its values at three non-colinear points.

The second fact is that if a function is linear in each of two neighboring triangles and its nodal values on the two common nodes of the triangles are equal, then the function is continuous across the common edge. To see this,

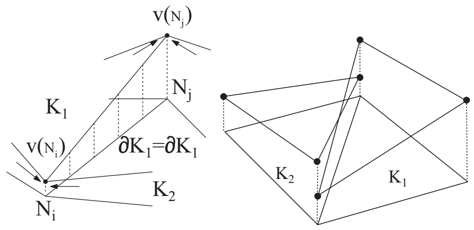

FIGURE 76.9. On the left, we show that a function that is piecewise linear on triangles reduces to a linear function of one variable on triangle edges. On the right, we plot a function that is piecewise linear on triangles whose values at the common nodes on two neighboring triangles do not agree.

let K_1 and K_2 be adjoining triangles with common boundary $\partial K_1 = \partial K_2$; see the figure on the left in Fig. 76.9. Parametrizing v along this boundary, we see that v is a linear function of one variable there. Such functions are determined uniquely by the value at two points, and therefore since the values of v on K_1 and K_2 at the common nodes agree, the values of v on the common boundary between K_1 and K_2 agree, and v is indeed continuous across the boundary.

To construct a set of basis functions for V_h, we begin by describing a set of *element basis functions* for triangles. Once again, assuming that a triangle K has nodes at $\{a^1, a^2, a^3\}$, the element nodal basis is the set of functions $\lambda_i \in \mathcal{P}(K)$, $i = 1, 2, 3$, such that

$$\lambda_i(a^j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

We show these functions in Fig. 76.10.

We construct the *global* basis functions for V_h by piecing together the element basis functions on neighboring elements using the continuity requirement, i.e. by matching element basis functions on neighboring trian-

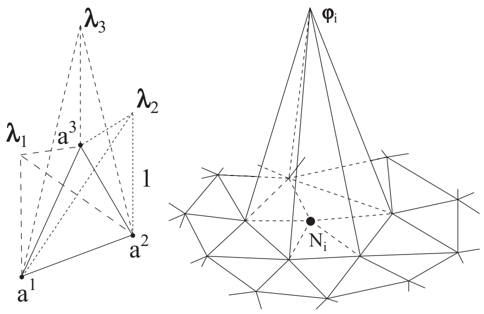

FIGURE 76.10. On the left, we show the three element nodal basis functions for the linear functions on K. On the right, we show a typical global basis "tent" function.

gles that have the same nodal values on the common edge. The resulting set of basis functions $\{\varphi_j\}_{j=1}^M$, where N_1, N_2,..., N_M is an enumeration of the nodes $N \in \mathcal{N}_h$, is called the set of *tent* functions. The tent functions can also be defined by specifying that $\varphi_j \in V_h$ satisfy

$$\varphi_j(N_i) = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases}$$

for $i, j = 1, ..., M$. We illustrate a typical tent function in Fig. 76.10. We see in particular that the *support* of φ_i is the set of triangles that share the common node N_i.

The tent functions are a nodal basis for V_h because if $v \in V_h$ then

$$v(x) = \sum_{i=1}^{M} v(N_i)\varphi_i(x).$$

76.5 Max-norm error estimates for piecewise linear interpolation

In this section we prove the basic pointwise maximum norm error estimate for linear interpolation on a triangle, which states that the interpolation error depends on the second order partial derivatives of the function being interpolated, i.e. on the "curvature" of the function, the mesh size and the shape of the triangle. Analogous results hold for other norms. The results also extend directly to more than two space dimensions.

Let K be a triangle with vertices $a^i, i = 1, 2, 3$. Given a continuous function v defined on K, let the linear interpolant $\pi_K v \in \mathcal{P}(K)$ be defined by

$$\pi_K v(a^i) = v(a^i), \quad i = 1, 2, 3.$$

We illustrate this in Fig. 76.11.

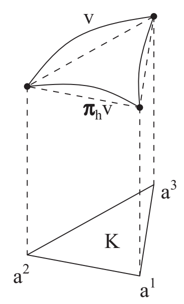

FIGURE 76.11. The nodal interpolant $\pi_K v$ of v.(change from π_h to π_K)

Theorem 76.1 *If v has continuous second derivatives, then*

$$\|v - \pi_K v\|_{L_\infty(K)} \leq \frac{1}{2} h_K^2 \|D^2 v\|_{L_\infty(K)}, \tag{76.1}$$

$$\|\nabla(v - \pi_K v)\|_{L_\infty(K)} \leq \frac{3}{\sin(\tau_K)} h_K \|D^2 v\|_{L_\infty(K)}, \tag{76.2}$$

where h_K is the largest side of K, τ_K is the smallest angle of K, and

$$D^2 v = \left(\sum_{i,j=1}^{2} \left(\frac{\partial^2 v}{\partial x_i \partial x_j}\right)^2 \right)^{1/2}.$$

If ∇v is continuous, then

$$\|v - \pi_K v\|_{L_\infty(K)} \leq h_K \|Dv\|_{L_\infty(K)}, \tag{76.3}$$

Note that the gradient estimate depends on the reciprocal of the sine of the smallest angle of K, and therefore this error bound deteriorates as the the triangle gets thinner.

The proof follows the same general outline as the proofs of corresponding results in the Chapter Piecewise linear approximation. Let λ_i, $i = 1, 2, 3$, be the element basis functions for $\mathcal{P}(K)$ defined by $\lambda_i(a^j) = 1$ if $i = j$, and $\lambda_i(a^j) = 0$ otherwise. A function $w \in \mathcal{P}(K)$ has the representation

$$w(x) = \sum_{i=1}^{3} w(a^i)\lambda_i(x) \quad \text{for } x \in K,$$

and thus

$$\pi_K v(x) = \sum_{i=1}^{3} v(a^i)\lambda_i(x) \quad \text{for } x \in K, \tag{76.4}$$

since $\pi_K v(a^i) = v(a^i)$. We shall derive representation formulas for the interpolation errors $v - \pi_K v$ and $\nabla(v - \pi_K v)$, using a Taylor expansion at $x \in K$:

$$v(y) = v(x) + \nabla v(x) \cdot (y - x) + R(x, y),$$

where

$$R(x, y) = \frac{1}{2} \sum_{i,j=1}^{2} \frac{\partial^2 v}{\partial x_i \partial x_j}(\xi)(y_i - x_i)(y_j - x_j),$$

is the remainder term of order 2 and ξ is a point on the line segment between x and y. In particular choosing $y = a^i = (a_1^i, a_2^i)$, we have

$$v(a^i) = v(x) + \nabla v(x) \cdot (a^i - x) + R_i(x), \tag{76.5}$$

where $R_i(x) = R(x, a^i)$. Inserting (76.5) into (76.4) gives for $x \in K$

$$\pi_K v(x) = v(x) \sum_{i=1}^{3} \lambda_i(x) + \nabla v(x) \cdot \sum_{i=1}^{3} (a^i - x)\lambda_i(x) + \sum_{i=1}^{3} R_i(x)\lambda_i(x). \tag{76.6}$$

We shall use the following identities that hold for $j, k = 1, 2$, and $x \in K$,

$$\sum_{i=1}^{3} \lambda_i(x) = 1, \quad \sum_{i=1}^{3} (a_j^i - x_j)\lambda_i(x) = 0, \tag{76.7}$$

$$\sum_{i=1}^{3} \frac{\partial}{\partial x_k}\lambda_i(x) = 0, \quad \sum_{i=1}^{3} (a_j^i - x_j)\frac{\partial \lambda_i}{\partial x_k} = \delta_{jk}, \tag{76.8}$$

where $\delta_{jk} = 1$ if $j = k$ and $\delta_{jk} = 0$ otherwise. The first of the identities in (76.7) follows by choosing $v(x) = 1$ in (76.6), and the second follows by choosing $v(x) = d_1 x_1 + d_2 x_2$ with $d_i \in \mathbb{R}$. Finally, (76.8) follows by differentiating (76.7).

Using (76.7), we obtain the following representation of the interpolation error,

$$v(x) - \pi_K v(x) = -\sum_{i=1}^{3} R_i(x)\lambda_i(x).$$

Since $|a^i - x| \leq h_K$, we can estimate the remainder term $R_i(x)$ as

$$|R_i(x)| \leq \frac{1}{2} h_K^2 \|D^2 v\|_{L_\infty(K)}, \quad i = 1, 2, 3.$$

where we used Cauchy's inequality twice to estimate an expression of the form $\sum_{ij} x_i c_{ij} x_j = \sum_i x_i \sum_j c_{ij} x_j$.

Now, using the fact that $0 \leq \lambda_i(x) \leq 1$ if $x \in K$, for $i = 1, 2, 3$, we obtain

$$|v(x) - \pi_K v(x)| \leq \max_i |R_i(x)| \sum_{i=1}^{3} \lambda_i(x) \leq \frac{1}{2} h_K^2 \|D^2 v\|_{L_\infty(K)} \quad \text{for } x \in K,$$

which proves (76.1).

To prove (76.2), we differentiate (76.4) with respect to $x_k, k = 1, 2$ to get

$$\nabla(\pi_K v)(x) = \sum_{i=1}^{3} v(a^i)\nabla\lambda_i(x),$$

which together with (76.5) and (76.8) gives the following error representation:

$$\nabla(v - \pi_K v)(x) = -\sum_{i=1}^{3} R_i(x)\nabla\lambda_i(x) \quad \text{for } x \in K.$$

We now note that

$$\max_{x\in K}|\nabla\lambda_i(x)| \leq \frac{2}{h_K \sin(\tau_K)},$$

which follows by an easy estimate of the shortest height (distance from a vertex to the opposite side) of K. We now obtain (76.2) and (76.3) finally follows using tha Mean Value theorem. The proof is now complete.

Let now $\mathcal{T}_h = \{K\}$ be a triangulation of a domain Ω with mesh function $h(x)$, and let π_h denote nodal interpolation into the corresponding space of continuous piecewise linear functions V_h on $\mathcal{T}_h$. The interpolation error estimates of Theorem 76.1 then take the form

$$\|v - \pi_h v\|_{L_\infty(\Omega)} \leq \frac{1}{2}\|h^2 D^2 v\|_{L_\infty(\Omega)}, \tag{76.9}$$

$$\|v - \pi_h v\|_{L_\infty(\Omega)} \leq \|hDv\|_{L_\infty(\Omega)}, \tag{76.10}$$

$$\|\nabla(v - \pi_h v)\|_{L_\infty(\Omega)} \leq \frac{3}{\sin(\tau)}\|hD^2 v\|_{L_\infty(\Omega)}, \tag{76.11}$$

where τ is the smallest of the τ_K. Below we shall use analogs of these estimates with the $L_\infty(\Omega)$ replaced by $L_2(\Omega)$.

76.6 Quadrature in $\mathbb{R}^2$

To compute the stiffness matrix and load vector a FEM, we have to compute integrals of the form $\int_K g(x)\,dx$, where K is a triangle or tetrahedron and g a given function. Sometimes we may evaluate these integrals exactly, but usually this is either impossible or inefficient. Instead we usually evaluate the integrals approximately using quadrature formulas. We briefly present some quadrature formulas for integrals over triangles.

In general, we would like to use quadrature formulas that do not affect the accuracy of the underlying finite element method, which of course requires an estimate of the error due to quadrature. A quadrature formula for an integral over an element K has the form

$$\int_K g(x)\,dx \approx \sum_{i=1}^{q} g(y^i)\omega_i, \tag{76.12}$$

for a specified choice of *nodes* $\{y^i\}$ in K and *weights* $\{\omega_i\}$. We now list some possibilities using the notation a_K^i to denote the vertices of a triangle K, a_K^{ij} to denote the midpoint of the side connecting a_K^i to a_K^j, and a_K^{123} to denote the center of mass of K, and denote by $|K|$ the area of K:

$$\int_K g\,dx \approx g\big(a_K^{123}\big)|K|, \tag{76.13}$$

$$\int_K g(x)\,dx \approx \sum_{j=1}^{3} g(a_K^j)\frac{|K|}{3}, \tag{76.14}$$

$$\int_K g\,dx \approx \sum_{1\le i<j\le 3} g\big(a_K^{ij}\big)\frac{|K|}{3}, \tag{76.15}$$

$$\int_K g\,dx \approx \sum_{j=1}^{3} g\big(a_K^i\big)\frac{|K|}{20} + \sum_{1\le i<j\le 3} g\big(a_K^{ij}\big)\frac{2|K|}{15} + g\big(a_K^{123}\big)\frac{9|K|}{20}. \tag{76.16}$$

We refer to (76.13) as the center of gravity quadrature, to (76.14) as the vertex quadrature, and to (76.15) as the midpoint quadrature. Recall that the accuracy of a quadrature formula is related to the *precision* of the formula. A quadrature formula has precision r if the formula gives the exact value of the integral if the integrand is a polynomial of degree at most $r-1$, but there is some polynomial of degree r such that the formula is not exact. The quadrature error for a quadrature rule of precision r is proportional to h^r, where h is the mesh size. More precisely, the error of a quadrature rule of the form (76.12) satisfies

$$\left|\int_K g\,dx - \sum_{i=1}^{q} g(y^i)\omega_i\right| \le C h_K^r \sum_{|\alpha|=r} \int_K |D^\alpha g|\,dx,$$

where C is a constant. Vertex and center of gravity quadrature have precision 2, midpoint quadrature has precision 3, while (76.16) has precision 4.

In finite element methods based on continuous piecewise linear functions, we often use nodal or vertex quadrature, often also referred to as *lumped mass* quadrature, because the mass matrix computed this way becomes diagonal.

EXAMPLE 76.1. In Fig. 76.12 and Fig. 76.13 we give two examples, one from electromagnetics and the other from fluid mechanics.

Chapter 76 Problems

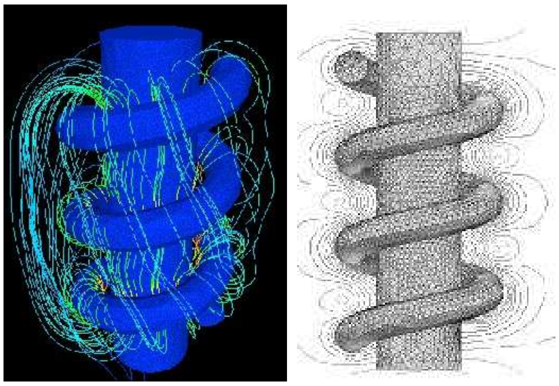

FIGURE 76.12. Magnetic field around coil and mesh

76.1. For a given triangle K, determine the relation between the smallest angle τ_K, the triangle diameter h_K and the diameter ρ_K of the largest inscribed circle.

76.2. Draw the refined mesh that results from sub-dividing the smallest two triangles in the mesh on the right in Fig. 76.5.

76.3. Let K be a tetrahedron with vertices $\{a^i, i = 1, ..., 4\}$. Show that a linear polynomial $v(x) = c_0 + c_1x_1 + c_2x_2 + c_3x_3$ on K is uniquely determined by the nodal values $\{v(a^i), i = 1, ..., 4\}$. Show that the corresponding finite element space V_h consists of continuous functions.

76.4. Prove that the quadrature formulas (76.13),(76.14), (76.15) and (76.16) have the indicated precision.

76.5. Prove that using nodal quadrature to compute a mass matrix for piecewise linears, gives a diagonal mass matrix where a diagonal term is the sum of the terms in the corresponding row in the exactly computed mass matrix. Motivate the term "lumped".

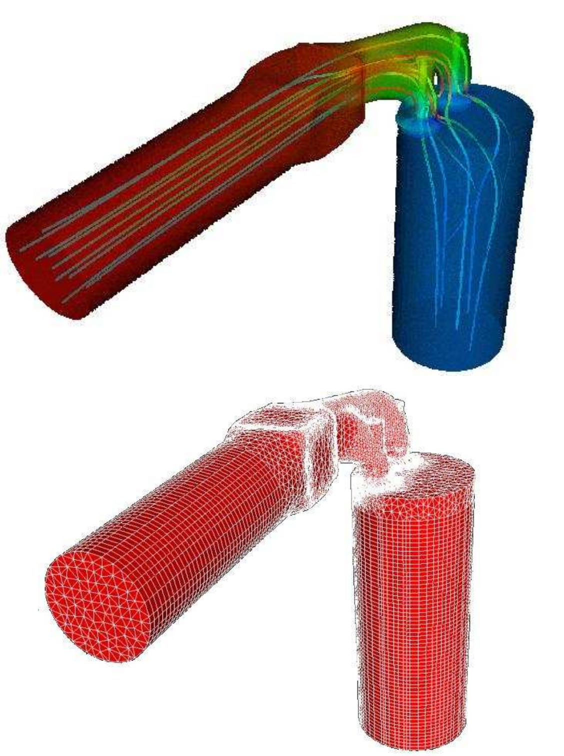

FIGURE 76.13. Flow in diesel engine cylinder inlet and mesh

77

FEM for boundary value problems in $\mathbb{R}^2$ and $\mathbb{R}^3$

...were very confused, skipping suddenly from one idea to another, from one formula to the next, with no attempt to give a connection between them. His presentations were obscure clouds, illuminated from time to time by flashes of pure genius. ... of the thirty who enrolled with me, I was the only one to see it through. (Menabrea about Cauchy 1832)

77.1 Introduction

In this chapter, we extend the cG(1) FEM for reaction-diffusion-convection problems in one space dimension to corresponding boundary value problems in $\mathbb{R}^2$ and $\mathbb{R}^3$ of the form: Find $u : \Omega \to \mathbb{R}$ such that

$$-\nabla \cdot (a\nabla u) + \nabla \cdot (ub) + cu = f \quad \text{in } \Omega, \tag{77.1}$$

together with boundary conditions of Dirichlet, Neumann or Robin type, where $a(x) > 0$, $b(x)$ and $c(x)$ are given variable coefficients, $f(x)$ is a given right hand side, and Ω is a bounded open domain in $\mathbb{R}^2$ or $\mathbb{R}^3$. Note that the coefficient b is a vector (typically corresponding to a given convection velocity), and that the equation (77.1) can alternatively be written

$$-\nabla \cdot (a\nabla u) + b \cdot \nabla u + \hat{c}u = f \quad \text{in } \Omega, \tag{77.2}$$

with $\hat{c} = c + \nabla \cdot b$. In general, problems of this form cannot be solved analytically and we have to rely on a numerical method such as FEM for computing the solution $u(x)$ for given data.

We consider below the extension to corresponding time dependent problems of the form

$$\dot{u} - \nabla \cdot (a\nabla u) + \nabla \cdot (ub) + cu = f, \tag{77.3}$$

together with initial and boundary value problems, including extensions to systems of such equations, using the material in the Chapters The general initial value problem and Adaptive IVP-solvers.

The most fundamental example of the form (77.1) is Poisson's equation with homogeneous Dirichlet boundary conditions corresponding to setting $a = 1$, $b = 0$ and $c = 0$:

$$\begin{cases} -\Delta u(x) = f(x) & \text{for } x \in \Omega, \\ u(x) = 0 & \text{for } x \in \Gamma, \end{cases} \tag{77.4}$$

where Ω is a bounded domain in $\mathbb{R}^2$ with polygonal boundary Γ. We recall that $\nabla \cdot (\nabla u) = \Delta u$. We shall now present the cG(1) method for (77.4) generalizing cG(1) for the two-point boundary value problem (53.9), and then extend to the general problem (77.1).

77.2 Variational formulation

We let $\mathcal{T}_h = \{K\}$ be a triangulation of Ω with mesh function $h(x)$ and internal nodes $N_1, ...N_M$, and we let V_h be the corresponding finite element space of continuous piecewise linear functions that vanish on the boundary Γ. We first give (77.4) the following preliminary variational formulation:

$$-\int_\Omega \Delta u \, v \, dx = \int_\Omega f \, v \, dx \tag{77.5}$$

for all suitable test functions v, which results from multiplying (77.4) by $v(x)$ and integrating over Ω. We now want to rewrite the left-hand side to move a derivative from Δu onto v. Assuming that the test function v is zero on Γ, Green's formula implies

$$-\int_\Omega \Delta u \, v \, dx = -\int_\Gamma \partial_n uv \, ds + \int_\Omega \nabla u \cdot \nabla v \, dx = \int_\Omega \nabla u \cdot \nabla v \, dx,$$

where ∂_n denotes the outward unit normal derivative on Γ. We find that a solution $u(x)$ of (77.4) satisfies

$$\int_\Omega \nabla u \cdot \nabla v \, dx = \int_\Omega f \, v \, dx, \tag{77.6}$$

for all test functions v with $v = 0$ on Γ.

77.3 The cG(1) FEM

We are thus led to the following formulation of the cG(1) FEM for (77.4): Find $U \in V_h$ such that

$$\int_\Omega \nabla U \cdot \nabla v \, dx = \int_\Omega f \, v \, dx \quad \text{for all } v \in V_h, \tag{77.7}$$

where V_h is the space of continuous piecewise linear functions on a triangulation $\mathcal{T}_h$ of Ω that vanish on the boundary Γ. Using the notation

$$(w, v) = \int_\Omega wv \, dx, \quad (\nabla w, \nabla v) = \int_\Omega \nabla w \cdot \nabla v \, dx,$$

we can write cG(1) in the form: Find $U \in V_h$ such that

$$(\nabla U, \nabla v) = (f, v) \quad \text{for all } v \in V_h. \tag{77.8}$$

We see that that the trial space and test spaces are equal ($= V_h$) and include the homogenous Dirichlet boundary condition. The Galerkin orthogonality is expressed by

$$(\nabla u - \nabla U, \nabla v) = 0 \quad \text{for all } v \in V_h, \tag{77.9}$$

which results upon subtracting (77.8) from (77.6) with $v \in V_h$.

We recall that the nodal basis functions $\{\varphi_i\}_{i=1}^M$ associated with the internal nodes $N_1, ...N_M$ of $\mathcal{T}_h$ is a basis for V_h. Expressing U in terms of this basis,

$$U(x) = \sum_{j=1}^M U(N_j)\varphi_j(x), \tag{77.10}$$

substituting into (77.8), and choosing $v = \varphi_i$ for $i = 1, ..., M$, gives

$$\sum_{j=1}^M (\nabla\varphi_j, \nabla\varphi_i) U(N_j) = (f, \varphi_i), \quad i = 1, ..., M.$$

This is equivalent to the linear system of equations

$$A\xi = b, \tag{77.11}$$

where $\xi = (\xi_j)$ is the vector of internal nodal values $\xi_j = U(N_j)$, $A = (a_{ij})$ is the *stiffness matrix* with elements $a_{ij} = (\nabla\varphi_j, \nabla\varphi_i)$ and $b = (b_i)$ with $b_i = (f, \varphi_i)$ is the *load vector*..

The stiffness matrix A is obviously symmetric and it is also positive-definite since for any $v = \sum_{i=1}^{M} \eta_i \varphi_i$ in V_h,

$$\sum_{i,j=1}^{M} \eta_i a_{ij} \eta_j = \sum_{i,j=1}^{M} \eta_i (\nabla \varphi_i, \nabla \varphi_j) \eta_j$$
$$= \left(\nabla \sum_{i=1}^{M} \eta_i \varphi_i, \nabla \sum_{j}^{M} \eta_j \varphi_j \right) = (\nabla v, \nabla v) > 0,$$

unless $\eta_i = 0$ for all i. This means in particular that (77.11) has a unique solution vector U and thus the cG(1) finite element problem (77.8) has a unique solution $U \in V_h$.

77.3.1 An example: uniform triangulation of a square

We compute the stiffness matrix A and load vector b in (77.11) explicitly on the uniform triangulation of the square $\Omega = [0,1] \times [0,1]$ pictured in Fig. 77.1. We choose an integer $m \geq 1$ and set $h = 1/(m+1)$, then construct the triangles as shown. The diameter of the triangles in $\mathcal{T}_h$ is $\sqrt{2}h$ and there are $M = m^2$ internal nodes. We number the nodes starting from the lower left and moving right, then working up across the rows.

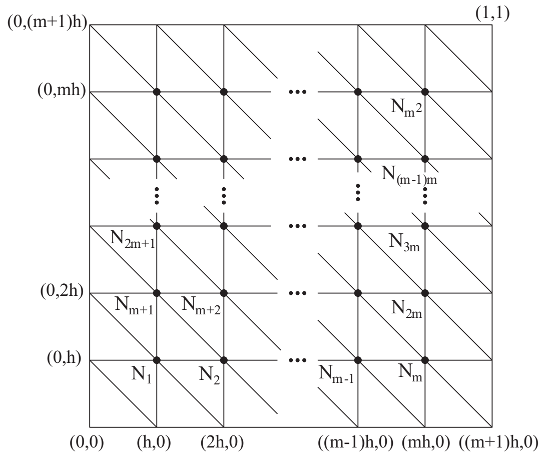

FIGURE 77.1. The standard triangulation of the unit square.

In Fig. 77.2, we show the support of the basis function corresponding to the node N_i along with parts of the basis functions for the neighboring nodes. As in one dimension, the basis functions are "almost" orthogonal in the sense that only basis functions φ_i and φ_j sharing a common triangle

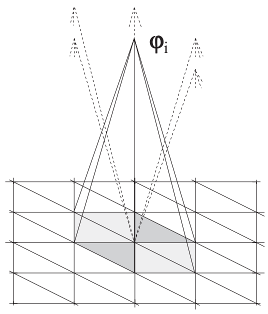

FIGURE 77.2. The support of the basis function φ_i together with parts of the neighboring basis functions.

in their supports yield a non-zero value in $(\nabla\varphi_i, \nabla\varphi_j)$. We show the nodes neighboring N_i in Fig. 77.3. The support of any two neighboring basis

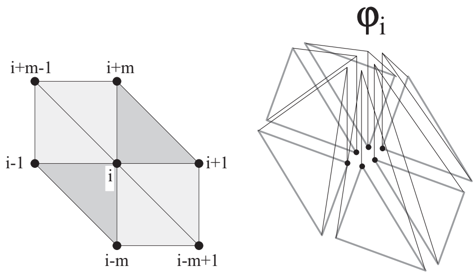

FIGURE 77.3. The indices of the nodes neighboring N_i and an "exploded" view of φ_i.

functions overlap on just two triangles, while a basis function "overlaps itself" on six triangles.

We first compute

$$a_{ii} = (\nabla\varphi_i, \nabla\varphi_i) = \int_\Omega |\nabla\varphi_i|^2 \, dx \;=\; \int_{\text{support of } \varphi_i} |\nabla\varphi_i|^2 \, dx,$$

for $i = 1, ..., m^2$. As noted, we only have to consider the integral over the domain pictured in Fig. 77.3, which is written as a sum of integrals over the six triangles making up the domain. Examining φ_i on these triangles, see Fig. 77.3, we see that there are only two different integrals to be computed since φ_i looks the same, except for orientation, on two of the six triangles and similarly the same on the other four triangles. We shade the corresponding triangles in Fig. 77.2. The orientation affects the direction of $\nabla\varphi_i$ of course, but does not affect $|\nabla\varphi_i|^2$.

We compute $(\nabla\varphi_i, \nabla\varphi_i)$ on the triangle shown in Fig. 77.4. In this case,

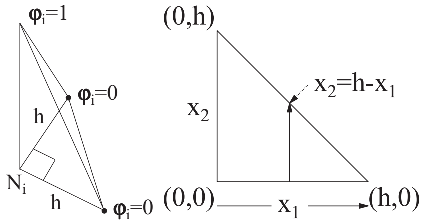

FIGURE 77.4. First case showing φ_i on the left together with the variables used in the reference triangle.

φ_i is one at the node located at the right angle in the triangle and zero at the other two nodes. We change coordinates to compute $(\nabla\varphi_i, \nabla\varphi_i)$ on the *reference triangle* shown in Fig. 77.4. Again, changing to these coordinates does not affect the value of $(\nabla\varphi_i, \nabla\varphi_i)$ since $\nabla\varphi_i$ is constant on the triangle. On the triangle, φ_i can be written $\varphi_i = ax_1 + bx_2 + c$ for some constants a, b, c. Since $\varphi_i(0,0) = 1$, we get $c = 1$. Similarly, we compute a and b to find that $\varphi_i = 1 - x_1/h - x_2/h$ on this triangle. Therefore, $\nabla\varphi_i = \left(-h^{-1}, -h^{-1}\right)$ and the integral is

$$\int_{\triangleright} |\nabla\varphi_i|^2\,dx = \int_0^h \int_0^{h-x_1} \frac{2}{h^2}\,dx_2\,dx_1 = 1.$$

In the second case, φ_i is one at a node located at an acute angle of the triangle and is zero at the other nodes. We illustrate this in Fig. 77.5. We

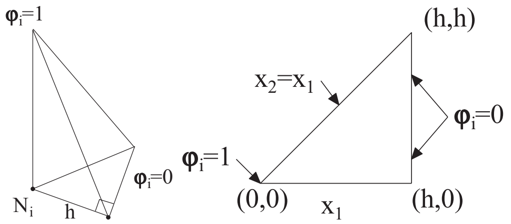

FIGURE 77.5. Second case showing φ_i and the reference triangle.

use the coordinate system shown in Fig. 77.5 to write $\varphi_i = 1 - x_1/h$. When we integrate over the triangle, we get $1/2$.
Summing the contributions from all the triangles gives

$$a_{ii} = (\nabla\varphi_i, \nabla\varphi_i) = 1 + 1 + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} = 4.$$

Next, we compute $(\nabla\varphi_i, \nabla\varphi_j)$ for indices corresponding to neighboring nodes. For a general node N_i, there are two cases of inner products (see

Fig. 77.2 and Fig. 77.3):

$$a_{i\,i-1} = (\nabla\varphi_i, \nabla\varphi_{i-1}) = (\nabla\varphi_i, \nabla\varphi_{i+1}) = (\nabla\varphi_i, \nabla\varphi_{i-m}) = (\nabla\varphi_i, \nabla\varphi_{i+m}),$$

and

$$a_{i\,i-m+1} = (\nabla\varphi_i, \nabla\varphi_{i-m+1}) = (\nabla\varphi_i, \nabla\varphi_{i+m-1}).$$

The orientation of the triangles in each of the two cases are different, but the inner product of the gradients of the respective basis functions is not affected by the orientation. Note that the the equations corresponding to nodes next to the boundary are special, because the nodal values on the boundary are zero, see Fig. 77.1. For example, the equation corresponding to N_1 only involves N_1, N_2 and N_{m+1}.

For the first case, we next compute $(\nabla\varphi_i, \nabla\varphi_{i+1})$. Plotting the intersection of the respective supports shown in Fig. 77.6, we conclude that there

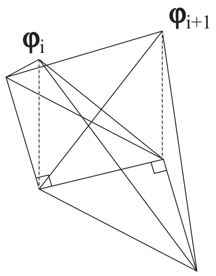

FIGURE 77.6. The overlap of φ_i and φ_{i+1}.

are equal contributions from each of the two triangles in the intersection. We choose one of the triangles and construct a reference triangle as above. Choosing suitable variables, we find that

$$\nabla\varphi_i \cdot \nabla\varphi_{i+1} = \Big(-\frac{1}{h}, -\frac{1}{h}\Big) \cdot \Big(\frac{1}{h}, 0\Big) = -\frac{1}{h^2},$$

and integrating over the triangle gives $-1/2$. Similarly, we see that

$$(\nabla\varphi_i, \nabla\varphi_{i-m+1}) = (\nabla\varphi_i, \nabla\varphi_{i+m-1}) = 0.$$

We can now determine the stiffness matrix A using the information above. We start by considering the first row. The first entry is $(\nabla\varphi_1, \nabla\varphi_1) = 4$ since N_1 has no neighbors to the left or below. The next entry is $(\nabla\varphi_1, \nabla\varphi_2) = -1$. The next entry after that is zero, because the supports of φ_1 and φ_3 do not overlap. This is true in fact of all the entries up to and including φ_m. However, $(\nabla\varphi_1, \nabla\varphi_{m+1}) = -1$, since these neighboring basis functions do share two supporting triangles. Finally, all the rest of the entries in that row are zero because the supports of the corresponding basis functions do not overlap. We continue in this fashion working row by row. The result

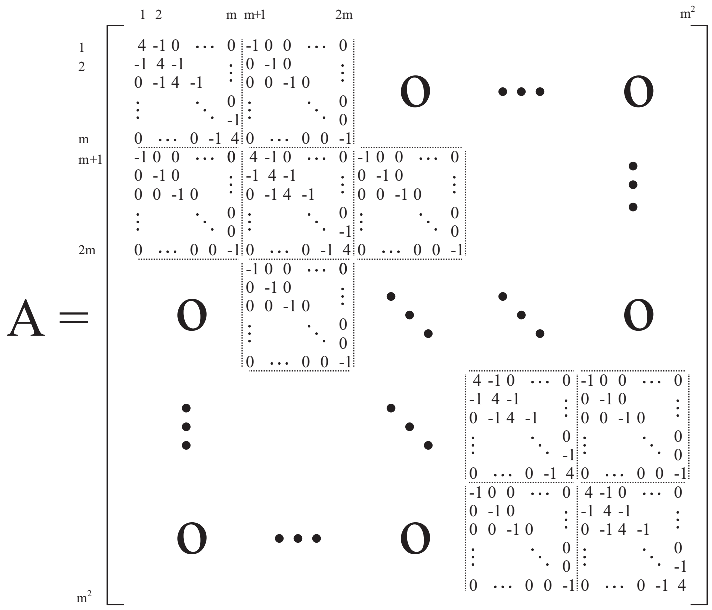

FIGURE 77.7. The stiffness matrix.

is pictured in Fig. 77.7. We see that A has a *block structure* consisting of banded $m \times m$ sub-matrices, most of which consist only of zeros. Note the pattern of entries around corners of the diagonal block matrices; it is a common mistake to program these values incorrectly.

The storage of a sparse matrix and the solution of a sparse system are both affected by the *structure* or *sparsity pattern* of the matrix. The sparsity pattern is affected in turn by the enumeration scheme used to mark the nodes.

There are several algorithms for reordering the coefficients of a sparse matrix to form a matrix with a smaller bandwidth. Reordering the coefficients is equivalent to computing a new basis for the vector space.

The load vector b is computed in the same fashion, separating each integral

$$\int_\Omega f\varphi_i\, dx = \int_{\text{support of } \varphi_i} f(x)\varphi_i(x)\, dx$$

into integrals over the triangles making up the support of φ_i. To compute the elements (f, φ_i) of the load vector, we often use one of the quadrature formulas presented in Chapter 76.

77.4 Basic data structures

To compute the finite element approximation U, we have to compute the coefficients of the stiffness matrix A and load vector b and solve the linear system of equations (77.11). We just computed A and b for a uniform triangulation of the unit square, and we now discuss the case of a general triangulation of a general domain.

We have to compute the non-zero elements $a_{ij} = (\nabla\varphi_j, \nabla\varphi_i)$ of the stiffness matrix A. We know that $a_{ij} = 0$ unless both N_i and N_j are nodes of the same triangle K, because only then the supports of basis functions φ_i and φ_j overlap. The common support corresponding to a non-zero element a_{ij} is equal to the support of φ_i if $j = i$, and is equal to the two triangles with the common edge connecting N_j and N_i if $i \neq j$. In each case a_{ij} is the sum of contributions

$$a_{ij}^K = \int_K \nabla\varphi_j \cdot \nabla\varphi_i \, dx \tag{77.12}$$

over the triangles K in the common support. The process of adding up the contributions a_{ij}^K from the relevant triangles K to get the element a_{ij}, is called *assembling* the stiffness matrix A. Arranging for a given triangle K the numbers a_{ij}^K, where N_i and N_j are nodes of K, into a 3×3 matrix, we obtain the *element stiffness matrix* for the triangle K. We refer to the assembled matrix A as the *global stiffness matrix*. Notice that we use *element* with two different meanings: as an element a_{ij} of the stiffness matrix A, and as a finite element or triangle of the triangulation.

To compute the element stiffness matrix a_{ij}^K for a given triangle K, we need the physical coordinates of the nodes of K. To perform the assembly where we loop over all elements and add the corresponding contributions to the global stiffness matrix, we need the node numbers of each triangle. Similar information is needed to compute the load vector. The required information is arranged in a *data structure*, or data base, containing (i) a list of the coordinates of the nodes numbered in some way, and (ii) a list of the node numbers of each triangle. A list of the numbers of the nodes on the boundary is also needed to handle the boundary conditions This information is typically the output of the mesh generator generating a triangulation of the domain.

77.5 Solving the discrete system

Once we have assembled the stiffness matrix A and computed the load vector b, we have to solve the linear system $AU = b$ to obtain the finite element approximation $U(x)$. We now discuss this topic briefly based on the material presented in Chapter 44. The stiffness matrix resulting from discretizing the Laplacian is symmetric and positive-definite and therefore invertible. These properties also mean that there is a wide choice in the methods used to solve the linear system $AU = b$, which take advantage of the fact that A is sparse.

In the case of the standard uniform discretization of a square, we saw that A is a banded matrix with five non-zero diagonals and bandwidth $m + 1$, where m is the number of nodes on a side. The dimension of A is m^2 and the asymptotic operations count for using Gaussian elimination is $O(m^4) = O(h^{-4})$. Note that even though A has mostly zero diagonals inside the band, fill-in occurs as the elimination is performed, so we may as well treat A as if it has non-zero diagonals throughout the band. Clever rearrangement of A to reduce the amount of fill-in leads to a solution algorithm with an operations count on the order of $O(m^3) = O(h^{-3})$. In contrast, if we treat A as a full matrix, we get an asymptotic operations count of $O(h^{-6})$, which is considerably larger for a large number of elements.

In general, we get a sparse stiffness matrix, though there may not be a band structure. If we want to use a Gaussian elimination method efficiently in general, then it is necessary to first reorder the system to bring the matrix into banded form.

We can also apply both the Jacobi and Gauss-Seidel methods to solve the linear system arising from discretizing the Poisson equation. In the case of the uniform standard discretization of a square for example, the operations count is $O(M)$ per iteration for both methods if we make use of the sparsity of A. Therefore a single step of either method is much cheaper than a direct solve. The question is: How many iterations do we need to compute in order to obtain an accurate solution?

Typically the spectral radius of the iteration matrix of the Jacobi or Gauss-Seidel method is equal to $1 - Ch^2$ with C some moderate positive constant, which means that the convergence rate quickly gets slow as h decreases: to reduce the error a certain factor, we need of the order of $O(h^{-2})$ iterations, and since each iteration takes $O(h^{-2})$ operations, the total number of operations is $O(h^{-4})$, which is the same order as using a banded Gaussian elimination solver.

There has been a lot of activity in developing iterative methods that converge more quickly than Jacobi and Gauss-Seidel. In recent years, very efficient *multi-grid methods* have been developed and are now becoming a standard tool. A multi-grid method is based on a sequence of Gauss-Seidel or Jacobi steps performed on a hierarchy of successively coarser meshes and are optimal in the sense that the solution work is proportional to the total number of unknowns (that is h^{-2} in the model problem).

77.6 An equivalent minimization problem

The variational problem (77.8) is equivalent to the following quadratic *minimization problem*: find $U \in V_h$ such that

$$F(u) \leq F(v) \quad \text{for all } v \in V_h, \tag{77.13}$$

where

$$F(v) = \frac{1}{2}\int_\Omega |\nabla v|^2\, dx - \int_\Omega fv\, dx = \frac{1}{2}(\nabla v, \nabla v) - (f, v).$$

The quantity $F(v)$ may be interpreted as the *total energy* of the function $v(x)$ composed of the *internal energy* $\frac{1}{2}(\nabla v, \nabla v)$ and the *load potential* $-(f, v)$. Thus, the solution U minimizes the total energy $F(v)$ with v varying over V_h.

To see the equivalence of (77.8) and (77.13), we assume first that $U \in V_h$ satisfies (77.8). Let then $v \in V_h$ and write $v = U + (v - U) = U + w$ with $w = v - U \in V_h$. Using $(\nabla U, \nabla w) = (\nabla w, \nabla U)$, we get

$$\begin{aligned} F(v) = F(U + w) = &\frac{1}{2}(\nabla U, \nabla U) + (\nabla U, \nabla w) + \frac{1}{2}(\nabla w, \nabla w) - (f, U) - (f, w) \\ =&F(U) + \frac{1}{2}(\nabla w, \nabla w) \geq F(U), \end{aligned}$$

with equality only if $w = 0$. We conclude that U satisfies (77.13).

Conversely, if U is the solution of (77.13), then we have for all $v \in V_h$

$$g_v(\epsilon) = F(U + \epsilon v) \geq g(0) = F(U) \quad \text{for all } \epsilon \in \mathbb{R},$$

and thus $\epsilon = 0$ is an interior minimum point of $g_v(\epsilon)$ with v fixed, and thus $g_v'(0) = 0$. Computing we get

$$0 = g_v'(0) = (\nabla U, \nabla v) - (f, v)$$

and thus U satisfies (77.8). We sum up in the following theorem:

Theorem 77.1 *The problems* (77.8) *and* (77.13) *are equivalent in the sense that they have the same unique solution.*

77.7 An energy norm a priori error estimate

In this section, we derive a priori and a posteriori estimates of the error $u - U$ in the *energy norm* $\|\nabla(u - U\|$ with

$$\|\nabla v\| = (\int_\Omega |\nabla v|^2\, dx)^{1/2}, \tag{77.14}$$

where u is the exact solution and U a finite element solution of Poisson's equation with homogeneous Dirichlet boundary conditions. The energy norm, which is the L_2 norm of the gradient of a function in this problem, arises naturally in the error analysis of the finite element method because it is closely tied to the variational problem. The gradient of the solution, representing heat flow, electric field, flow velocity, or stress for example, can be a variable of physical interest as much as the solution itself, representing temperature, potential or displacement for example, and in this case, the energy norm is the relevant error measure.

We first prove that the Galerkin finite element approximation is the best approximation of the true solution in V_h with respect to the energy norm.

Theorem 77.2 *Assume that u satisfies the Poisson equation* (77.4) *and U is the Galerkin finite element approximation satisfying* (77.8). *Then*

$$\|\nabla(u-U)\| \leq \|\nabla(u-v)\| \quad \text{for all } v \in V_h. \tag{77.15}$$

The proof goes as follows: Using the Galerkin orthogonality (77.9) with v replaced by $U - v \in V_h$, we can write

$$\|\nabla e\|^2 = (\nabla e, \nabla(u-U)) = (\nabla e, \nabla(u-U)) + (\nabla e, \nabla(U-v)).$$

Adding the terms involving U on the right, whereby U drops out, and using Cauchy's inequality, we get

$$\|\nabla e\|^2 = (\nabla e, \nabla(u-v)) \leq \|\nabla e\| \, \|\nabla(u-v)\|,$$

which proves the theorem after dividing by $\|\nabla e\|$.

Choosing $v = \pi_h u$ and using an $L_2(\Omega)$ analog of the interpolation estimate (76.11), we get the following quantitative a priori error estimate (with $\|v\| = \|v\|_{L_2(\Omega)}$):

Corollary 77.3 *There exists a constant C_i depending only on the minimal angle τ in $\mathcal{T}_h$, such that*

$$\|\nabla(u-U)\| \leq C_i \|hD^2 u\|. \tag{77.16}$$

77.8 An energy norm a posteriori error estimate

We now prove an a posteriori error estimate following the strategy used for the two-point boundary value problem in Chapter 53. A new feature occurring in higher dimensions is the appearance of integrals over the internal edges S in $\mathcal{S}_h$. We start by writing an equation for the error $e = u - U$ using (77.6) and (77.8) to get

$$\begin{aligned}\|\nabla e\|^2 &= (\nabla(u-U), \nabla e) = (\nabla u, \nabla e) - (\nabla U, \nabla e)\\ &= (f, e) - (\nabla U, \nabla e) = (f, e - \pi_h e) - (\nabla U, \nabla(e - \pi_h e)),\end{aligned}$$

where $\pi_h e \in V_h$ is an interpolant of e. We now break up the integrals over Ω into sums of integrals over the triangles K in $\mathcal{T}_h$ and integrate by parts over each triangle in the last term to get

$$\|\nabla e\|^2 = \sum_K \int_K (f + \Delta U)(e - \tilde{\pi}_h e)\, dx - \sum_K \int_{\partial K} \frac{\partial U}{\partial n_K}(e - \pi_h e)\, ds, \quad (77.17)$$

where $\partial U/\partial n_K$ denotes the derivative of U in the outward normal direction n_K of the boundary ∂K of K. In the boundary integral sum in (77.17), each internal edge $S \in \mathcal{S}_h$ occurs twice as a part of each of the boundaries ∂K of the two triangles K that have S as a common side. Of course the outward normals n_K from each of the two triangles K sharing S point in opposite directions. For each side S, we choose one of these normal directions and denote by $\partial_S v$ the derivative of a function v in that direction on S. We note that if $v \in V_h$, then in general $\partial_S v$ is different on the two triangles sharing S; see Fig. 76.9, which indicates the "kink" over S in the graph of v. We can express the sum of the boundary integrals in (77.17) as a sum of integrals over edges of the form

$$\int_S [\partial_S U](e - \pi_h e)\, ds,$$

where $[\partial_S U]$ is the difference, or jump, in the derivative $\partial_S U$ computed from the two triangles sharing S. The jump appears because the outward normal directions of the two triangles sharing S are opposite. We further note that $e - \tilde{\pi}_h e$ is continuous across S, but in general does not vanish on S even if π_h is the nodal interpolant. This is different than the one-dimensional case, where the corresponding sum over nodes does indeed vanish because $e - \pi_h e$ vanishes at the nodes. We may thus rewrite (77.17) as follows with the second sum replaced by a sum over internal edges S:

$$\|\nabla e\|^2 = \sum_K \int_K (f + \Delta U)(e - \pi_h e)\, dx + \sum_{S \in \mathcal{S}_h} \int_S [\partial_S U](e - \pi_h e)\, ds.$$

Next, we return to a sum over element edges ∂K by just distributing each jump equally to the two triangles sharing it, to obtain an *error representation* of the energy norm of the error in terms of the residual error:

$$\|\nabla e\|^2 = \sum_K \int_K (f + \Delta U)(e - \pi_h e)\, dx + \sum_K \frac{1}{2} \int_{\partial K} h_K^{-1} [\partial_S U](e - \pi_h e) h_K\, ds,$$

where we have prepared to estimate the second sum by inserting a factor h_K and compensating. In crude terms, the residual error results from substituting U into the differential equation $-\Delta u - f = 0$, but in reality straightforward substitution is not possible because U is not twice differentiable in Ω. The integral on the right over K is the remainder from substituting U into the differential equation inside each triangle K, while the integral over ∂K arises because $\partial_S U$ in general is different when computed from the two triangles sharing S.

We estimate the first term in the error representation by inserting a factor h, compensating and using the estimate $\|h^{-1}(e - \pi_h e)\| \leq C_i \|\nabla e\|$ analogous to (76.11), to obtain

$$|\sum_K \int_K h(f + \Delta U) h^{-1}(e - \pi_h e)\, dx| \\ \leq \|hR_1(U)\| \|h^{-1}(e - \pi_h e)\| \leq C_i \|hR_1(U)\| \|\nabla e\|,$$

where $R_1(U)$ is the function defined on Ω by setting $R_1(U) = |f + \Delta U|$ on each triangle $K \in \mathcal{T}_h$. We estimate the contribution from the jumps on the edges similarly. Formally, the estimate results from replacing $h_K\, ds$ by dx corresponding to replacing the integrals over element boundaries ∂K by integrals over elements K. Dividing by $\|\nabla e\|$, we obtain the following a posteriori error estimate:

Theorem 77.4 *There is an interpolation constant C_i only depending on the minimal angle τ such that the error of the Galerkin finite element approximation U of the solution u of the Poisson equation satisfies*

$$\|\nabla u - \nabla U\| \leq C_i \|hR(U)\|, \tag{77.18}$$

where $R(U) = R_1(U) + R_2(U)$ with

$$R_1(U) = |f + \Delta U| \quad \text{on } K \in \mathcal{T}_h,$$
$$R_2(U) = \frac{1}{2} \max_{S \subset \partial K} h_K^{-1} \big|[\partial_S U]\big| \quad \text{on } K \in \mathcal{T}_h.$$

Note that $R_1(U)$ is the contribution to the total residual from the interior of the elements K. In the present case of piecewise linear approximation, $R_1(U) = |f|$. Further, $R_2(U)$ is the contribution to the residual from the jump of the normal derivative of U across edges. In the one dimensional problem considered in Chapter 53, this contribution does not appear because the interpolation error may be chosen to be zero at the node points.

77.9 Adaptive error control

The basic goal of adaptive error control is to find a triangulation $\mathcal{T}_h$ with a least number of nodes such that the corresponding finite element approximation U satisfies

$$\|\nabla u - \nabla U\| \leq \text{TOL}. \tag{77.19}$$

Using the a posteriori error estimate we are thus led to find a triangulation $\mathcal{T}_h$ with a least number of nodes such that the corresponding finite element approximation U satisfies

$$C_i \|hR(U)\| \leq \text{TOL}. \tag{77.20}$$

This is a nonlinear constrained minimization problem with U depending on $\mathcal{T}_h$. If (77.18) is a reasonably sharp estimate of the error, then a solution of this optimization problem will meet our original goal.

We cannot expect to be able to solve this minimization problem analytically. Instead, a solution has to be sought by an iterative process in which we start with a coarse initial mesh and then successively modify the mesh by seeking to satisfy the stopping criterion (77.20) with a minimal number of elements. More precisely, we follow the following *adaptive algorithm*:

1. Choose an initial triangulation $\mathcal{T}_h^{(0)}$.
2. Given the j^{th} triangulation $\mathcal{T}_{h^{(j)}}$ with mesh function $h^{(j)}$, compute the corresponding finite element approximation $U^{(j)}$.
3. Compute the corresponding residuals $R_1(U^{(j)})$ and $R_2(U^{(j)})$ and check whether or not (77.20) holds. If it does, stop.
4. Find a new triangulation $\mathcal{T}_{h^{(j+1)}}$ with mesh function $h^{(j+1)}$ and with a minimal number of nodes such that $C_i \|h^{(j+1)} R(U^{(j)})\| \leq \text{TOL}$, and then proceed to #2.

The success of this iteration hinges on the mesh modification strategy used to perform step #4. A natural strategy for error control based on the L_2 norm uses the *principle of equidistribution* of the error in which we try to equalize the contribution from each element to the integral defining the L_2 norm. The rationale is that refining an element with large contribution to the error norm gives a large pay-off in terms of error reduction per new degree of freedom.

In other words, the approximation computed on the optimal mesh $\mathcal{T}_h$ in terms of computational work satisfies

$$\|\nabla e\|^2_{L_2(K)} \approx \frac{\text{TOL}^2}{M} \quad \text{for all } K \in \mathcal{T}_h,$$

where M is the number of elements in $\mathcal{T}_h$. Based on (77.18), we would therefore like to compute the triangulation at step #4 so that

$$C_i^2 \big(\big\|h^{(j+1)} R\big(U^{(j+1)}\big)\big\|^2_{L_2(K)} \approx \frac{\text{TOL}^2}{M^{(j+1)}} \quad \text{for all } K \in \mathcal{T}_{h^{(j+1)}}, \tag{77.21}$$

where $M^{(j+1)}$ is the number of elements in $\mathcal{T}_{h^{(j+1)}}$. However, (77.21) is a nonlinear equation, since we don't know $M^{(j+1)}$ and $U^{(j+1)}$ until we have chosen the triangulation. Hence, we replace (77.21) by

$$C_i^2 \big(\big\|h^{(j+1)} R\big(U^{(j)}\big)\big\|^2_{L_2(K)} \approx \frac{\text{TOL}^2}{M^{(j)}} \quad \text{for all } K \in \mathcal{T}_h{}^{(j+1)}, \tag{77.22}$$

and use this formula to compute the new mesh size $h^{(j+1)}$.

There are several questions that need to be answered about the process described here, including: how much efficiency is lost by replacing (77.19) by (77.20)? In other words, how much bigger is the right-hand side of (77.18) than the left-hand? Does the iterative process #1–#4 converge to a solution of the minimization problem? How should the initial triangulation $\mathcal{T}_{h^{(0)}}$ be chosen and how does this affect the convergence of the adaptive procedure? Is the approximation (77.22) justified? We address these issues in the advanced companion volume.

77.10 An example

We want to compute the the solution

$$u(x) = \frac{a}{\pi}\,\exp\bigl(-a(x_1^2 + x_2^2)\bigr), \quad a = 400,$$

of Poisson's equation $-\Delta u = f$ on the square $(-.5,.5)\times(-.5,.5)$ with $f(x)$ being the following "approximate delta function":

$$f(x) = \frac{4}{\pi}\,a^2\,\bigl(1 - ax_1^2 - ax_2^2\bigr)\,\exp\bigl(-a(x_1^2 + x_2^2)\bigr),$$

We plot f in Fig. 77.8 (note the vertical scale), together with the initial mesh with 224 elements. The adaptive algorithm took 5 steps to achieve

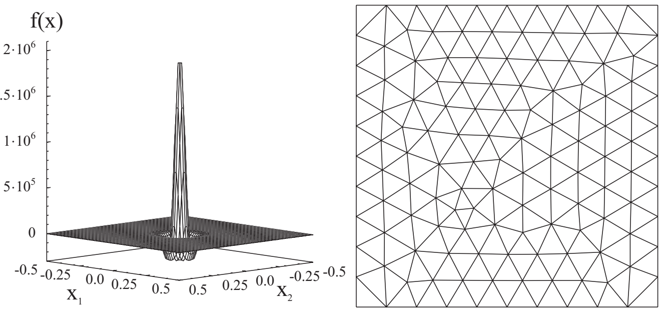

FIGURE 77.8. The approximate delta forcing function f and the initial mesh used for the finite element approximation.

an estimated .5% relative error. We plot the final mesh together with the associated finite element approximation in Fig. 77.9. The algorithm produced meshes with 224, 256, 336, 564, 992, and 3000 elements respectively.

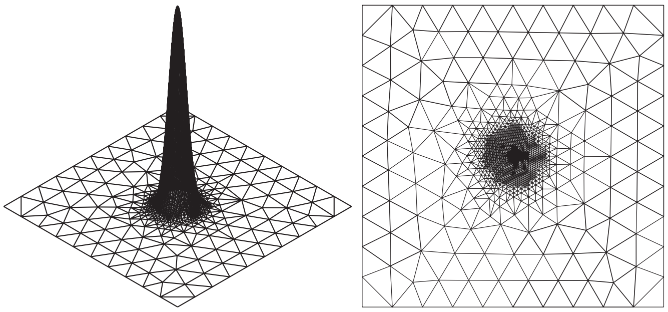

FIGURE 77.9. The finite element approximation with a relative error of .5% and the final mesh used to compute the approximation. The approximation has a maximum height of roughly 5.

77.11 Extension to non-homogeneous Dirichlet boundary conditions

We now consider Poisson's equation with non-homogeneous Dirichlet boundary conditions:

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = g & \text{on } \Gamma, \end{cases} \tag{77.23}$$

where g is the given boundary data.

We compute a finite element approximation on a triangulation $\mathcal{T}_h$, where we now also include the nodes on the boundary, denoting the internal nodes by $\mathcal{N}_h$ as above and the set of nodes on the boundary by $\mathcal{N}_b$. We compute an approximation U of the form

$$U = \sum_{N_j \in \mathcal{N}_b} \xi_j \varphi_j + \sum_{N_j \in \mathcal{N}_h} \xi_j \varphi_j, \tag{77.24}$$

where φ_j denotes the basis function corresponding to node N_j in an enumeration $\{N_j\}$ of all the nodes, and, because of the boundary conditions, $\xi_j = g(N_j)$ for $N_j \in \mathcal{N}_b$. Thus the boundary values of U are given by g on Γ and only the coefficients of U corresponding to the interior nodes remain to be found. To this end, we substitute (77.24) into the variational formulation of (77.23) with the test functions being all the basis functions for the internal nodes, and we then get the following a square system of linear equations for the unknown coefficients of U:

$$\sum_{N_j \in \mathcal{N}_h} \xi_j (\nabla \varphi_j, \nabla \varphi_i) = (f, \varphi_i) - \sum_{N_j \in \mathcal{N}_b} g(N_j)(\nabla \varphi_j, \nabla \varphi_i), \quad N_i \in \mathcal{N}_h.$$

where the terms with known boundary values of U are shifted to the right hand side as data.

77.12 The L-shaped membrane

We present an example that shows the performance of the adaptive algorithm on a problem with a *boundary singularity* with the derivatives of the exact solution being infinite at a corner of the boundary. We consider the Laplace equation in an L-shaped domain that has a non-convex corner at the origin satisfying homogeneous Dirichlet boundary conditions at the sides meeting at the origin and non-homogeneous conditions on the other sides, see Fig. 77.10. We choose the boundary conditions so that the exact solution is $u(r, \theta) = r^{2/3}\sin(2\theta/3)$ in polar coordinates (r, θ) centered at the origin, which has a typical singularity of a corner problem:

$$\frac{\partial u}{\partial r}(r, \theta) = \frac{2}{3} r^{-1/3}\sin(2\theta/3),$$

which tends to infinity as r tends to zero (unless $\theta = 0$ or $\theta = \frac{3\pi}{2}$).

We use the knowledge of the exact solution to evaluate the performance of the adaptive algorithm.

We compute using an adaptive FEM-solver with energy norm control based on (77.18) to achieve an error tolerance of TOL$=$ .005 using h refinement mesh modification. In Fig. 77.10, we show the initial mesh $\mathcal{T}_{h^{(0)}}$ with 112 nodes and 182 elements. In Fig. 77.11, we show the level curves

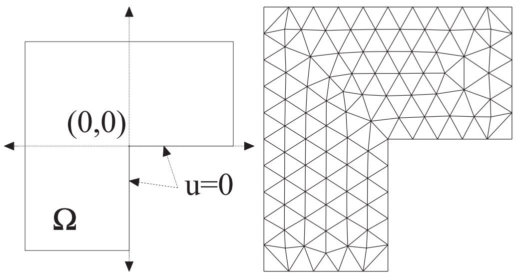

FIGURE 77.10. The L-shaped domain and the initial mesh.

of the solution and the final mesh with 295 nodes and 538 elements that achieves the desired error bound. The interpolation constant was set to $C_i = 1/8$. The quotient between the estimated and true error on the final mesh was 1.5.

Since the exact solution is known in this example, we can also use the a priori error estimate to determine a mesh that gives the desired accuracy. We do this by combining the a priori error estimate (77.16) and the principle of equidistribution of error to determine $h(r)$ so that $C_i\|hD^2u\| = $ TOL while keeping h as large as possible (and keeping the number of elements at a minimum). Since $D^2u(r) \approx r^{-4/3}$, as long as $h \leq r$, that is up to the

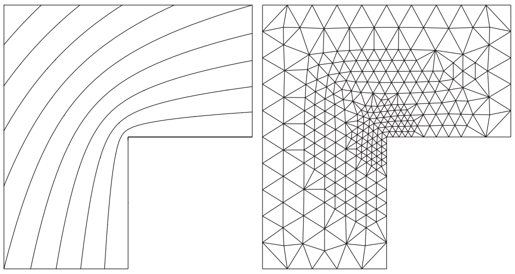

FIGURE 77.11. Level curves of the solution and final adapted mesh on the L-shaped domain.

elements touching the corner, we determine that

$$\left(hr^{-4/3}\right)^2 h^2 \approx \frac{\mathrm{TOL}^2}{M} \qquad \text{or} \quad h^2 = \mathrm{TOL}\, M^{-1/2} r^{4/3},$$

where M is the number of elements and h^2 measures the element area. To compute M from this relation, we note that $M \approx \int_\Omega h^{-2}\, dx$, since the number of elements per unit area is $O(h^{-2})$, which gives

$$M \approx M^{1/2} \mathrm{TOL}^{-1} \int_\Omega r^{-4/3}\, dx.$$

Since the integral is convergent (prove this), it follows that $M \propto \mathrm{TOL}^{-2}$, which implies that $h(r) \propto r^{1/3}\,\mathrm{TOL}$. Note that the total number of unknowns, up to a constant, is the same as that required for a smooth solution without a singularity, namely TOL^{-2}. This depends on the very local nature of the singularity in the present case. In general, of course solutions with singularies may require a much larger number of elements than smooth solutions do.

77.13 Robin and Neumann boundary conditions

Next, we consider Poisson's equation with homogeneous Dirichlet conditions on part Γ_1 of the boundary and non-homogeneous Robin conditions on the remaining part of the boundary Γ_2:

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma_1, \\ \partial_n u + \kappa u = g & \text{on } \Gamma_2, \end{cases} \tag{77.25}$$

where $\kappa \geq 0$ is a given coefficient, and f and g are given data. Setting $\kappa = 0$ gives the Neumann condition $\partial_n u + \kappa u = g$. To find a variational formulation, we multiply the Poisson equation by a test function v satisfying the homogenous Dirichlet boundary condition, integrate over Ω, and use Green's formula to move derivatives from u to v:

$$\begin{aligned}(f, v) &= -\int_\Omega \Delta u\, v\, dx = \int_\Omega \nabla u \cdot \nabla v\, dx - \int_\Gamma \partial_n u v\, ds \\ &= \int_\Omega \nabla u \cdot \nabla v\, dx + \int_{\Gamma_2} \kappa u v\, ds - \int_{\Gamma_2} g v\, ds,\end{aligned}$$

where we use the boundary conditions to rewrite the boundary integral. We are thus led to the following cG(1) FEM based on a space V_h of continuous piecewise linear functions vanishing on Γ_1: find $U \in V_h$ such that

$$(\nabla U, \nabla v) + \int_{\Gamma_2} \kappa U v\, ds = (f, v) + \int_{\Gamma_2} g v\, ds \quad \text{for all } v \in V_h. \tag{77.26}$$

We recall that boundary conditions like the Dirichlet condition that are enforced explicitely in the choice of the space V_h are called *essential boundary conditions*. Boundary conditions like the Robin condition that are implicitly contained in the weak formulation are called *natural boundary conditions*. (To remember that we must assume essential conditions: there are two "ss" in assume and essential.)

Note that the stiffness matrix and load vector related to (77.26) contain contributions from both integrals over Ω and Γ_2 related to the basis functions corresponding to the nodes on the boundary Γ_2.

To illustrate, we compute the solution of Laplace's equation with a combination of Dirichlet, Neumann and Robin boundary conditions on the domain shown in Fig. 77.12 using an adaptive FEM-solver. We show the boundary conditions in the illustration. The problem models e.g. stationary heat flow around a hot water pipe in the ground. We show the mesh

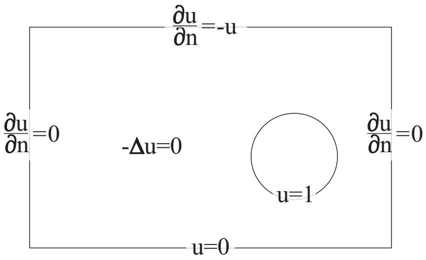

FIGURE 77.12. A problem with Robin boundary conditions.

used to compute the approximation so that the error in the L_2 norm is smaller than .0013 together with a contour plot of the approximation in Fig. 77.13. We notice that the level curves are parallel to a boundary with a homogeneous Dirichlet condition, and orthogonal to a boundary with a homogeneous Neumann condition.

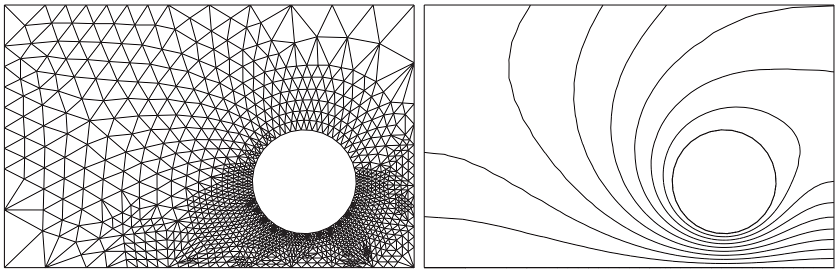

FIGURE 77.13. The adaptive mesh and contour lines of the approximate solution of the problem shown in Fig. 77.12 computed with error tolerance .0013.

77.14 Stationary convection-diffusion-reaction problems

We now consider the extension to a convection-diffusion-reaction problem of the form

$$\begin{aligned} -\nabla \cdot (a\nabla u) + \nabla \cdot (ub) + cu = f \quad &\text{in } \Omega, \\ a\frac{\partial u}{\partial n} + \kappa u = g \quad &\text{on } \Gamma. \end{aligned} \tag{77.27}$$

with Robin boundary conditions, where f and g are given data, and $a > 0$, b, c and $\kappa \geq 0$ are given coefficients, and Ω is a given domain in $\mathbb{R}^2$ with boundary Γ. The term cu models absorption if $c \geq 0$ and production $c < 0$.

Let V_h be the space of continuous piecewise linear functions on a triangulation of Ω with no restriction on the nodal values on the boundary. The cG(1) FEM for (77.27) takes the form: Find $U \in V_h$ such that

$$\int_\Omega a\nabla U\cdot\nabla v\,dx + \int_\Omega \nabla\cdot(ub)v\,dx + \int_\Omega cuv\,dx + \int_\Gamma \kappa uv\,ds = \int_\Omega fv\,dx + \int_\Gamma gv\,ds, \tag{77.28}$$

for all $v \in V_h$. Note that the extension to include the terms $\nabla \cdot (ub)$ and cu is very natural and that the corresponding terms in the variational formulation are obtained by multiplying by the test function v without any partial integration. For the term $-\nabla \cdot (a\nabla u)$ we note that multiplication by $v(x)$ and integration over Ω gives using the Divergence theorem

$$-\int_\Omega \nabla \cdot (a\nabla u)v\,dx = \int_\Omega a\nabla U \cdot \nabla v\,dx - \int_\Gamma a\frac{\partial u}{\partial n}v\,ds$$

and the variational formulation results from replacing $-a\frac{\partial u}{\partial n}v\,ds$ by $ku - g$ using the Robin boundary condition.

The matrix equation corresponding to (77.28) has a banded and sparse stiffness matrix, but the symmetry is lost if $b \neq 0$, as is evident from the presence of the non-symmetric term $\int_\Omega \nabla \cdot (ub)v\,dx$. The non-symmetry of the convection term eliminates the best approximation property of FEM, but FEM still may give good results. If $|b| > \frac{a}{h}$, where $h(x)$ is the mesh size, and the exact solution is non-smooth with rapid variation, then the FEM-solution may exhibit spurious oscillations. In such cases the cG(1)-method (77.28) will have to be modified, see the CDE-book. If $c < 0$ then solutions may be non-unique corresponding to non-zero solutions (eigen-functions) of the homogeneous problem $-\nabla \cdot (a\nabla u) + \nabla \cdot (ub) + cu = 0$.

77.15 Time-dependent convection-diffusion-reaction problems

We now consider the time-dependent analog of (77.27), that is the problem

$$\begin{aligned} \dot{u} - \nabla \cdot (a\nabla u) + \nabla \cdot (ub) + cu = f \quad &\text{in } \Omega \times (0,T], \\ a\frac{\partial u}{\partial n} + \kappa u = g \quad &\text{on } \Gamma \times (0,T], \\ u(\cdot) = u^0 \quad &\text{in } \Omega, \end{aligned} \tag{77.29}$$

where $[0,T]$ is a given time interval, and u^0 a given initial value. For the time discretization we may use e.g. dG(0) or cG(1) on a subdivision $0 = t_0 < t_1 < \cdots < t_N = T$ into time intervals $I_n = (t_{n-1}, t_n]$ with time steps $k_n = t_n - t_{n-1}$. Using dG(0) we seek $U^n \in V_h$ for $n = 1, ..., N$, such that for $n = 1, ..., N$,

$$\begin{aligned} &\int_\Omega U^n v\,dx + \int_{\Omega\times I_n} a\nabla U^n \cdot \nabla v\,dx\,dt \\ &+ \int_{\Omega\times I_n} \nabla \cdot (U^n b) v\,dx\,dt + \int_{\Omega\times I_n} cU^n v\,dx\,dt + \int_{\Gamma\times I_n} \kappa U^n v\,ds\,dt \\ &= \int_\Omega U^{n-1} v\,dx + \int_{\Omega\times I_n} fv\,dx\,dt + \int_{\Gamma\times I_n} gv\,ds\,dt, \end{aligned} \tag{77.30}$$

for all $v \in V_h$, where $U^0 = u^0$. The corresponding discrete system for U^n takes the form

$$M\xi^n + k_n A_n \xi^n = M\xi^{n-1} + k_n b^n$$

where the vector ξ^n contains the nodal values of $U^n \in V_h$, M is the mass matrix related to V_h, A_n is the relevant stiffness matrix connected to the convection-diffusion-reaction terms, and b^n the relevant load vector.

77.16 The wave equation

We now consider the extension to the wave equation with homogeneous Dirichlet boundary conditions:

$$\begin{aligned} \ddot{u} - \Delta u = f \quad & \text{in } \Omega \times (0,T], \\ u = 0 \quad & \text{on } \Gamma \times (0,T], \\ u = u^0, \quad \dot{u} = \dot{u}^0 \quad & \text{in } \Omega, \end{aligned} \tag{77.31}$$

where u^0 andf $\dot{u}^0$ are given initial conditions. As above we let V_h be the set of piecewise linears functions on a triangulation of Ω satisfying the homogeneous Dirichlet boundary conditions, and we let $0 = t_0 < t_1 < \cdots < t_N = T$ be a subdivision of $[0,T]$ into time intervals $I_n = (t_{n-1}, t_n]$ with time steps $k_n = t_n - t_{n-1}$. We appply cG(1) in space and cG(1) in time and seek a discrete solution U in the space of functions W_h spanned by the functions

$$v(x,t) = \sum_{n=0}^{N} \sum_{j=1}^{M} \eta_j^n \varphi_j(x) \psi_n(t),$$

where $\{\varphi_j(x)\}_{j=1}^M$ is a basis for V_h, and $\{\psi_n(t)\}_{n=0}^N$ is a basis for the space of continuous piecewise linear functions on the subdivision $0 = t_0 < t_1 < \cdots < t_N = T$. The corresponding discrete system takes the following explicit form if mass lumping is used in space as well as time and the time step is constant $k_n = k$:

$$\xi^{n+1} = 2\xi^n - \xi^{n-1} + k^2 A \xi^n \quad \text{for } n = 1, ..., N-1,$$

with appropriate starting values ξ^0 and ξ^1 computed from the initial conditions, and A the relevant stiffness matrix related to the Laplacian.

77.17 Examples

We present some examples of systems of non-linear reaction-diffusion-convection equations (77.27) arising in physics, chemistry and biology of the form

$$\dot{u}_i - \nabla \cdot (a_i \nabla u_i) + \nabla \cdot (u_i b_i) + c_i u_i = f_i(u_1, ..., u_d) \quad \text{in } \Omega \times I,\, i = 1, ..., d, \tag{77.32}$$

where the $a_i > 0$, b_i and c_i are given coefficients, and the $f_i : \mathbb{R}^d \to \mathbb{R}$. These systems may be solved numerically by a direct extension of the cG(1) method in space and time presented above. We will return in detail to this issue below. In all the examples, a is a positive constant.

EXAMPLE 77.1. *The bistable equation for ferro-magnetism*

$$\dot{u} - a\Delta u = u - u^3. \tag{77.33}$$

EXAMPLE 77.2. *Superconductivity of fluids*

$$\begin{aligned} \dot{u}_1 - a\Delta u_1 &= (1 - |u|^2)u_1, \\ \dot{u}_2 - a\Delta u_2 &= (1 - |u|^2)u_2. \end{aligned} \tag{77.34}$$

EXAMPLE 77.3. *Flame propagation*

$$\begin{aligned} \dot{u}_1 - a\Delta u_1 &= -u_1 e^{-\alpha_1/u_2}, \\ \dot{u}_2 - a\Delta u_2 &= \alpha_2 u_1 e^{-\alpha_1/u_2}, \end{aligned} \tag{77.35}$$

where $\alpha_1, \alpha_2 > 0$ are constants.

EXAMPLE 77.4. *Interaction of two species*

$$\begin{aligned} \dot{u}_1 - a\Delta u_1 &= u_1 M(u_1, u_2), \\ \dot{u}_2 - a\Delta u_2 &= u_2 N(u_1, u_2), \end{aligned} \tag{77.36}$$

where $M(u_1, u_2)$ and $N(u_1, u_2)$ are given functions describing various situations such as (i) predator-prey ($M_{u_2} < 0$, $N_{u_1} > 0$) (ii) competing species ($M_{u_2} < 0$, $N_{u_1} < 0$) and (iii) symbiosis ($M_{u_2} > 0$, $N_{u_1} > 0$).

EXAMPLE 77.5. *Morphogenesis of patterns (zebra)*

$$\begin{aligned} \dot{u}_1 - a\Delta u_1 &= -u_1 u_2^2 + \alpha_1(1 - u_1), \\ \dot{u}_2 - a\Delta u_2 &= u_1 u_2^2 - (\alpha_1 + \alpha_2)u_2. \end{aligned} \tag{77.37}$$

EXAMPLE 77.6. *Belousov-Zhabotinski reaction in chemical kinetics*

$$\begin{aligned} \dot{u}_1 - a\Delta u_1 &= \alpha_1(u_2 - u_1 u_2 + u_1 - \alpha_2 u_2^2), \\ \dot{u}_2 - a\Delta u_2 &= \alpha_1^{-1}(\alpha_3 u_3 - u_2 - u_1 u_2). \\ \dot{u}_3 - a\Delta u_3 &= \alpha_4(u_1 - u_3), \end{aligned} \tag{77.38}$$

where $\alpha \approx 10^2$, $\alpha_2 \approx 10^{-2}$, $\alpha_3 \approx 1$, $\alpha_4 \approx 10^{-1}$.

Chapter 77 Problems

77.1. Compute the coefficients of the mass matrix M on the standard triangulation of the square of mesh size h. Hint: it is possible to use quadrature based on the midpoints of the sides of the triangle because this is exact for quadratic functions. The diagonal terms are $h^2/2$ and the off-diagonal terms are all equal to $h^2/12$. The sum of the elements in a row is equal to h^2.

77.2. Compute the stiffness matrix for cG(1) for the problem $-\Delta u = 1$ in $\Omega = (0,1) \times (0,1)$ with $u = 0$ on the side with $x_2 = 0$ and $\partial_n u + u = 1$ on the other three sides of Ω using the standard triangulation. Note the contribution to the stiffness matrix from the nodes on the boundary.

77.3. Describe the sparsity pattern of the stiffness matrices A for the Poisson equation with homogeneous Dirichlet data on the unit square corresponding to the continuous piecewise linear finite element method on the standard triangulation using the three numbering schemes pictured in Fig. 77.14.

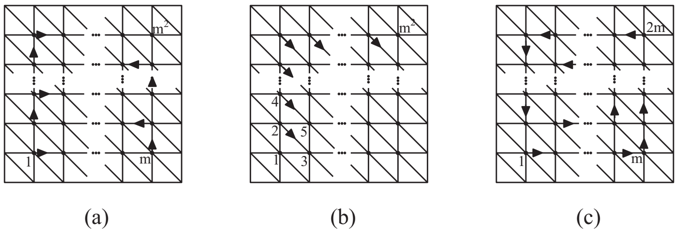

FIGURE 77.14. Three node numbering schemes for the standard triangulation of the unit square.

77.4. Compute the load vector for $f(x) = x_1 + x_2^2$ on the standard triangulation of the unit square using exact integration and the lumped mass (trapezoidal rule) quadrature.

77.5. Write a code to solve $A\xi = b$ using both the Jacobi and Gauss-Seidel iteration methods, making use of the sparsity of A in storage and operations. Compare the convergence rate of the two methods using the result from a direct solver as a reference value.

77.6. Write a code to solve the system $A\xi = b$ with A a band matrix.

77.7. Compute the stiffness matrix for the Poisson equation with homogeneous Dirichlet boundary conditions for (a) the *union jack* triangulation of a square shown in Fig. 77.15 and (b) the triangulation of triangular domain shown in Fig. 77.15.

77.8. Compute the discrete equations for the finite element approximation for $-\Delta u = 1$ on $\Omega = (0,1) \times (0,1)$ with boundary conditions $u = 0$ for $x_1 = 0$, $u = x_1$ for $x_2 = 0$, $u = 1$ for $x_1 = 1$ and $u = x_1$ for $x_2 = 1$ using the standard triangulation (Fig. 77.1).

77.9. (a) Show that the element stiffness matrix (77.12) for the linear polynomials on a triangle K with vertices at $(0,0)$, $(h,0)$, and $(0,h)$ numbered 1, 2 and 3, is given by

$$\begin{pmatrix} 1 & -1/2 & -1/2 \\ -1/2 & 1/2 & 0 \\ -1/2 & 0 & 1/2 \end{pmatrix}.$$

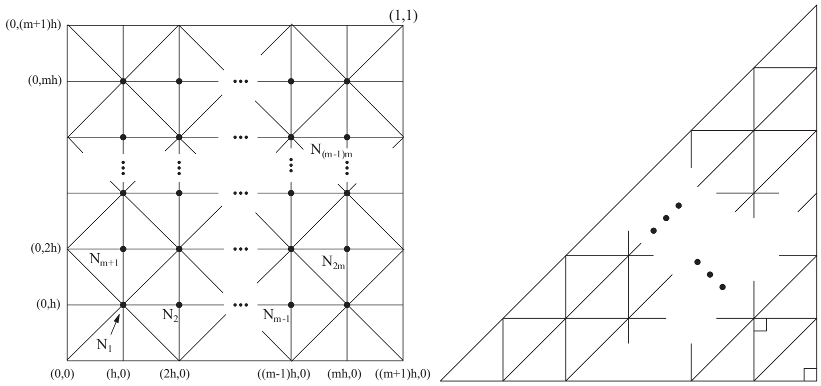

FIGURE 77.15. The "union jack" triangulation of the unit square and a uniform triangulation of a right triangle.

(b) Use this result to verify the formula computed for the stiffness matrix A for the continuous piecewise linear finite element method for the Poisson equation with homogeneous boundary conditions on the unit square using the standard triangulation. (c) Compute the element stiffness matrix for a triangle K with nodes $\{a^i\}$.

77.10. Compute the asymptotic operations count for the direct solution of the system $A\xi = b$ using the three A computed in Problem 77.3.

77.11. Apply the finite element method with piecewise linear approximation to the Poisson equation in three dimensions with a variety of boundary conditions. Compute the stiffness matrix and load vector in some simple cases.

77.12. Derive a priori error bound in the energy norm for cG(1) FEM for Poisson's equation with Robin boundary conditions. Generalize to problems of the form $-\nabla \cdot (a\nabla u) + cu = f$, where $a(x) > 0$ and $c \geq 0$.

77.13. Derive a posteriori error bound in the energy norm for cG(1) FEM for Poisson's equation with Robin boundary conditions. Generalize to problems of the form $-\nabla \cdot (a\nabla u) + cu = f$, where $a(x) > 0$ and $c \geq 0$.

77.14. Implement adaptive energy norm error control for cG(1) for Poisson's equation based on an a posteriori error estimate.

77.15. Find an exact solution of the L-shaped membrane problem with the Dirichlet condition replaced by a Neumann condition on one of the sides meeting at $\frac{3\pi}{2}$ corner. What is the nature of the singularity?

77.16. Let $\omega(x)$ be a positive weight function defined on the domain $\Omega \subset \mathbb{R}^2$. Assume that the mesh function $h(x)$ minimizes the integral $\int_\Omega h^2(x)\omega(x)\,dx$ under the constraint $\int_\Omega h^{-2}(x)\,dx = N$, where N is a given positive integer. Prove that $h^4(x)\omega(x)$ must be constant. Interpret the result as equidistribution in the context of error control. Hint: argue that $h^4(x)\omega(x)$ is the gain adding one more node.