Chapter 10

Finite Elements

Until now, our development of multigrid methods has relied on finite differences. However, there are several other discretization methods, all of which have their place in computation. We now turn to the finite element method. This approach may seem more abstract and less direct at the outset, but its use of variational properties provides a powerful framework that is well suited to developing and analyzing multigrid methods. It is impossible to cover finite elements completely in this short chapter, so we focus instead on surveying the critical ideas, leaving the important details to the exercises and further study.

Said simply, finite difference methods replace the problem domain by a grid and produce a vector whose components are approximations to the solution at the grid points. On the other hand, finite element methods partition the problem domain into subregions and produce a simple function in each subregion that approximates the solution. Finite element methods can be applied to two major categories of problems: differential equations and functional minimization. When the differential equation is defined in terms of a self-adjoint linear operator and the functional is defined appropriately, these two categories coincide. To exploit this fortunate duality between differential equations and functional minimization, we focus on the self-adjoint case.

Let Ω be a suitably nice bounded domain in the plane with boundary $\partial\Omega$. Consider the following Poisson equation with homogeneous Dirichlet boundary conditions:

$$\begin{aligned} Lu \equiv -u_{xx} - u_{yy} &= f \qquad \text{in } \Omega, \\ u &= 0 \qquad \text{on } \partial\Omega. \end{aligned}$$

We first need to introduce some new tools and notation. Most of our time will be spent in the space $L^2(\Omega)$ of square integrable functions on Ω (functions that satisfy $\int_\Omega u^2 \, d\Omega < \infty$) equipped with the $L^2(\Omega)$ inner product $(\cdot,\cdot)$ defined by

$$(u, v) = \int_\Omega uv \, d\Omega.$$

Because we will be taking partial derivatives, it is not enough for the functions we consider just to be $L^2(\Omega)$ integrable. We need to know that they are differentiable. We will be vague about this for the moment and just assume that the functions reside in a subspace $\mathcal{H} \subset L^2(\Omega)$ that contains *suitably differentiable* functions. We also assume that the functions in $\mathcal{H}$ vanish on $\partial\Omega$ so that they satisfy the boundary conditions.

There are two important properties of the operator L that we need (Exercise 2):

- L is *self-adjoint* in the sense that $(Lu, v) = (u, Lv)$ for any $u, v \in \mathcal{H}$; and
- L is *positive* in the sense that $(Lu, u) > 0$ for any nonzero $u \in \mathcal{H}$.

It is worth noting that if L is a matrix, u and v are vectors, and $(\cdot,\cdot)$ is the usual Euclidean inner product, then self-adjoint and positive correspond to symmetric and positive definite, respectively (Exercise 3).

With these properties in hand, we can demonstrate the crucial duality mentioned earlier: solving the differential equation $Lu = f$ is *formally* equivalent to minimizing the functional

$$F(u) \equiv \frac{1}{2}(Lu, u) - (f, u)$$

over $u \in \mathcal{H}$. The functional F is *quadratic* in u and it has instructive scalar and matrix analogues (Exercises 4 and 5). The minimization of F is often written in the compact notation

$$u = \operatorname{argmin}_{v\in\mathcal{H}} F(v), \tag{10.1}$$

which means "find the argument that minimizes F over all functions in $\mathcal{H}$." This link between the differential equation and the minimization problem lies at the heart of the finite element formulation. It is so important that we should study it for a moment to understand it.

Suppose u is a candidate function for minimizing F and $v \neq 0$ is any other function in $\mathcal{H}$. Consider the value of F at the point $u + v$. Using the linearity of L and the bilinearity of the inner product, we see that

$$\begin{aligned} F(u+v) &= \frac{1}{2}(L(u+v), u+v) - (f, u+v) \\ &= \frac{1}{2}\left((Lu,u) + (Lu,v) + (u,Lv) + (Lv,v)\right) - (f,u) - (f,v) \\ &= F(u) + \frac{1}{2}\left((Lu,v) + (u,Lv) + (Lv,v)\right) - \langle f, v\rangle . \end{aligned}$$

Because L is self-adjoint, we know that $(u, Lv) = (Lu, v)$. This allows us to simplify the above expression further:

$$\begin{aligned} F(u+v) &= F(u) + (Lu,v) - (f,v) + \frac{1}{2}(Lv,v) \\ &= F(u) + (Lu - f, v) + \underbrace{\frac{1}{2}(Lv,v)}_{\text{positive}} . \end{aligned}$$

The last term in this expression is positive because L is a positive operator. Therefore, we see that if $Lu = f$, then $F(u + v) \geq F(u)$ for all $v \in \mathcal{H}$, which means that u minimizes F over $\mathcal{H}$. Conversely, if u minimizes F, then it follows that $(Lu - f, v) = 0$ for all $v \in \mathcal{H}$, which means that

$$(Lu, v) = (f, v) \quad \text{for all } v \in \mathcal{H} . \tag{10.2}$$

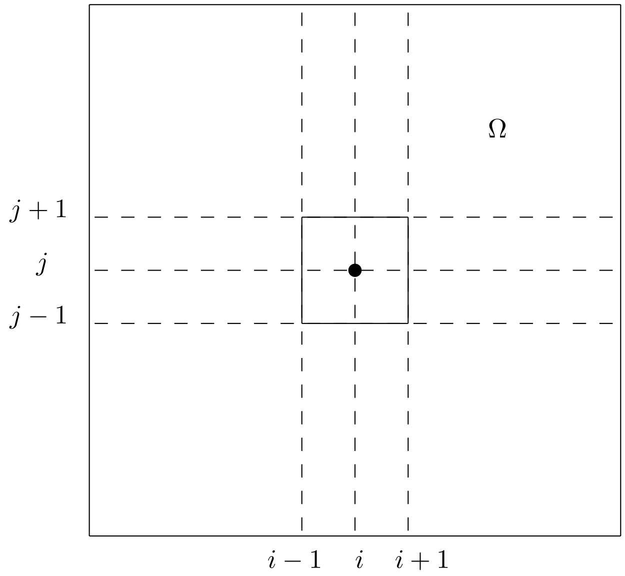

Figure 10.1: *A domain* Ω *showing the four elements surrounding grid point* (i, j).

This condition, which is useful for implementing finite elements, is essentially another way of saying that $Lu = f$. Thus, we have shown that the differential equation and minimization problem are essentially equivalent.

The finite element discretization of the problem can now be described. Suppose that the domain is the unit square, $\Omega = (0,1) \times (0,1)$, on which we place a uniform $(n+1) \times (n+1)$ grid, Ω^h, with mesh size $h = \frac{1}{n}$. As before, let (x_i, y_j) be the grid point with coordinates (ih, jh). However, we now focus not on the grid points, but on the square regions surrounding each grid point; these small regions are called *elements* (Fig. 10.1). Notice that there are four elements surrounding each grid point. The sets of four elements corresponding to neighboring grid points overlap in one or two elements. (Everything that follows can also be carried out on nonuniform grids.)

We begin with a common choice of approximating functions. Let H^h be the subspace of $\mathcal{H}$ consisting of *piecewise bilinear* functions u^h: each $u^h \in H^h$ is zero on $\partial\Omega$, continuous on Ω, and bilinear within each element. This means that a typical function in H^h has the form $u^h(x,y) = axy + bx + cy + d$ on each element, as depicted in Fig. 10.2. The level h discretization of (10.1) is

$$u^h = \text{argmin}_{v^h \in H^h} F(v^h). \qquad (10.3)$$

We have seen that the minimization problem on $\mathcal{H}$ is equivalent to solving (10.2) for $u \in \mathcal{H}$. This equivalence is also true on H^h. Thus, the discrete minimization problem (10.3) is equivalent to finding $u^h \in H^h$ so that

$$(Lu^h, v^h) \ = \ (f, v^h) \quad \text{for all } v^h \in H^h. \qquad (10.4)$$

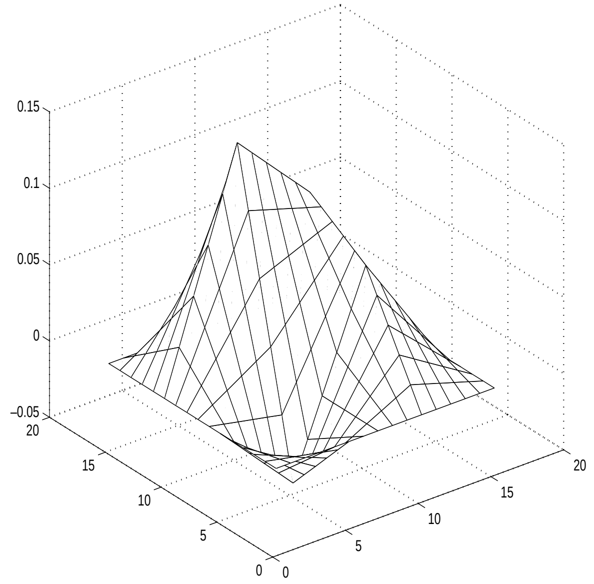

Figure 10.2: *A typical function in the space H^h used to approximate the solution to the problem. It is zero on $\partial\Omega$, continuous on Ω, and bilinear within each of the nine elements pictured.*

Finite Element Terminology. The terminology associated with finite element methods is not entirely standard and can get confusing. The discretization of the minimization problem is generally called the *Rayleigh–Ritz* formulation. The finite element method also has a *Galerkin* form, which starts not from the minimization problem (10.1), but from the equation $Lu = f$. As with Rayleigh–Ritz, the idea behind the Galerkin form is to approximate the solution $u \in \mathcal{H}$ by $u^h \in H^h$. However, the Galerkin approach requires the residual $Lu^h - f$ to be orthogonal to every $v^h \in H^h$. This condition also leads to (10.4). Thus, when L is a positive, self-adjoint, linear operator, and F is quadratic, the Rayleigh–Ritz and Galerkin formulations are essentially the same.

The practical solution of (10.4) requires several additional considerations. First, we need to choose a *local* basis for H^h consisting of functions, each of which is nonzero on its own particular patch of four elements. Specifically, let $\epsilon^h_{i,j}(x, y)$ denote the piecewise bilinear function in H^h that is centered on an interior grid point (x_i, y_j); it has the value 1 at (x_i, y_j) and is zero at all other grid points (Fig. 10.3). Any $u^h \in H^h$ can then be expanded as

$$u^h(x, y) = \sum_{i,j=1}^{n-1} u^h_{i,j}\epsilon^h_{i,j}(x, y). \tag{10.5}$$

The *nodal value* $u^h_{i,j}$ gives the value of u^h at (x_i, y_j); for this reason, (10.5) is called a *nodal basis expansion.*

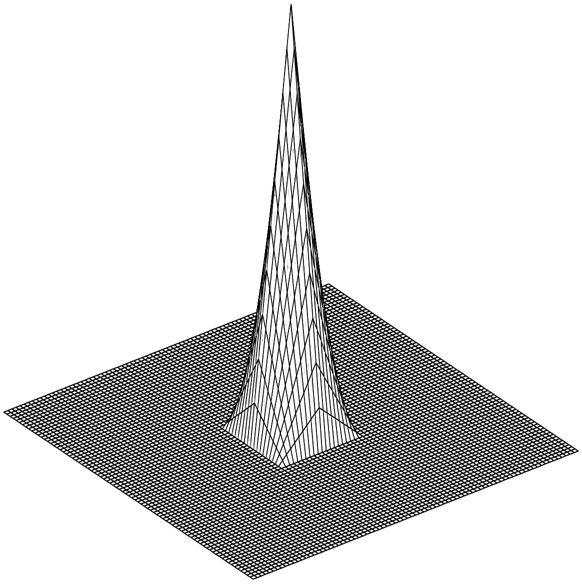

Figure 10.3: *A typical basis function $\epsilon^h_{i,j}(x,y)$ for the space H^h. It is nonzero, only on the patch of four elements that have (x_i, y_j) as a node.*

The next step is to substitute expansion (10.5) into expression (10.4). However, this task leads to a paradox: the operator L is second order and the approximating function u^h is linear in x and y, so $Lu^h = 0$ in each element! How can Lu^h be zero when we know generally that $(Lu^h, u^h) > 0$? The answer is that Lu^h is *not* zero on Ω. In fact, Lu^h does not even make sense on Ω because u^h is not suitably smooth: the first partials of u^h are piecewise smooth and therefore square integrable, but their discontinuities at the element boundaries prevent second derivatives from being square integrable. This means that we really cannot substitute (10.5) into (10.4). Fortunately, there is a way out of this dilemma: we just rewrite (10.4) in a form that requires fewer derivatives. This is done by applying the Gauss divergence theorem (the analogue of integrating by parts) to (Lu, v), where $u, v \in \mathcal{H}$, and using the fact that u and v vanish on $\partial\Omega$ (Exercise 1):

$$\begin{aligned}(Lu, v) &= \int_\Omega (-u_{xx} - u_{yy}) v \, d\Omega \\ &= \int_\Omega (u_x v_x + u_y v_y) \, d\Omega \\ &= (\nabla u, \nabla v) \,.\end{aligned}$$

This maneuver replaces the second-order derivatives in the problem by first-order derivatives. Making this replacement in (10.4), we are led to the *weak form*

$$(\nabla u^h, \nabla v^h) = (f, v^h) \quad \text{for all } v^h \in H^h. \tag{10.6}$$

Strictly speaking, this new problem is more general or *weaker* than the differential equation: a solution may exist, but it may not be twice differentiable, which the classical solution of $Lu = f$ requires. We stay with the weak form for the remainder of the chapter, so we are now free to be specific about what *suitably smooth* means: $\mathcal{H} \subset L^2(\Omega)$ is assumed to consist of functions that vanish on $\partial\Omega$ and that have first partials in $L^2(\Omega)$.

With the weak form of the problem, we can now substitute the expansion (10.5) into (10.6). Furthermore, we choose the so-called *trial* functions v^h to be the basis functions $\epsilon_{i,j}$. The result of this substitution is a linear system whose unknowns are the nodal values $u^h_{i,j}$. The matrix coefficients in this system are inner products of the form $(\nabla\epsilon^h_{i,j}, \nabla\epsilon^h_{k,\ell})$ and the right-side values are $(f, \epsilon^h_{i,j})$. Note that with our chosen bilinear basis functions, only neighboring basis functions overlap; thus, the inner products $(\nabla\epsilon^h_{i,j}, \nabla\epsilon^h_{k,\ell})$ are zero unless $k = i$ or $i \pm 1$ and $\ell = j$ or $j \pm 1$. Evaluating the nine inner products associated with the (i,j) patch results in a row of the *stiffness matrix* (Exercise 6) given by the stencil

$$A^h_{i,j} = \frac{1}{3}\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}. \tag{10.7}$$

This is the just the nine-point finite difference stencil scaled by $\frac{1}{3}$ (instead of the usual $\frac{1}{3h^2}$).

The inner products $(f, \epsilon^h_{i,j})$ involve the integrals

$$(f, \epsilon^h_{i,j}) = \int_{x_{i-1}}^{x_{i+1}} \int_{y_{i-1}}^{y_{i+1}} f\epsilon^h_{i,j}\, dxdy,$$

which are generally approximated numerically. The simplest numerical integration scheme amounts to replacing the function f by its value $f(x_i, y_j)$ (Exercise 7):

$$\begin{aligned} (f, \epsilon^h_{i,j}) &= \int_{x_{i-1}}^{x_{i+1}} \int_{y_{i-1}}^{y_{i+1}} f\epsilon^h_{i,j}\, dxdy \\ &\approx f(x_i, y_j) \int_{x_{i-1}}^{x_{i+1}} \int_{y_{i-1}}^{y_{i+1}} \epsilon^h_{i,j}\, dxdy \\ &= \frac{h^2}{4} f(x_i, y_j). \end{aligned} \tag{10.8}$$

When all rows of stiffness matrix (10.7) and corresponding right sides (10.8) are assembled, the result is the matrix equation

$$A^h\mathbf{u}^h = \mathbf{f}^h. \tag{10.9}$$

Here we have introduced the source vector $(\mathbf{f}^h)_{i,j} = \left(\frac{h^2}{4} f(x_i, y_j)\right)$ and the solution vector $(\mathbf{u}^h)_{i,j} = \left(u^h_{i,j}\right) \in \Omega^h$.

This discretization shows that finite elements and finite differences boil down to similar matrix problems. But it is important to keep in mind that the methodology used to obtain them is quite different. Practical implementation of a multigrid scheme for finite elements can be done easily if we are guided by—*and remain faithful to*—the basic principle of functional minimization. For example, it may seem convenient to rescale (10.9) by dividing both sides by h^2, but this would change the relationship between the various grids used in the multigrid scheme, and it could easily introduce conceptual errors into the overall process.

We develop the multigrid scheme in the abstract by focusing on the functions u^h that solve (10.3) and (10.4), rather than their nodal values $\mathbf{u}^h$. First consider relaxation, whose goal is to provide an inexpensive method for eliminating oscillatory errors in the approximation, v^h. We can do this by making local changes of the form

$$v^h \leftarrow v^h - s\epsilon^h_{i,j}, \tag{10.10}$$

where $s \in \mathbf{R}$ is a suitably chosen step size. But how should s be chosen? Our plan of being faithful to the minimization principle gives us the answer: choose the best step size in the sense that it minimizes the functional over all possible choices. The mathematical statement is

$$s = \text{argmin}_{t\in\mathbf{R}}\, F(v^h - t\epsilon^h_{i,j}). \tag{10.11}$$

This gives us the following relaxation scheme, which amounts to using (10.10) and (10.11) to sweep over the grid points:

For each $i, j = 1, 2, ..., n-1$:

compute $s = \text{argmin}_{t\in\mathbf{R}}\, F(v^h - t\epsilon^h_{i,j})$ and make the replacement $v^h \leftarrow v^h - s\epsilon^h_{i,j}$.

For our special functional F, this *coordinate relaxation* scheme is none other than Gauss–Seidel applied to (10.7) (Exercise 8).

The coarse-grid correction process is also easy to formulate in the abstract. We define the coarse-grid space $H^{2h} \subset H^h$ as the set of piecewise bilinear functions associated with the standard coarse grid Ω^{2h} formed by deleting the odd-numbered lines of Ω^h. The goal is to change the approximation v^h by a function $v^{2h} \in H^{2h}$ that approximates the presumably smooth error. The form of this correction is $v^h \leftarrow v^h + v^{2h}$. But how should v^{2h} be chosen? Again, being faithful to the minimization principle gives us the answer: choose the best coarse-grid correction in the sense that it minimizes the functional over all possible choices. The mathematical statement is

$$v^{2h} = \text{argmin}_{w^{2h}\in H^{2h}}\, F(v^h + w^{2h}). \tag{10.12}$$

The coarse-grid correction scheme is

Compute $v^{2h} = \text{argmin}_{w^{2h}\in H^{2h}}\, F(v^h + w^{2h})$ and set $v^h \leftarrow v^h + v^{2h}$.

This correction step, together with the coordinate relaxation scheme defined above, constitutes the core of the multigrid method. We can design practical algorithms based on this core just as we did in Chapter 3.

Our development of the finite element coarse-grid correction step appears simpler than it was for finite differences, because it comes naturally from the minimization principle. However, this simplicity is somewhat deceptive because it must still be expressed in terms of nodal vectors, and this requires intergrid transfer and coarse-grid operators. We will see that there are no choices that need to be made here: these operators are determined by the spaces and bases that we have already selected.

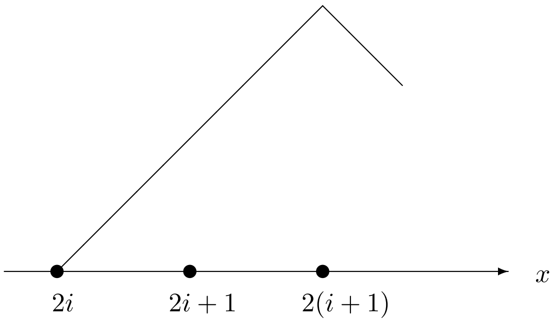

Figure 10.4: *Cross-section of the grid along a line of constant y showing the coarse-grid points $(2i, 2j)$ and $(2(i+1), 2j)$ and the intermediate fine-grid point $(2i+1, 2j)$. The coarse-grid element is linear in x between the two coarse-grid points.*

Consider first the process of adding an Ω^{2h} function, $v^{2h} \in H^{2h}$, to an Ω^h function. This process requires that we take the Ω^{2h} nodal representation for v^{2h},

$$v^{2h}(x,y) = \sum_{i,j=1}^{\frac{n}{2}-1} v_{i,j}^{2h} \epsilon_{i,j}^{2h}(x,y)\,,$$

and convert it to a Ω^h nodal representation,

$$v^{2h}(x,y) = \sum_{i,j=1}^{n} v_{i,j}^{h} \epsilon_{i,j}^{h}(x,y)\,.$$

In other words, we must find the coefficients $v_{i,j}^h$ that allow us to write v^{2h} as a function in H^h. We know that we can do this because H^{2h} is a subset of H^h, so v^{2h} must be in H^h. What we are looking for here is really just the interpolation operator I_{2h}^h that allows us to determine $\mathbf{v}^h = I_{2h}^h \mathbf{v}^{2h}$.

Consider the grid Ω^{2h} node with indices (i,j) that is located at the point $(i(2h), j(2h))$. This point is also the grid Ω^h node $((2i)h, (2j)h)$ with indices $(2i, 2j)$. Remembering that the coefficients in the nodal representations are just the function values at the nodes, we have

$$v_{2i,2j}^h = v^{2h}((2i)h, (2j)h) = v^{2h}(i(2h), j(2h)) = v_{i,j}^{2h}\,.$$

This says that $v_{2i,2j}^h = v_{i,j}^{2h}$.

Consider now the fine-grid node at point $((2i+1)h, (2j)h)$ (see Fig. 10.4). This node is halfway between nodes $((2i)h, (2j)h)$ and $(2(i+1)h, (2j)h)$ along a horizontal grid line. This line segment is part of a single coarse-grid element in which v^{2h} is bilinear and y is constant; thus, v^{2h} must be linear in x along this line. This leads us to conclude that

$$v_{2i+1,2j}^h = \frac{1}{2}\left(v_{i,j}^{2h} + v_{i+1,j}^{2h}\right)\,.$$

For similar reasons, we have

$$v_{2i,2j+1}^h = \frac{1}{2}\left(v_{i,j}^{2h} + v_{i,j+1}^{2h}\right)$$

and

$$v_{2i+1,2j}^h = \frac{1}{4}\left(v_{i,j}^{2h} + v_{i+1,j}^{2h} + v_{i,j+1}^{2h} + v_{i+1,j+1}^{2h}\right) .$$

As you can see, we have just reinvented the interpolation operator I_{2h}^h introduced in Chapter 3.

The next task is to determine the coarse-grid version of the operator A^h, which will eventually be called A^{2h}. It is most easily discovered by working with the nodal vectors and translating the minimization principle into matrix terms. As we have seen, discrete minimization principle (10.3) is equivalent to discrete equation (10.4), which in turn is equivalent to matrix equation (10.9). Similar reasoning shows that matrix equation (10.9) is equivalent to the matrix minimization principle

$$\mathbf{u}^h = \text{argmin}_{\mathbf{v}^h \in \Omega^h} F^h(\mathbf{v}^h)\,, \tag{10.13}$$

where

$$F^h(\mathbf{v}^h) \equiv \frac{1}{2}(A^h\mathbf{v}^h, \mathbf{v}^h) - (\mathbf{f}^h, \mathbf{v}^h)$$

(Exercise 5). Furthermore, similar reasoning also shows that coarse-grid correction problem (10.12) is equivalent to the matrix problem

$$\mathbf{v}^{2h} = \text{argmin}_{\mathbf{w}^{2h} \in \Omega^{2h}} F^h(\mathbf{v}^h + I_{2h}^h\mathbf{w}^{2h}).$$

We use this form of the coarse-grid correction problem to carry out an informative calculation. Using the properties of inner products, we have (read this one with a pencil and paper!)

$$\begin{aligned}
& F^h(\mathbf{v}^h + I_{2h}^h\mathbf{w}^{2h}) \\
= \; & \frac{1}{2}\left(A^h\left(\mathbf{v}^h + I_{2h}^h\mathbf{w}^{2h}\right), \mathbf{v}^h + I_{2h}^h\mathbf{w}^{2h}\right) - (\mathbf{f}^h, \mathbf{v}^h + I_{2h}^h\mathbf{w}^{2h}) \\
= \; & \frac{1}{2}(A^h\mathbf{v}^h, \mathbf{v}^h) + \frac{1}{2}(A^h I_{2h}^h\mathbf{w}^{2h}, \mathbf{v}^h) + \frac{1}{2}(\mathbf{v}^h, A^h I_{2h}^h\mathbf{w}^{2h}) \\
& + \frac{1}{2}(A^h I_{2h}^h\mathbf{w}^{2h}, I_{2h}^h\mathbf{w}^{2h}) - (\mathbf{f}^h, \mathbf{v}^h) - (\mathbf{f}^h, I_{2h}^h\mathbf{w}^{2h}) \\
= \; & \frac{1}{2}(A^h\mathbf{v}^h, \mathbf{v}^h) - (\mathbf{f}^h, \mathbf{v}^h) \\
& + \frac{1}{2}(A^h I_{2h}^h\mathbf{w}^{2h}, I_{2h}^h\mathbf{w}^{2h}) + (\mathbf{v}^h, A^h I_{2h}^h\mathbf{w}^{2h}) - (\mathbf{f}^h, I_{2h}^h\mathbf{w}^{2h}) \\
= \; & F^h(\mathbf{v}^h) + \frac{1}{2}\left(\underbrace{\left(I_{2h}^h\right)^T A^h I_{2h}^h}_{A^{2h}}\mathbf{w}^{2h}, \mathbf{w}^{2h}\right) - \left(\underbrace{\left(I_{2h}^h\right)^T}_{I_h^{2h}}\left(\mathbf{f}^h - A^h\mathbf{v}^h\right), \mathbf{w}^{2h}\right) .
\end{aligned}$$

Notice that the first underbraced item, $\left(I_{2h}^h\right)^T A^h I_{2h}^h$, is a matrix that operates on a coarse-grid vector. Furthermore, the role it plays in the inner product is analogous to the role played by A^h in the functional F^h. This suggests that we let $A^{2h} = \left(I_{2h}^h\right)^T A^h I_{2h}^h$, the coarse-grid version of A^h.

The second underbraced item, $\left(I_{2h}^h\right)^T$, must take a fine-grid vector into a coarse-grid vector; that is, it plays the role of a restriction operator. It therefore makes sense to let $I_h^{2h} = \left(I_{2h}^h\right)^T$.

With these operators identified, we may conclude the calculation. Letting $\mathbf{r}^h = \mathbf{f}^h - A^h\mathbf{v}^h$ and $\mathbf{f}^{2h} = \left(I_h^{2h}\right)\mathbf{r}^h$, we have

$$\begin{aligned} F^h(\mathbf{v}^h + I_{2h}^h\mathbf{w}^{2h}) &= F^h(\mathbf{v}^h) + \frac{1}{2}\left(A^{2h}\mathbf{w}^{2h}, \mathbf{w}^{2h}\right) - \left(\left(I_h^{2h}\right)\mathbf{r}^h, \mathbf{w}^{2h}\right) \\ &= F^h(\mathbf{v}^h) + F^{2h}(\mathbf{w}^{2h})\,. \end{aligned}$$

We should stand back and see what we have done. First, because $F^h(\mathbf{v}^h)$ is independent of $\mathbf{w}^{2h}$, the minimization of $F^h(\mathbf{v}^h + I_{2h}^h\mathbf{w}^{2h})$ is equivalent to the minimization of $F^{2h}(\mathbf{w}^{2h})$ over vectors $\mathbf{w}^{2h} \in \Omega^{2h}$; this is just the coarse-grid correction scheme introduced in Chapter 3.

Equally important, we have seen the variational properties of Chapter 5 emerge rather naturally through the minimization principle. They bear repeating:

$$\begin{aligned} A^{2h} &= I_h^{2h}A^hI_{2h}^h \qquad \text{(Galerkin property)}, \\ I_h^{2h} &= \left(I_{2h}^h\right)^T. \end{aligned}$$

Thus, the finite element coarse-grid correction scheme amounts to choosing bilinear interpolation, its transpose for restriction, and the Galerkin property for determining the coarse-grid operator. Except for scaling differences, we have reinvented the transfer operators, coarse-grid operators, and the coarse-grid correction scheme developed in the previous chapters. It is important to keep in mind that for positive self-adjoint problems, we now have the option of developing multigrid solvers directly from the differential equations, as we did before, or using the minimization principle, as we did here. The differential equations may provide more flexibility simply because you do not have to adhere to optimality, but the choice to follow the minimization principle may give guidance and assurance that is less forthcoming from the differential equations.

Exercises

1. **Weak form.** For functions u, v that are sufficiently smooth on Ω and vanish on $\partial\Omega$, show that $(Lu, v) = (\nabla u, \nabla v)$. Hint: First apply the divergence theorem to the product $v\nabla \cdot \mathbf{u}$, where $\mathbf{u}$ is a vector-valued function and v is a scalar. Then remember that $Lu = -\Delta u = -\nabla \cdot (\nabla u)$.

2. ***L* is self-adjoint and positive.** Let u, v be sufficiently smooth functions that vanish on $\partial\Omega$. Show that the Poisson operator $L = -\frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2}$ is self-adjoint in the sense that $(Lu, v) = (u, Lv)$. Hint: Use Exercise 1. Show that L is positive in the sense that $(Lu, u) > 0$ for $u \neq 0$. Hint: Show that $(Lu, u) = 0$ implies u is constant.

3. **Self-adjoint and positive for matrices.** Show that if L is a matrix, u and v are vectors, and $(\cdot, \cdot)$ is the usual vector dot product, then self-adjoint and positive correspond to symmetric and positive definite, respectively.

4. **Scalar analogue.** Show that if $L > 0$ and f are scalars and $(\cdot, \cdot)$ is just scalar multiplication, then (Lu, u) is a quadratic function. Furthermore, $\frac{1}{2}(Lu, u) - (f, u)$ is minimized by the solution of $Lu = f$, which is $u = f/L$.

5. **Matrix analogue.** Suppose that L is a symmetric positive-definite $n \times n$ matrix, u and f are n-vectors, and $(\cdot, \cdot)$ is the usual vector dot product. Show that $\frac{1}{2}(Lu, u) - (f, u)$ is minimized by the solution of $Lu = f$, which is $u = L^{-1}f$.

6. **Stiffness matrices.** For fixed indices i and j, show that the inner products $(\nabla\epsilon^h_{i,j}, \nabla\epsilon^h_{k,\ell})$ for $k = i$ or $i \pm 1$ and $\ell = j$ or $j \pm 1$ give the 3×3 stencil (10.7).

7. **Right-side inner products.** Show that

$$\int_{x_{i-1}}^{x_{i+1}} \int_{y_{i-1}}^{y_{i+1}} \epsilon^h_{i,j} f \; dxdy \approx \frac{h^2}{4} f(x_i, y_j),$$

where the region of integration is a single element.

8. **Gauss–Seidel.** Show that minimization problem (10.11) results in Gauss–Seidel applied to (10.9).

9. **Variational property.** Verify each of the steps in the simplification of $F^h(\mathbf{v}^h + I^h_{2h}\mathbf{w}^{2h})$ that led to the variational properties.