8 Boundary Value Problems for PDEs

Before we specialize to boundary value problems for PDEs — which only make sense for elliptic equations — we need to explain the terminology "elliptic".

8.1 Classification of Partial Differential Equations

We therefore consider general second-order partial differential equations (PDEs) of the form

$$Lu = au_{tt} + bu_{xt} + cu_{xx} + f = 0, \tag{75}$$

where u is an unknown function of x and t, and a, b, c, and f are given functions. If these functions depend only on x and t, then the PDE (75) is called *linear*. If a, b, c, or f depend also on u, u_x, or u_t, then the PDE is called *quasi-linear*.

Remark 1. The notation used in (75) suggests that we think of one of the variables, t, as time, and the other, x, as space.

2. In principle, we could also have second-order PDEs involving more than one space dimension. However, we limit the discussion here to PDEs with a total of two independent variables.

3. Of course, a second-order PDE can also be independent of time, and contain two space variables only (such as Laplace's equation). These will be the elliptic equations we are primarily interested in.

There are three fundamentally different types of second-order quasi-linear PDEs:

- If $b^2 - 4ac > 0$, then L is *hyperbolic*.
- If $b^2 - 4ac = 0$, then L is *parabolic*.
- If $b^2 - 4ac < 0$, then L is *elliptic*.

Example 1. The *wave equation*

$$u_{tt} = \alpha^2 u_{xx} + f(x,t)$$

is a second-order linear hyperbolic PDE since $a \equiv 1$, $b \equiv 0$, and $c \equiv -\alpha^2$, so that

$$b^2 - 4ac = 4\alpha^2 > 0.$$

2. The *heat* or *diffusion equation*

$$u_t = ku_{xx}$$

is a second-order quasi-linear parabolic PDE since $a = b \equiv 0$, and $c \equiv -k$, so that

$$b^2 - 4ac = 0.$$

3. For *Poisson's equation* (or *Laplace's equation* in case $f \equiv 0$)

$$u_{xx} + u_{yy} = f(x,y)$$

we use y instead of t. This is a second-order linear elliptic PDE since $a = c \equiv 1$ and $b \equiv 0$, so that

$$b^2 - 4ac = -4 < 0.$$

Remark In cases where a, b, and c depend on x, t, u, u_x, and u_t the classification of the PDEs above may even vary from point to point.

8.2 Boundary Value Problems for Elliptic PDEs: Finite Differences

We now consider a boundary value problem for an elliptic partial differential equation. The discussion here is similar to Section 7.2 in the Iserles book.

We use the following Poisson equation in the unit square as our model problem, i.e.,

$$\begin{aligned} \nabla^2 u = u_{xx} + u_{yy} &= f(x,y), \qquad (x,y) \in \Omega = (0,1)^2, \\ u(x,y) &= \phi(x,y), \qquad (x,y) \text{ on } \partial\Omega. \end{aligned} \tag{76}$$

This problem arises, e.g., when we want to determine the steady-state temperature distribution u in a square region with prescribed boundary temperature ϕ. Of course, this simple problem can be solved analytically using Fourier series.

However, we are interested in numerical methods. Therefore, in this section, we use the usual finite difference discretization of the partial derivatives, i.e.,

$$u_{xx}(x,y) = \frac{1}{h^2}\left[u(x+h,y) - 2u(x,y) + u(x-h,y)\right] + \mathcal{O}(h^2) \tag{77}$$

and

$$u_{yy}(x,y) = \frac{1}{h^2}\left[u(x,y+h) - 2u(x,y) + u(x,y-h)\right] + \mathcal{O}(h^2). \tag{78}$$

The computational grid introduced in the domain $\overline{\Omega} = [0,1]^2$ is now

$$(x_k, y_\ell) = (kh, \ell h), \quad k, \ell = 0, \ldots, m+1,$$

with *mesh size* $h = \frac{1}{m+1}$.

Using the compact notation

$$u_{k,\ell} = u(x_k, y_\ell), \quad u_{k+1,\ell} = u(x_k + h, y_\ell), \quad \text{etc.},$$

the Poisson equation (76) turns into the difference equation

$$\frac{1}{h^2}\left[u_{k-1,\ell} - 2u_{k,\ell} + u_{k+1,\ell}\right] + \frac{1}{h^2}\left[u_{k,\ell-1} - 2u_{k,\ell} + u_{k,\ell+1}\right] = f_{k,\ell}. \tag{79}$$

This equation can be rewritten as

$$4u_{k,\ell} - u_{k-1,\ell} - u_{k+1,\ell} - u_{k,\ell-1} - u_{k,\ell+1} = -h^2 f_{k,\ell}. \tag{80}$$

Example Let's consider a computational mesh of 5×5 points, i.e., $h = \frac{1}{4}$, or $m = 3$. Discretizing the boundary conditions in (76), the values of the approximate solution around the boundary

$$\begin{array}{ll} u_{0,\ell},\ u_{4,\ell} & \ell = 0, \ldots, 4, \\ u_{u,0},\ u_{k,4} & k = 0, \ldots, 4, \end{array}$$

are determined by the appropriate values of ϕ. There remain 9 points in the interior of the domain that have to be determined using the *stencil* (80). Figure 6 illustrates one instance of this task. By applying the stencil to each of the interior points, we obtain 9 conditions for the 9 undetermined values.

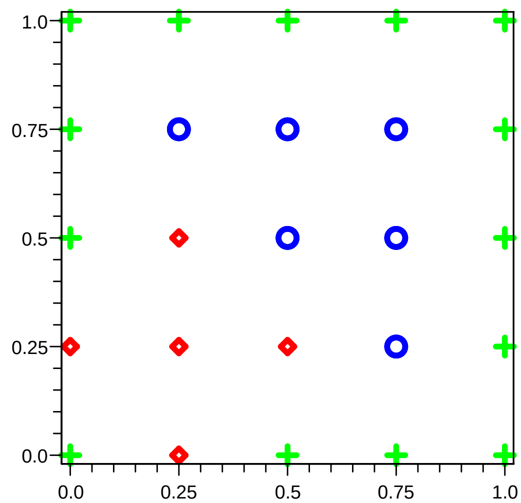

Figure 6: Illustration of finite difference method for Poisson equation on 5×5 grid. Interior mesh points are indicated with blue $\circ$, green $+$ correspond to given boundary values, and points marked with red $\diamondsuit$ form a typical stencil.

Thus, we obtain the following 9 equations

$$\begin{aligned}
4u_{1,1} - u_{2,1} - u_{1,2} &= u_{0,1} + u_{1,0} - h^2 f_{1,1} \\
4u_{2,1} - u_{1,1} - u_{3,1} - u_{2,2} &= u_{2,0} - h^2 f_{2,1} \\
4u_{3,1} - u_{2,1} - u_{3,2} &= u_{4,1} + u_{3,0} - h^2 f_{3,1} \\
4u_{1,2} - u_{2,2} - u_{1,1} - u_{1,3} &= u_{0,2} - h^2 f_{1,2} \\
4u_{2,2} - u_{1,2} - u_{3,2} - u_{2,1} - u_{2,3} &= -h^2 f_{2,2} \\
4u_{3,2} - u_{2,2} - u_{3,1} - u_{3,3} &= u_{4,2} - h^2 f_{3,2} \\
4u_{1,3} - u_{2,3} - u_{1,2} &= u_{1,4} + u_{0,3} - h^2 f_{1,3} \\
4u_{2,3} - u_{1,3} - u_{3,3} - u_{2,2} &= u_{2,4} - h^2 f_{2,3} \\
4u_{3,3} - u_{2,3} - u_{3,2} &= u_{4,3} + u_{3,4} - h^2 f_{3,3}.
\end{aligned}$$

The first equation corresponds to the stencil shown in Figure 6. The other equations are obtained by moving the stencil row-by-row across the grid from left to right.

We can also write the above equations in matrix form. To this end we introduce the vector

$$\boldsymbol{u} = [u_{1,1}, u_{2,1}, u_{3,1}, u_{1,2}, u_{2,2}, u_{3,2}, u_{1,3}, u_{2,3}, u_{3,3}]^T$$

of unknowns. Here we have used the *natural* (row-by-row) ordering of the mesh points. Then we get

$$A\boldsymbol{u} = \boldsymbol{b}$$

with

$$A = \begin{bmatrix} \begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix} & \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} & \begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix} & \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} & \begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix} \end{bmatrix}$$

and

$$\boldsymbol{b} = \begin{bmatrix} u_{0,1} + u_{1,0} - h^2 f_{1,1} \\ u_{2,0} - h^2 f_{2,1} \\ u_{4,1} + u_{3,0} - h^2 f_{3,1} \\ u_{0,2} - h^2 f_{1,2} \\ -h^2 f_{2,2} \\ u_{4,2} - h^2 f_{3,2} \\ u_{1,4} + u_{0,3} - h^2 f_{1,3} \\ u_{2,4} - h^2 f_{2,3} \\ u_{4,3} + u_{3,4} - h^2 f_{3,3} \end{bmatrix}.$$

We can see that A is a block-tridiagonal matrix of the form

$$A = \begin{bmatrix} T & -I & O \\ -I & T & -I \\ O & -I & T \end{bmatrix}.$$

In general, for problems with $m \times m$ interior mesh points, A will be of size $m^2 \times m^2$ (since there are m^2 unknown values at interior mesh points), but contain no more than $5m^2$ nonzero entries (since equation (80) involves at most 5 points at one time). Thus, A is a classical example of a *sparse matrix*. Moreover, A still has a block-tridiagonal structure

$$A = \begin{bmatrix} T & -I & O & \dots & O \\ -I & T & -I & & \vdots \\ O & \ddots & \ddots & \ddots & O \\ \vdots & & -I & T & -I \\ O & \dots & O & -I & T \end{bmatrix}$$

with $m \times m$ blocks

$$T = \begin{bmatrix} 4 & -1 & 0 & \dots & 0 \\ -1 & 4 & -1 & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -1 & 4 & -1 \\ 0 & \dots & 0 & -1 & 4 \end{bmatrix}$$

as well as $m \times m$ identity matrices I, and zero matrices O.

Remark 1. Since A is sparse (and symmetric positive definite) it lends itself to an application of an iterative system solver such as *Gauss-Seidel iteration*. After initializing the values at all mesh points (including those along the boundary) to some appropriate value (in many cases zero will work), we can simply iterate with formula (80), i.e., we obtain the algorithm fragment for M steps of Gauss-Seidel iteration

for $i = 1$ to M do

for $k = 1$ to m do

for $\ell = 1$ to m do

$u_{k,\ell} = \left(u_{k-1,\ell} + u_{k+1,\ell} + u_{k,\ell-1} + u_{k,\ell+1} - h^2 f_{k,\ell}\right)/4$

end

end

end

Note that the matrix A never has to be fully formed or stored during the computation.

2. State-of-the-art algorithms for the Poisson (or homogeneous Laplace) equation are so-called *fast Poisson solvers* based on the fast Fourier transform, or multigrid methods.

While we know that at each gridpoint the Laplacian $u_{xx} + u_{yy}$ is approximated by finite differences with accuracy $\mathcal{O}(h^2)$, one can show that (globally) the error is also of order $\mathcal{O}(h^2)$.

Theorem 8.1 *The maximum pointwise error of the finite difference method with the 5-point stencil introduced above applied to the Poisson problem on a square, rectangular, or L-shaped domain is given by*

$$\max_{k,\ell=1,\dots,m} |u(x_k, y_\ell) - u_{k,\ell}| \le Ch^2, \quad \text{as } h \to 0,$$

where $u(x_k, y_\ell)$ is the exact solution at (x_k, y_ℓ), and $u_{k,\ell}$ is the corresponding approximate solution obtained by the finite difference method.

We emphasize that this estimate holds only for the type of domains specified in the theorem. If the stencil does not match the domain exactly, then we need to use special boundary correction terms to maintain $\mathcal{O}(h^2)$ accuracy (more details are given in the Iserles book on pages 121/122).