Mat1062: Introductory Numerical Methods for PDE

Mary Pugh

January 13, 2009

1 Ownership

These notes are the joint property of Rob Almgren and Mary Pugh.

2 Boundary Conditions

We now want to discuss in detail methods for solving the linear diffusion equation for $u(x,t)$

$$u_t \;=\; D\,u_{xx} \tag{1}$$

We shall suppose that the domain is $X_L \le x \le X_R$, and that initial data $u_0(x)$ is given at $t=0$:

$$u(x,0) \;=\; u_0(x), \qquad X_L \le x \le X_R.$$

and that suitable boundary conditions are given on $x = X_L$ and $x = X_R$ for $t > 0$. We will consider boundary conditions that are *Dirichlet*, *Neumann*, or *Robin*. Dirichlet boundary conditions specify the value of u at the endpoints:

$$u(X_L,t) \;=\; u_L(t), \quad u(X_R,t) \;=\; u_R(t)$$

where u_L and u_R are specified functions of time. Neumann conditions specify the derivative u_x at the endpoints:

$$u_x(X_L,t) \;=\; u_L(t), \quad u_x(X_R,t) \;=\; u_R(t).$$

Robin boundary conditions specify a linear combination of u and u_x at the endpoints:

$$a\,u(X_L,t) + b\,u_x(X_L,t) = u_L(t), \quad c\,u(X_R,t) + d\,u_x(X_R,t) = u_R(t)$$

where a, b, c, and d are fixed constants usually determined by material properties.

2.1 Space Discretization

For given X_L and X_R, we choose a number N. Our grid spacing is

$$h \;=\; \frac{X_R - X_L}{N}$$

and we place grid points at locations

$$x_j \;=\; X_L \;+\; j\,h, \qquad j = 0, 1, \ldots, N,$$

so that $x_0 = X_L$ and $x_N = X_R$. We represent our solution by a vector $U(t)$ of length $N+1$, whose components are

$$U_j(t) \;\approx\; u\big(x_j, t\big), \qquad j = 0, \ldots, N.$$

That is, the jth component of the vector, $U_j(t)$ is supposed to approximate the PDE's solution at the x_j grid point: $u(x_j, t)$.

We take the initial data to be an exact sample of the true initial data $u_0(x)$:

$$U_j(0) \;=\; u_0\big(x_j\big).$$

Given Du_{xx},, a domain $[X_L, X_R]$, and boundary conditions, these determine a linear operator on a function space. We will approximate this linear operator by a matrix operating on the vector U. This allows us to define $U(t)$ as satisfying a linear ODE $U_t = LU$, which we can solve by standard methods. As we will see below, even if the domain is unchanged and the operation Du_{xx} is unchanged, different boundary conditions lead to different linear operators and hence different dynamics. This holds in the continuous case (where the operator is acting on a function space) and in the discretized case (where the approximating operator is just a matrix). Boundary conditions matter!

2.2 Dirichlet Boundary Conditions

Let us consider the case of Dirichlet boundary conditions: $u(X_L, t) = u_L(t)$ and $u(X_R, t) = u_R(t)$. There are $N+1$ meshpoints and $U(t)$ has a component for each meshpoint. However, the the components $U_0(t)$ and $U_N(t)$ are specified by the boundary conditions and so there are only $N-1$ unknowns: $U_1(t) \ldots U_{N-1}(t)$.

From the January 8 notes, we need to approximate $u_{xx}(x_j, t)$ for $1 \leq j \leq N-1$. This is done with the finite-difference approximation. In matrix form this is

$$\begin{pmatrix} u_{xx}(x_1) \\ \vdots \\ u_{xx}(x_{N-1}) \end{pmatrix} \approx \frac{1}{h^2} \begin{pmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} U_0 \\ U_1 \\ \vdots \\ U_{N-1} \\ U_N \end{pmatrix}.$$

However, these expressions are not quite what we want. The vector on the left is of size $N-1$ and will determine the interior grid values. The vector on the right is of size $N+1$; it involves both the interior grid values and the boundary values. We need to express derivatives at the interior grid points in terms only of interior grid values and the Dirichlet boundary conditions. This means that we need to eliminate U_0 and U_N from the above.

Substituting the boundary values $u_0 = u_L(t)$, $u_N = u_R(t)$, we approximate u_{xx} (with Dirichlet boundary conditions on $[X_L, X_R]$) via the *inhomogeneous* linear operator L_1

$$L_1 U = M_1 U + R_1,$$

where M_1 is a $(N-1) \times (N-1)$ tridiagonal matrix and R_1 is an $(N-1)$-vector containing the boundary data:

$$M_1 = \frac{1}{h^2} \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}, \qquad R_1(t) = \frac{1}{h^2} \begin{pmatrix} u_L(t) \\ 0 \\ \vdots \\ 0 \\ u_R(t) \end{pmatrix} \tag{2}$$

R_1 depends on time if the boundary conditions do. In practice, it is often more convenient to store the boundary values in their natural locations at the endpoints of a linear grid, and then the derivatives can be computed by doing the $(N-1)\times(N+1)$ multiplication above.

Of course, the matrix M_1 never actually exists in the computer, nor do we ever actually do a matrix multiply as written above: since most entries are zero, storing the matrix and doing a matrix multiply would be a huge waste of memory and time. "Multiplication" really means "apply the linear operator," which we do by taking differences.

2.3 Neumann Boundary Conditions

For Neumann boundary conditions ($u_x(X_L,t) = u_L(t)$, $u_x(X_R,t) = u_R(t)$) the values at the endponts, $U_0(t)$ and $U_N(t)$, are unknown. Again, we approximate the differentiation operation as

$$\begin{pmatrix} u_{xx}(x_0) \\ \vdots \\ u_{xx}(x_N) \end{pmatrix} \approx \frac{1}{h^2} \begin{pmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} U_{-1} \\ U_0 \\ U_1 \\ \vdots \\ U_{N-1} \\ U_N \\ U_{N+1} \end{pmatrix}.$$

Note that we've done something strange here. We refer to U_{-1} and U_{N+1} which are the values at $X_L - h$ and $X_R + h$ respectively. These grid points are not in our domain. They are artifices introduced to help implement the finite difference approximation of u_{xx} at the endpoints. Such points are often called "ghost points".

The vector on the left is of size $N+1$ and will determine the interior grid values and the boundary grid values. The vector on the right is of size $N+3$; it involves the interior grid values, the boundary grid values, and the values at the two ghost points. We need to express the right-hand side in terms of interior grid values and the boundary grid values only. This means that we need to use the Neumann boundary conditions to determine the values at the ghost points. This is done as follows:

$$u_L(t) = u_x(X_L,t) \sim \frac{U_1 - U_{-1}}{2h} \quad \Longrightarrow \quad U_{-1} = U_1 - 2hu_L(t)$$

as a result

$$u_{xx}(X_L,t) \sim \frac{U_1 - 2U_0 + U_{-1}}{h^2} = \frac{2U_1 - 2U_0}{h^2} - \frac{2u_L(t)}{h}.$$

Similarly, at the right-hand endpoint

$$0 = u_x(X_R, t) \sim \frac{U_{N+1} - U_{N-1}}{2h} \quad \Longrightarrow \quad U_{N+1} = U_{N-1} + 2h u_R(t)$$

hence

$$u_{xx}(X_R, t) \sim \frac{U_{N+1} - 2U_N + U_{N-1}}{h^2} = \frac{2U_{N-1} - 2U_N}{h^2} + \frac{2u_R(t)}{h}.$$

And so, we approximate u_{xx} (with Neumann boundary conditions on $[X_L, X_R]$) with the *inhomogeneous* linear operator

$$M_2 U + R_2,$$

where M_2 is a $(N+1) \times (N+1)$ tridiagonal matrix and R_2 is an $N+1$-vector containing the boundary data:

$$M_2 = \frac{1}{h^2}\begin{pmatrix} -2 & 2 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 2 & -2 \end{pmatrix}, \qquad R_2(t) = \frac{2}{h}\begin{pmatrix} -u_L(t) \\ 0 \\ \vdots \\ 0 \\ u_R(t) \end{pmatrix} \tag{3}$$

A natural question to ask is why we used centered finite differences above to approximate $u_x(X_L)$ and $u_x(X_R)$ rather than using a left-difference for $u_x(X_L)$ and a right-difference for $u_x(X_R)$:

$$0 = u_x(X_L) \sim \frac{U_0 - U_{-1}}{h} \qquad 0 = u_x(X_R) \sim \frac{U_{N+1} - U_N}{h}$$

We will revisit this question when we discuss the convergence and accuracy of our scheme.

2.4 Robin Boundary Conditions

We handle Robin boundary conditions ($a\ u(X_L, t) + b\ u_x(X_L, t) = u_L(t)$, $c\ u(X_R, t) + d\ u_x(X_R, t) = u_R(t)$) in the same manner that we handle Neumann boundary conditions: by introducing ghost grid points the values at which are then determined from the boundary conditions.

$$u_L(t) = a\ u(X_L, t) + b\ u_x(X_L, t) \sim a\ U_0 + b\ \frac{U_1 - U_{-1}}{2h}$$
$$\Longrightarrow \quad U_{-1} = U_1 + \frac{2ah}{b}\ U_0 - \frac{2h u_L(t)}{b}$$

as a result

$$u_{xx}(X_L,t) \sim \frac{U_1 - 2U_0 + U_{-1}}{h^2} = \frac{2U_1 - (2 - 2ah/b)U_0}{h^2} - \frac{2u_L(t)}{bh}.$$

Similarly,

$$u_R(t) = c\, u(X_R,t) + d\, u_x(X_R,t) \sim c\, U_N + d\, \frac{U_{N+1} - U_{N-1}}{2h}$$
$$\implies \quad U_{N+1} = U_{N-1} - \frac{2ch}{d}\, U_N + \frac{2hu_R(t)}{d}$$

hence

$$u_{xx}(X_R,t) \sim \frac{U_1 - 2U_0 + U_{-1}}{h^2} = \frac{2U_{N-1} - (2 + 2ch/d)U_N}{h^2} + \frac{2u_R(t)}{dh}.$$

In this way, we can approximate u_{xx} (with Robin boundary conditions on $[X_L, X_R]$) as the *inhomogeneous* linear operator

$$= M_3 U + R_3$$

where M_2 is a $(N+1) \times (N+1)$ tridiagonal matrix:

$$M_3 = \frac{1}{h^2} \begin{pmatrix} -2 + 2ah/b & 2 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 2 & -2 - 2ch/d \end{pmatrix}, \quad R_3(t) = \frac{2}{h} \begin{pmatrix} -u_L(t)/b \\ 0 \\ \vdots \\ 0 \\ u_R(t)/d \end{pmatrix}. \tag{4}$$

As a reality check, note that if $a = c = 0$ and if $b = d = 1$ then this matrix is the same matrix as for the Neumann boundary conditions.

2.5 Summary

The spatial derivative operator

$$\begin{cases} \mathcal{L}u \;=\; Du_{xx} \qquad \text{on } [X_L, X_R] \\ u(X_L,t) = u_L(t) \\ u(X_R,t) = u_R(t) \end{cases}$$

has the discrete approximation $LU = MU + R$ where $M = DM_1$ and $R = DR_1$ with M_1 and R_1 given in (2). Similarly, the spatial derivative operator

$$\begin{cases} \mathcal{L}u = Du_{xx} \qquad \text{on } [X_L, X_R] \\ u_x(X_L, t) = u_L(t) \\ u_x(X_R, t) = u_R(t) \end{cases}$$

has the discrete approximation $LU = MU + R$ where $M = DM_2$ and $R = DR_2$ with M_2 and R_2 given in (3). Finally, the spatial derivative operator

$$\begin{cases} \mathcal{L}u = Du_{xx} \qquad \text{on } [X_L, X_R] \\ au(X_L, t) + bu_x(X_L, t) = u_L(t) \\ cu(X_R, t) + Du_x(X_R, t) = u_R(t) \end{cases}$$

has the discrete approximation $LU = MU + R$ where $M = DM_3$ and $R = DR_3$ with M_3 and R_3 given in (4).

In all three cases, we have L which is an affine linear operator. But the three matrices M_1, M_2, and M_3 are all different from one another. They will have different eigenvalues and eigenvectors. All of the eigenvalues of M_1 and M_2 will be either zero or negative. The matrix M_3 can have positive eigenvalues. They correspond to modes that grow exponentially in time, rather than decaying — this shows how powerful boundary effects can be!