Forward Difference and Crank Nicolson Methods

10.2 Parabolic Equations

Heat Equation

As an example of parabolic differential equations, we consider the one-dimensional heat equation

(1) $$u_t(x, t) = c^2 u_{xx}(x, t) \quad \text{for } 0 \le x < a \text{ and } 0 < t < b,$$

with the initial condition

(2) $$u(x, 0) = f(x) \quad \text{for } t = 0 \text{ and } 0 \le x \le a,$$

and the boundary conditions

(3) $$\begin{aligned} u(0, t) &= g_1(t) \equiv c_1 \quad \text{for } x = 0 \text{ and } 0 \le t \le b, \\ u(a, t) &= g_2(t) \equiv c_2 \quad \text{for } x = a \text{ and } 0 \le t \le b. \end{aligned}$$

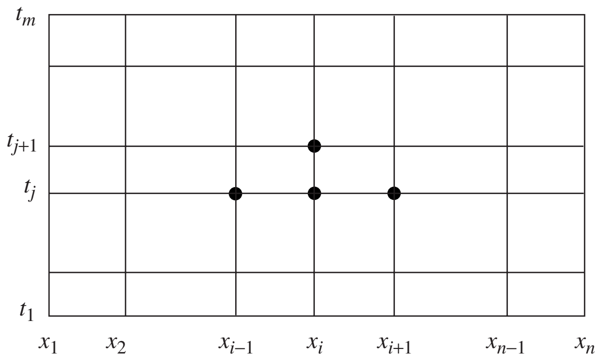

Figure 10.8 The grid for solving $u_t(x,t) = c^2 u_{xx}(x,t)$ over R.

The heat equation models the temperature in an insulated rod with ends held at constant temperatures c_1 and c_2 and the initial temperature distribution along the rod being $f(x)$. Although analytic solutions to the heat equation can be obtained with Fourier series, we use the problem as a prototype of a parabolic equation for numerical solution.

Derivation of the Difference Equation

Assume that the rectangle $R = \{(x,t) : 0 \le x \le a, 0 \le t \le b\}$ is subdivided into $n-1$ by $m-1$ rectangles with sides $\Delta x = h$ and $\Delta t = k$, as shown in Figure 10.8. Start at the bottom row, where $t = t_1 = 0$, and the solution is $u(x_i, t_1) = f(x_i)$. A method for computing the approximations to $u(x,t)$ at grid points in successive rows $\{u(x_i, t_j) : i = 1, 2, \ldots, n\}$, for $j = 2, 3, \ldots, m$, will be developed.

The difference formulas used for $u_t(x,t)$ and $u_{xx}(x,t)$ are

$$u_t(x,t) = \frac{u(x, t+k) - u(x,t)}{k} + \boldsymbol{O}(k) \tag{4}$$

and

$$u_{xx}(x,t) = \frac{u(x-h,t) - 2u(x,t) + u(x+h,t)}{h^2} + \boldsymbol{O}(h^2). \tag{5}$$

The grid spacing is uniform in every row: $x_{i+1} = x_i + h$ (and $x_{i-1} = x_i - h$), and it is uniform in every column: $t_{j+1} = t_j + k$. Next, we drop the terms $\boldsymbol{O}(k)$ and $\boldsymbol{O}(h^2)$ and use the approximation $u_{i,j}$ for $u(x_i, t_j)$ in equations (4) and (5), which are in turn substituted into equation (1) to obtain

$$\frac{u_{i,j+1} - u_{i,j}}{k} = c^2 \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2}, \tag{6}$$

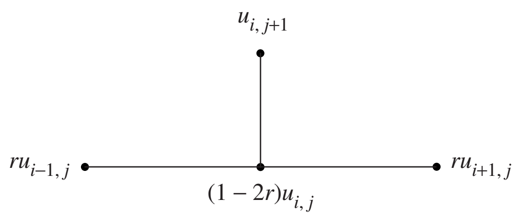

Figure 10.9 The forward-difference stencil.

which approximates the solution to (1). For convenience, the substitution $r = c^2k/h^2$ is introduced in (6), and the result is the explicit forward-difference equation

(7) $$u_{i,j+1} = (1-2r)u_{i,j} + r(u_{i-1,j} + u_{i+1,j}).$$

Equation (7) is employed to create the $(j+1)$th row across the grid, assuming that approximations in the jth row are known. Notice that this formula explicitly gives the value $u_{i,j+1}$ in terms of $u_{i-1,j}$, $u_{i,j}$, and $u_{i+1,j}$. The computational stencil representing the situation in formula (7) is given in Figure 10.9.

The simplicity of formula (7) makes it appealing to use. However, it is important to use numerical techniques that are stable. If any error made at one stage of the calculations is eventually dampened out, the method is called *stable*. The explicit forward-difference equation (7) is stable if and only if r is restricted to the interval $0 \le r \le \frac{1}{2}$. This means that the step size k must satisfy $k \le h^2/(2c^2)$. If this condition is not fulfilled, errors committed in one line $\{u_{i,j}\}$ might be magnified in subsequent lines $\{u_{i,p}\}$ for some $p > j$. The next example illustrates this point.

Example 10.3. Use the forward-difference method to solve the heat equation

(8) $$u_t(x,t) = u_{xx}(x,t) \qquad \text{for } 0 < x < 1 \text{ and } 0 < t < 0.20,$$

with the initial condition

(9) $$u(x,0) = f(x) = 4x - 4x^2 \qquad \text{for } t = 0 \text{ and } 0 \le x \le 1,$$

and the boundary conditions

(10) $$\begin{aligned} u(0,t) &= g_1(t) \equiv 0 \qquad \text{for } x = 0 \text{ and } 0 \le t \le 0.20, \\ u(1,t) &= g_2(t) \equiv 0 \qquad \text{for } x = 1 \text{ and } 0 \le t \le 0.20. \end{aligned}$$

For the first illustration, we use the step sizes $\Delta x = h = 0.2$ and $\Delta t = k = 0.02$ and $c = 1$, so the ratio is $r = 0.5$. The grid will be $n = 6$ columns wide by $m = 11$ rows high. In this case, formula (7) becomes

(11) $$u_{i,j+1} = \frac{u_{i-1,j} + u_{i+1,j}}{2}.$$

Table 10.3 Using the Forward-difference Method with $r = 0.5$

	$x_1 = 0.00$	$x_2 = 0.20$	$x_3 = 0.40$	$x_4 = 0.60$	$x_5 = 0.80$	$x_6 = 1.00$
$t_1 = 0.00$	0.000000	0.640000	0.960000	0.960000	0.640000	0.000000
$t_2 = 0.02$	0.000000	0.480000	0.800000	0.800000	0.480000	0.000000
$t_3 = 0.04$	0.000000	0.400000	0.640000	0.640000	0.400000	0.000000
$t_4 = 0.06$	0.000000	0.320000	0.520000	0.520000	0.320000	0.000000
$t_5 = 0.08$	0.000000	0.260000	0.420000	0.420000	0.260000	0.000000
$t_6 = 0.10$	0.000000	0.210000	0.340000	0.340000	0.210000	0.000000
$t_7 = 0.12$	0.000000	0.170000	0.275000	0.275000	0.170000	0.000000
$t_8 = 0.14$	0.000000	0.137500	0.222500	0.222500	0.137500	0.000000
$t_9 = 0.16$	0.000000	0.111250	0.180000	0.180000	0.111250	0.000000
$t_{10} = 0.18$	0.000000	0.090000	0.145625	0.145625	0.090000	0.000000
$t_{11} = 0.20$	0.000000	0.072812	0.117813	0.117813	0.072812	0.000000

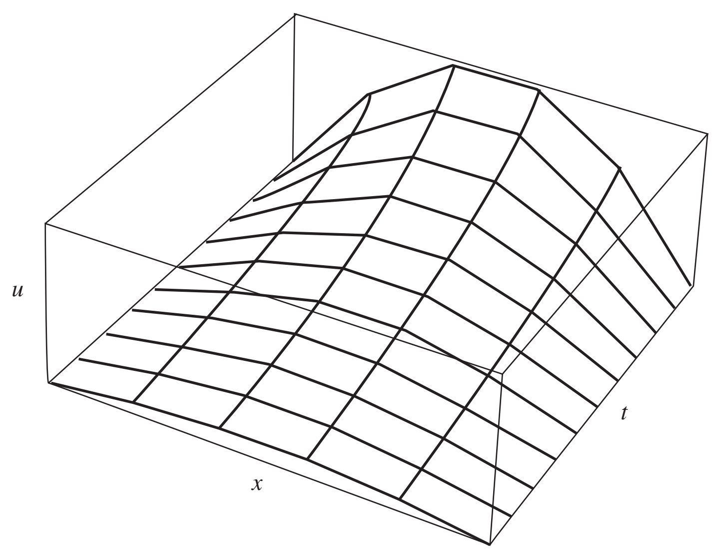

Figure 10.10 Using the forward-difference method with $r = 0.5$.

Formula (11) is stable for $r = 0.5$ and can be used successfully to generate reasonably accurate approximations to $u(x, t)$. Successive rows in the grid are given in Table 10.3. A three-dimensional presentation of the data in Table 10.3 is given in Figure 10.10.

For our second illustration, we use the step sizes $\Delta x = h = 0.2$ and $\Delta t = k = \frac{1}{30} \approx 0.033333$, so that the ratio is $r = 0.833333$. In this case, formula (7) becomes

(12) $$u_{i,j+1} = -0.666665u_{i,j} + 0.833333(u_{i-1,j} + u_{i+1,j}).$$

Table 10.4 Using the Forward-difference Method with $r = 0.833333$

	$x_1 = 0.00$	$x_2 = 0.20$	$x_3 = 0.40$	$x_4 = 0.60$	$x_5 = 0.80$	$x_6 = 1.00$
$t_1 = 0.000000$	0.000000	0.640000	0.960000	0.960000	0.640000	0.000000
$t_2 = 0.033333$	0.000000	0.373333	0.693333	0.693333	0.373333	0.000000
$t_3 = 0.066667$	0.000000	0.328889	0.426667	0.426667	0.328889	0.000000
$t_4 = 0.100000$	0.000000	0.136296	0.345185	0.345185	0.136296	0.000000
$t_5 = 0.133333$	0.000000	0.196790	0.171111	0.171111	0.196790	0.000000
$t_6 = 0.166667$	0.000000	0.011399	0.192510	0.192510	0.011399	0.000000
$t_7 = 0.200000$	0.000000	0.152826	0.041584	0.041584	0.152826	0.000000
$t_8 = 0.233333$	0.000000	−0.067230	0.134286	0.134286	−0.067230	0.000000
$t_9 = 0.266667$	0.000000	0.156725	−0.033644	−0.033644	0.156725	0.000000
$t_{10} = 0.300000$	0.000000	−0.132520	0.124997	0.124997	−0.132520	0.000000
$t_{11} = 0.333333$	0.000000	0.192511	−0.089601	−0.089601	0.192511	0.000000

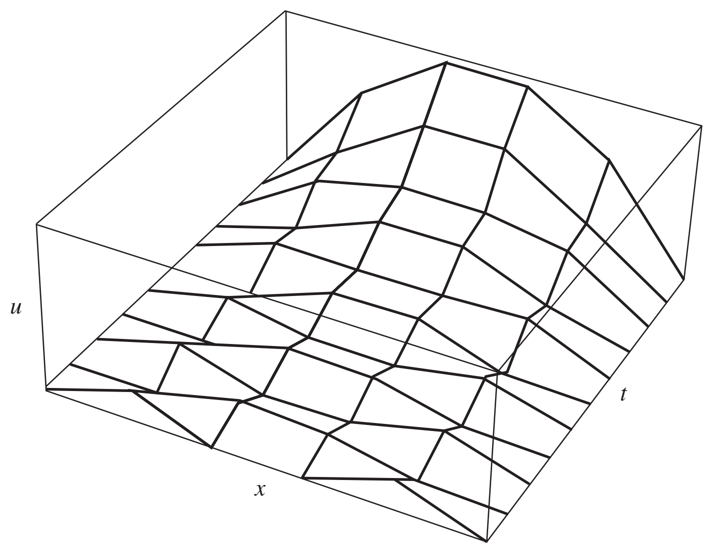

Figure 10.11 Using the forward-difference method with $r = 0.833333$.

Formula (12) is unstable in this case, because $r > \frac{1}{2}$, and errors committed at one row will be magnified in successive rows. Numerical values that turn out to be imprecise approximations to $u(x, t)$, for $0 \le t \le 0.33333$, are given in Table 10.4. A three-dimensional presentation of the data in Table 10.4 is given in Figure 10.11. ■

The difference equation (7) has accuracy of the order $\boldsymbol{O}(k) + \boldsymbol{O}(h^2)$. Because the term $\boldsymbol{O}(k)$ decreases linearly as k tends to zero, it is not surprising that it must be made small to produce good approximations. However, the stability requirement introduces further considerations. Suppose that the solutions over the grid are not sufficiently accurate and that both the increments $\Delta x = h_0$ and $\Delta t = k_0$ must be reduced. For simplicity, suppose that the new x increment is $\Delta x = h_1 = h_0/2$. If the same ratio r is used, k_1 must satisfy

$$k_1 = \frac{r(h_1)^2}{c^2} = \frac{r(h_0)^2}{4c^2} = \frac{k_0}{4}.$$

This results in a doubling and quadrupling of the number of grid points along the x-axis and t-axis, respectively. Consequently, there must be an eightfold increase in the total computational effort when reducing the grid size in this manner. This extra effort is usually prohibitive and demands that we explore a more efficient method that does not have stability restrictions. The method proposed will be implicit rather than explicit. The apparent rise in the level of complexity will have the immediate payoff of being unconditionally stable.

Crank-Nicolson Method

An implicit scheme, invented by John Crank and Phyllis Nicolson, is based on numerical approximations for solutions of equation (1) at the point $(x, t + k/2)$ that lies between the rows in the grid. Specifically, the approximation used for $u_t(x, t + k/2)$ is obtained from the central-difference formula,

$$u_t\left(x, t + \frac{k}{2}\right) = \frac{u(x, t + k) - u(x, t)}{k} + \boldsymbol{O}(k^2). \tag{13}$$

The approximation used for $u_{xx}(x, t + k/2)$ is the average of the approximations $u_{xx}(x, t)$ and $u_{xx}(x, t + k)$, which has an accuracy of the order $\boldsymbol{O}(h^2)$:

$$\begin{aligned} u_{xx}\left(x, t + \frac{k}{2}\right) = \frac{1}{2h^2}(&u(x - h, t + k) - 2u(x, t + k) + u(x + h, t + k) \\ &+ u(x - h, t) - 2u(x, t) + u(x + h, t)) + \boldsymbol{O}(h^2). \end{aligned} \tag{14}$$

In a fashion similar to the previous derivation, we substitute (13) and (14) into (1) and neglect the error terms $\boldsymbol{O}(h^2)$ and $\boldsymbol{O}(k^2)$. Then employing the notation $u_{i,j} = u(x_i, t_j)$ will produce the difference equation

$$\frac{u_{i,j+1} - u_{i,j}}{k} = c^2 \frac{u_{i-1,j+1} - 2u_{i,j+1} + u_{i+1,j+1} + u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{2h^2}. \tag{15}$$

Also, the substitution $r = c^2k/h^2$ is used in (15). But this time we must solve for the three "yet to be computed" values $u_{i-1,j+1}$, $u_{i,j+1}$, and $u_{i+1,j+1}$. This is accomplished by placing them all on the left side of the equation. Then rearrangement of the terms in equation (15) results in the implicit difference formula

$$\begin{aligned} -ru_{i-1,j+1} + (2 + 2r)u_{i,j+1} - ru_{i+1,j+1} \\ = (2 - 2r)u_{i,j} + r(u_{i-1,j} + u_{i+1,j}). \end{aligned} \tag{16}$$

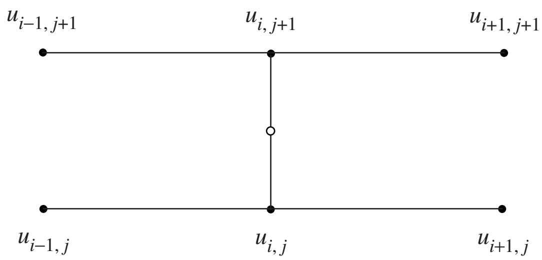

Figure 10.12 The Crank-Nicolson stencil.

for $i = 2, 3, \ldots, n-1$. The terms on the right-hand side of equation (16) are all known. Hence the equations in (16) form a tridiagonal linear system $AX = B$. The six points used in the Crank-Nicolson formula (16), together with the intermediate grid point where the numerical approximations are based, are shown in Figure 10.12.

Implementation of formula (16) is sometimes done by using the ratio $r = 1$. In this case the increment along the t-axis is $\Delta t = k = h^2/c^2$, and the equations in (16) simplify and become

$$-u_{i-1,j+1} + 4u_{i,j+1} - u_{i+1,j+1} = u_{i-1,j} + u_{i+1,j}, \tag{17}$$

for $i = 2, 3, \ldots, n-1$. The boundary conditions are used in the first and last equations (i.e., $u_{1,j} = u_{1,j+1} = c_1$ and $u_{n,j} = u_{n,j+1} = c_2$, respectively). Equations (17) are especially pleasing to view in their tridiagonal matrix form $\boldsymbol{AX} = \boldsymbol{B}$.

$$\begin{bmatrix} 4 & -1 & & & & & \\ -1 & 4 & -1 & & \boldsymbol{O} & & \\ & & \ddots & & & & \\ & & -1 & 4 & -1 & & \\ & & & & \ddots & & \\ & \boldsymbol{O} & & & -1 & 4 & -1 \\ & & & & & -1 & 4 \end{bmatrix} \begin{bmatrix} u_{2,j+1} \\ u_{3,j+1} \\ \vdots \\ u_{p,j+1} \\ \vdots \\ u_{n-2,j+1} \\ u_{n-1,j+1} \end{bmatrix} = \begin{bmatrix} 2c_1 + u_{3,j} \\ u_{2,j} + u_{4,j} \\ \vdots \\ u_{p-1,j} + u_{p+1,j} \\ \vdots \\ u_{n-3,j} + u_{n-1,j} \\ u_{n-2,j} + 2c_2 \end{bmatrix}.$$

When the Crank-Nicolson method is implemented with a computer, the linear system $\boldsymbol{AX} = \boldsymbol{B}$ can be solved by either direct means or by iteration.

Example 10.4. Use the Crank-Nicolson method to solve the equation

$$u_t(x,t) = u_{xx}(x,t) \quad \text{for } 0 < x < 1 \text{ and } 0 < t < 0.1, \tag{18}$$

with the initial condition

$$u(x,0) = f(x) = \sin(\pi x) + \sin(3\pi x) \quad \text{for } t = 0 \text{ and } 0 \le x \le 1, \tag{19}$$

Table 10.5 Values $u(x_i, t_i)$ Using the Crank-Nicolson Method with $t_j = (j-1)/100$

	$x_2 = 0.1$	$x_3 = 0.2$	$x_4 = 0.3$	$x_5 = 0.4$	$x_6 = 0.5$	$x_7 = 0.6$	$x_8 = 0.7$	$x_9 = 0.8$	$x_{10} = 0.9$
t_1	1.118034	1.538842	1.118034	0.363271	0.000000	0.363271	1.118034	1.538842	1.118034
t_2	0.616905	0.928778	0.862137	0.617659	0.490465	0.617659	0.862137	0.928778	0.616905
t_3	0.394184	0.647957	0.718601	0.680009	0.648834	0.680009	0.718601	0.647957	0.394184
t_4	0.288660	0.506682	0.625285	0.666493	0.673251	0.666493	0.625285	0.506682	0.288660
t_5	0.233112	0.425766	0.556006	0.625082	0.645788	0.625082	0.556006	0.425766	0.233112
t_6	0.199450	0.372035	0.499571	0.575402	0.600242	0.575402	0.499571	0.372035	0.199450
t_7	0.175881	0.331490	0.451058	0.525306	0.550354	0.525306	0.451058	0.331490	0.175881
t_8	0.157405	0.298131	0.408178	0.477784	0.501545	0.477784	0.408178	0.298131	0.157405
t_9	0.141858	0.269300	0.369759	0.433821	0.455802	0.433821	0.369759	0.269300	0.141858
t_{10}	0.128262	0.243749	0.335117	0.393597	0.413709	0.393597	0.335117	0.243749	0.128262
t_{11}	0.116144	0.220827	0.303787	0.356974	0.375286	0.356974	0.303787	0.220827	0.116144

and the boundary conditions

$$u(0,t) = g_1(t) \equiv 0 \quad \text{for } x = 0 \text{ and } 0 \le t \le 0.1,$$
$$u(1,t) = g_2(t) \equiv 0 \quad \text{for } x = 1 \text{ and } 0 \le t \le 0.1.$$

For simplicity, we use the step sizes $\Delta x = h = 0.1$ and $\Delta t = k = 0.01$ so that the ratio is $r = 1$. The grid will be $n = 11$ columns wide by $m = 11$ rows high. Applying the algorithm generates the values in Table 10.5 for $0 < x_i < 1$ and $0 \le t_j \le 0.1$.

The values obtained with the Crank-Nicolson method compare favorably with the analytic solution $u(x,t) = \sin(\pi x)e^{-\pi^2 t} + \sin(3\pi x)e^{-9\pi^2 t}$, the true values for the final row being

t_{11}	0.115285	0.219204	0.301570	0.354385	0.372569	0.354385	0.301570	0.219204	0.115285

A three-dimensional presentation of the data in Table 10.5 is given in Figure 10.13. ■

Program 10.2 (Forward-Difference Method for the Heat Equation). To approximate the solution of $u_t(x,t) = c^2 u_{xx}(x,t)$ over $R = \{(x,t) : 0 \le x \le a, 0 \le t \le b\}$ with $u(x,0) = f(x)$, for $0 \le x \le a$, and $u(0,t) = c_1$, $u(a,t) = c_2$, for $0 \le t \le b$.

```
function U=forwdif(f,c1,c2,a,b,c,n,m)
%Input  - f=u(x,0) as a string 'f'
%       - c1=u(0,t) and c2=u(a,t)
%       - a and b right endpoints of [0,a] and [0,b]
%       - c the constant in the heat equation
%       - n and m number of grid points over [0,a] and [0,b]
%Output - U solution matrix; analogous to Table 10.4
```

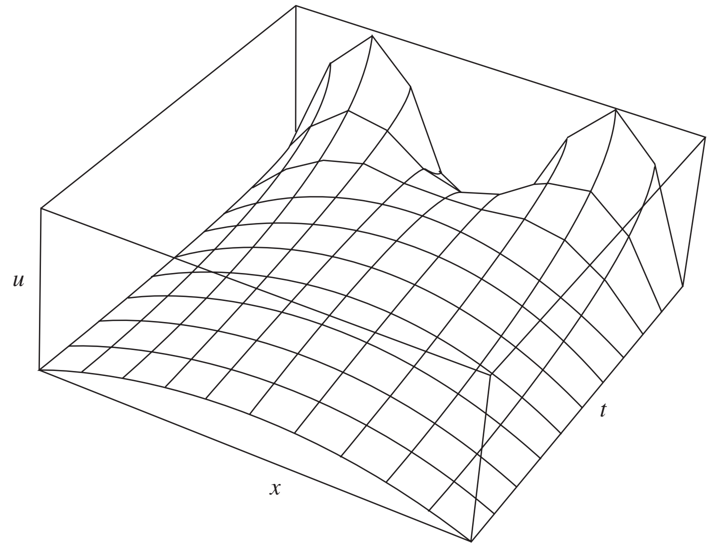

Figure 10.13 $u = u(x_i, t_j)$ from the Crank-Nicolson method.

Numerical Methods Using Matlab, 4th Edition, 2004

John H. Mathews and Kurtis K. Fink

ISBN: 0-13-065248-2

Prentice-Hall Inc.
Upper Saddle River, New Jersey, USA
http://vig.prenhall.com/