Objective. We will begin our discussion on the numerical solution of parabolic partial differential equations.

Agenda. In today's class, we will discuss the following.

1. Announcements

2. We discuss the material in Chapter 9, Section 9.1 of LeVeque's book.

Topic: Diffusion Equations and Parabolic Problems. We now apply what we have learned about boundary value problems and initial value problems to solve the canonical parabolic partial differential equation

$$u_t = \kappa u_{xx}, \quad \text{in } (0,1) \times (0,T), \quad \kappa > 0.$$

For this initial-boundary value problem, we must prescribe an initial condition of the form

$$u(x,0) = \eta(x)$$

and Dirichlet boundary conditions (for example) of the form

$$\begin{aligned} u(0,t) &= g_0(t) \quad \text{for } t > 0 \\ u(1,t) &= g_1(t) \quad \text{for } t > 0. \end{aligned}$$

To solve this problem, we introduce a spatial grid of the form $x_i = ih$ with mesh width h and a temporal grid of the form $t_n = nk$ with time step k. Let

$$U_i^n \approx u(x_i, t_n)$$

be the numerical approximation of the solution at (x_i, t_n).

The main idea in the numerical solution is to discretize the spatial derivatives to obtain a system of differential equations that we march in time. Suppose we use a second-order centered finite difference approximation for the spatial derivative and a forward Euler's method for the time marching to obtain

$$\frac{U_i^{n+1} - U_i^n}{k} = \frac{\kappa}{h^2}(U_{i-1}^n - 2U_i^n + U_{i+1}^n).$$

Notice that this is an explicit method since we can apply an update formula of the form

$$U_i^{n+1} = U_i^n + \frac{k\kappa}{h^2}(U_{i-1}^n - 2U_i^n + U_{i+1}^n).$$

We will show that this method is not useful in practice. Instead, we study the so-called Crank-

Nicolson method

$$\frac{U_i^{n+1} - U_i^n}{k} = \frac{\kappa}{2h^2}(U_{i-1}^{n+1} - 2U_i^{n+1} + U_{i+1}^{n+1}) + \frac{\kappa}{2h^2}(U_{i-1}^n - 2U_i^n + U_{i+1}^n).$$

Notice that this method is implicit. It involves an average of the spatial derivative at time level n and $n+1$. If we rearrange terms in the equation above, we obtain

$$U_i^{n+1} - \frac{k\kappa}{2h^2}(U_{i-1}^{n+1} - 2U_i^{n+1} + U_{i+1}^{n+1}) = U_i^n + \frac{k\kappa}{2h^2}(U_{i-1}^n - 2U_i^n + U_{i+1}^n).$$

Let A be the following tridiagonal matrix

$$A = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix}.$$

Then, for the vector $\mathbf{U}^n$ whose entries are $\mathbf{U}^n = (U_1^n, U_2^n, \cdots, U_m^n)$, the Crank-Nicolson method is given by

$$(I - \frac{1}{2}k\kappa A)\mathbf{U}^{n+1} = (I + \frac{1}{2}k\kappa A)\mathbf{U}^n.$$

So, for each time step, one needs only to solve the tridiagonal system of equations. Hence, each time step requires only $O(m)$ operations.

Remember that we need to incorporate the boundary conditions! For Dirichlet boundary conditions, we first evaluate the finite difference scheme at $i = 1$ and obtain

$$U_1^{n+1} - \frac{k\kappa}{2h^2}(U_0^{n+1} - 2U_1^{n+1} + U_2^{n+1}) = U_1^n + \frac{k\kappa}{2h^2}(U_0^n - 2U_1^n + U_2^n).$$

Substituting $U_0^n = g_0(t_n)$, we obtain

$$U_1^{n+1} - \frac{k\kappa}{2h^2}(g_0(t_{n+1}) - 2U_1^{n+1} + U_2^{n+1}) = U_1^n + \frac{k\kappa}{2h^2}(g_0(t_n) - 2U_1^n + U_2^n).$$

Rearranging terms, we obtain

$$U_1^{n+1} - \frac{k\kappa}{2h^2}(-2U_1^{n+1} + U_2^{n+1}) = U_1^n + \frac{k\kappa}{2h^2}(-2U_1^n + U_2^n) + \frac{k\kappa}{2h^2}[g_0(t_{n+1}) + g_0(t_n)].$$

Next, we evaluate the finite difference formula at $i = m$ and obtain

$$U_m^{n+1} - \frac{k\kappa}{2h^2}(U_{m-1}^{n+1} - 2U_m^{n+1} + U_{m+1}^{n+1}) = U_m^n + \frac{k\kappa}{2h^2}(U_{m-1}^n - 2U_m^n + U_{m+1}^n).$$

Substituting $U_{m+1}^n = g_1(t_n)$, we obtain

$$U_m^{n+1} - \frac{k\kappa}{2h^2}(U_{m-1}^{n+1} - 2U_m^{n+1} + g_1(t_{n+1})) = U_m^n + \frac{k\kappa}{2h^2}(U_{m-1}^n - 2U_m^n + g_1(t_n)).$$

Rearranging terms, we obtain

$$U_m^{n+1} - \frac{k\kappa}{2h^2}(U_{m-1}^{n+1} - 2U_m^{n+1}) = U_m^n + \frac{k\kappa}{2h^2}(U_{m-1}^n - 2U_m^n) + \frac{k\kappa}{2h^2}[g_1(t_{n+1}) + g_1(t_n)].$$

So, in fact, we solve at each time step

$$(I - \frac{1}{2}k\kappa A)\mathbf{U}^{n+1} = (I + \frac{1}{2}k\kappa A)\mathbf{U}^n + \frac{k\kappa}{2h^2}\begin{bmatrix} g_0(t_{n+1}) + g_0(t_n) \\ 0 \\ \vdots \\ 0 \\ g_1(t_{n+1}) + g_1(t_n) \end{bmatrix},$$

starting with

$$U_i^0 = \eta(x_i).$$

Thus, to solve this Dirichlet initial-boundary value problem, we follow the procedure below.

1. Compute the spatial grid $x_i = ih$ with mesh width h for $i = 1, \cdots, m$.
2. Set the time step k.
3. Compute the matrices $I - \frac{1}{2}k\kappa A$ and $I + \frac{1}{2}k\kappa A$ (use sparse matrix storage!).
4. Set the initial condition $\mathbf{U}^0 = \eta$.
5. For each time step, solve the tridiagonal system

$$(I - \frac{1}{2}k\kappa A)\mathbf{U}^{n+1} = (I + \frac{1}{2}k\kappa A)\mathbf{U}^n + \frac{k\kappa}{2h^2}\begin{bmatrix} g_0(t_{n+1}) + g_0(t_n) \\ 0 \\ \vdots \\ 0 \\ g_1(t_{n+1}) + g_1(t_n) \end{bmatrix}.$$

An example Matlab code is given by `Heat1D.m` with these notes.

Local Truncation Error. Let us now examine the local truncation error for the two methods described above. To compute the local truncation error, we insert the exact solution $u(x,t)$ into the finite difference equation and determine the leading order error in satisfying the finite difference equation.

For the forward Euler method:

$$U_i^{n+1} = U_i^n + \frac{k\kappa}{h^2}(U_{i-1}^n - 2U_i^n + U_{i+1}^n)$$

we substitute $u(x,t)$ and obtain

$$\tau(x,t) = \frac{u(x,t+k) - u(x,t)}{k} - \frac{\kappa}{h^2}[u(x-h,t) - 2u(x,t) + u(x+h,t)].$$

By substituting the following Taylor series expansions

$$\begin{aligned} u(x,t+k) &= u(x,t) + ku_t(x,t) + \frac{k^2}{2}u_{tt}(x,t) + \frac{k^3}{6}u_{ttt}(x,t) + \cdots \\ u(x \pm h,t) &= u(x,t) \pm hu_x(x,t) + \frac{h^2}{2}u_{xx}(x,t) \pm \frac{h^3}{6}u_{xxx}(x,t) + \frac{h^4}{24}u_{xxxx}(x,t) + \cdots, \end{aligned}$$

we obtain

$$\tau(x,t) = \left[u_t(x,t) + \frac{k}{2}u_{tt}(x,t) + \cdots\right] - \kappa\left[u_{xx}(x,t) + \frac{1}{12}h^2 u_{xxxx} + \cdots\right].$$

We know that $u_t = \kappa u_{xx}$ from the original PDE. Moreover, we differentiate this PDE to find that $u_{tt} = \kappa(u_{xx})_t = \kappa^2 u_{xxxx}$. Thus, we find to leading order that

$$\tau(x,t) = \left[\frac{\kappa^2}{2}k - \frac{1}{12}h^2\right]u_{xxxx} + O(k^2 + h^4).$$

As a result, we say that this method is second order in space and first order in time.

For the Crank-Nicolson method, the local truncation error is defined as

$$\begin{aligned} \tau(x,t) = \frac{u(x,t+k) - u(x,t)}{k} &- \frac{\kappa}{2h^2}[u(x-h,t+k) - 2u(x,t+k) + u(x+h,t+k)] \\ &- \frac{\kappa}{2h^2}[u(x-h,t) - 2u(x,t) + u(x+h,t)]. \end{aligned}$$

By substituting Taylor series expansions into this expression, we find that

$$\tau(x,t) = \left[u_t(x,t) + \frac{k}{2}u_{tt}(x,t) + \frac{k^2}{6}u_{ttt}(x,t) + \frac{k^3}{24}u_{ttt}(x,t) + \cdots\right]$$
$$- \frac{\kappa}{2}\left[u_{xx}(x,t) + k u_{xxt}(x,t) + \frac{h^2}{12}u_{xxxx}(x,t) + \frac{1}{2}k^2 u_{xxtt}(x,t) + \frac{1}{12}kh^2 u_{xxxxt}(x,t) + \cdots\right]$$
$$- \frac{\kappa}{2}\left[u_{xx}(x,t) + \frac{1}{12}h^2 u_{xxxx} + \cdots\right]$$

Collecting like powers of k and h, we obtain

$$\tau(x,t) = [u_t - \kappa u_{xx}] + k\frac{1}{2}[u_{tt} - \kappa u_{xxt}] + k^2\left[\frac{1}{6}u_{ttt} - \frac{\kappa}{4}u_{xxtt}\right] - k\kappa h^2\frac{1}{24}u_{xxxxt} - h^2\frac{\kappa}{12}u_{xxxx} + k^3\frac{1}{24}u_{tttt} + \cdots$$

which simplifies to

$$\tau(x,t) = -k^2\kappa\frac{1}{12}u_{tttt} - h^2\frac{\kappa}{12}u_{xxxx} - kh^2\frac{\kappa}{24}u_{xxxxt} + k^3\frac{1}{24}u_{tttt} + \cdots$$

and so we say the Crank-Nicolson method is second order in space and second order in time.

We say that a method is consistent if $\tau(x,t) \to 0$ as $k,\ h \to 0$.