This section is a copy from GitHub/numerical-moc notebooks.

0.1. The Crank-Nicolson Method

The Crank-Nicolson method is a well-known finite difference method for the numerical integration of the heat equation and closely related partial differential equations. We often resort to a Crank-Nicolson (CN) scheme when we integrate numerically reaction-diffusion systems in one space dimension

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + f(u),$$
$$\left.\frac{\partial u}{\partial x}\right|_{x=0,L} = 0,$$

where u is our concentration variable, x is the space variable, D is the diffusion coefficient of u, f is the reaction term, and L is the length of our one-dimensional space domain.

Note that we use Neumann boundary conditions and specify that the solution u has zero space slope at the boundaries, effectively prohibiting entrance or exit of material at the boundaries (no-flux boundary conditions).

0.2. Finite difference methods

Many fantastic textbooks and tutorials have been written about finite difference methods, for instance a free textbook by Lloyd Trefethen. Here we describe a few basic aspects of finite difference methods. The above reaction-diffusion equation describes the time evolution of variable $u(x,t)$ in one space dimension (u is a line concentration). If we knew an analytic expression for $u(x,t)$ then we could plot u in a two-dimensional coordinate system with axes t and x.

To approximate $u(x,t)$ numerically we discretize this two-dimensional coordinate system resulting, in the simplest case, in a two-dimensional regular grid. This picture is employed commonly when constructing finite differences methods, see for instance Figure 3.2.1 of Trefethen.

Let us discretize both time and space as follows:

$$t_n = n\Delta t,\ n = 0, \ldots, N-1, \tag{0.1}$$

$$x_j = j\Delta x,\ j = 0, \ldots, J-1, \tag{0.2}$$

where N and J are the number of discrete time and space points in our grid respectively. Δt and Δx are the time step and space step respectively and defined as follows:

$$\Delta t = T/N, \tag{0.3}$$

$$\Delta x = L/J \tag{0.4}$$

where T is the point in time up to which we will integrate u numerically.

Our ultimate goal is to construct a numerical method that allows us to approximate the unknonwn analytic solution $u(x,t)$ reasonably well in these discrete grid points. That is we want construct a method that computes values $U(j\Delta x, n\Delta t)$ (note: capital U) so that

$$U(j\Delta x, n\Delta t) \approx u(j\Delta x, n\Delta t) \tag{0.5}$$

As a shorthand we will write $U_j^n = U(j\Delta x, n\Delta t)$ and (j,n) to refer to grid point $(j\Delta x, n\Delta t)$. Let us define $\sigma = \frac{D\Delta t}{2\Delta x^2}$ and reorder the above approximation of our reaction-diffusion equation:

$$\boxed{-\sigma U_{j-1}^{n+1} + (1+2\sigma)U_j^{n+1} - \sigma U_{j+1}^{n+1} = \sigma U_{j-1}^n + (1-2\sigma)U_j^n + \sigma U_{j+1}^n + \Delta t f(U_j^n).} \tag{0.6}$$

This equation makes sense for space indices $j = 1, \ldots, J-2$ but it does not make sense for indices $j = 0$ and $j = J-1$ (on the boundaries):

$$j = 0: \quad -\sigma U_{-1}^{n+1} + (1+2\sigma)U_0^{n+1} - \sigma U_1^{n+1} = \sigma U_{-1}^n + (1-2\sigma)U_0^n + \sigma U_1^n + \Delta t f(U_0^n), \tag{0.7}$$

$$j = J-1: \quad -\sigma U_{J-2}^{n+1} + (1+2\sigma)U_{J-1}^{n+1} - \sigma U_J^{n+1} = \sigma U_{J-2}^n + (1-2\sigma)U_{J-1}^n + \sigma U_J^n + \Delta t f(U_{J-1}^n). \tag{0.8}$$

The problem here is that the values U_{-1}^n and U_J^n lie outside our grid. However, we can work out what these values should equal by considering our Neumann boundary condition. Let us discretize our boundary condition at $j = 0$ with the backward difference and at $j = J-1$ with the forward difference:

$$\frac{U_1^n - U_0^n}{\Delta x} = 0, \tag{0.9}$$

$$\frac{U_J^n - U_{J-1}^n}{\Delta x} = 0. \tag{0.10}$$

These two equations make it clear that we need to amend our above numerical approximation for $j = 0$ with the identities $U_0^n = U_1^n$ and $U_0^{n+1} = U_1^{n+1}$, and for $j = J-1$ with the identities $U_{J-1}^n = U_J^n$ and $U_{J-1}^{n+1} = U_J^{n+1}$. Let us reinterpret our numerical approximation of the line concentration of u in a fixed point in time as a vector $\mathbf{U}^n$:

$$\mathbf{U}^n = \begin{bmatrix} U_0^n \\ \vdots \\ U_{J-1}^n \end{bmatrix}. \tag{0.11}$$

Using this notation we can now write our above approximation for a fixed point in time, $t = n\Delta t$, compactly as a linear system:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ -\sigma & 1+2\sigma & -\sigma & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & -\sigma & 1+2\sigma & -\sigma & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\sigma & 1+2\sigma & -\sigma \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U_0^{n+1} \\ U_1^{n+1} \\ U_2^{n+1} \\ \vdots \\ U_{J-2}^{n+1} \\ U_{J-1}^{n+1} \end{bmatrix} =$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \sigma & 1-2\sigma & \sigma & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & \sigma & 1-2\sigma & \sigma & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma & 1-2\sigma & \sigma \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} U_0^n \\ U_1^n \\ U_2^n \\ \vdots \\ U_{J-2}^n \\ U_{J-1}^n \end{bmatrix} + \begin{bmatrix} f(U_1^n) \\ 0 \\ 0 \\ \vdots \\ 0 \\ g(U_{J-2}^n) \end{bmatrix}.$$

Note that since our numerical integration starts with a well-defined initial condition at $n = 0$, $\mathbf{U}^0$, the vector $\mathbf{U}^{n+1}$ on the left-hand side is the only unknown in this system of linear equations. Thus, to integrate numerically our reaction-diffusion system from time point n to $n + 1$ we need to solve numerically for vector $\mathbf{U}^{n+1}$. Let us call the matrix on the left-hand side A, the one on the right-hand side B, and the vector on the right-hand side $\mathbf{f}^n$. Using this notation we can write the above system as

$$A\mathbf{U}^{n+1} = B\mathbf{U}^n + f^n. \tag{0.12}$$

In this linear equation, matrices A and B are defined by our problem: we need to specify these matrices once for our problem and incorporate our boundary conditions in them. Vector $\mathbf{f}^n$ is a function of $\mathbf{U}^n$ and so needs to be reevaluated in every time point n. We also need to carry out one matrix-vector multiplication every time point, $B\mathbf{U}^n$, and one vector-vector addition, $B\mathbf{U}^n + f^n$. The most expensive numerical operation is inversion of matrix A to solve for $\mathbf{U}^{n+1}$, however we may get away with doing this only once and store the inverse of A as A^{-1}:

$$\mathbf{U}^{n+1} = A^{-1}\left(B\mathbf{U}^n + f^n\right). \tag{0.13}$$