The Crank–Nicolson method

November 5, 2015

It is my impression that many students found the Crank–Nicolson method hard to understand. These notes are intended to complement Kreyszig's quite messy discussion of the method.

The notes take the same approach as I did in the lecture, i.e. obtaining Crank–Nicolson by transforming the PDE into a system of ODEs and applying a Runge–Kutta method to solve that system.

1 Semidiscretization of the heat equation

The Crank–Nicolson method applies to the heat equation

$$\frac{\partial u}{\partial t}(x,t) = \frac{\partial^2 u}{\partial x^2} u(x,t) \tag{1}$$

for $(x,t) \in [0,1] \times [0,\infty)$, and similar parabolic partial differential equations.

Divide $[0,1]$ into N intervals with the nodes $x_0, x_1, \dots, x_N$ spaced h apart, i.e. $h = 1/N$. By applying the *central difference approximation*[1],

$$\frac{\partial^2 u}{\partial x^2} u(x,t) \approx \frac{1}{h^2}(u(x+h,t) - 2u(x,t) + u(x-h,t)),$$

we turn equation (1) into a system of equations

$$\frac{\partial u}{\partial t}(ih,t) \approx \frac{1}{h^2}(u(ih+h,t) - 2u(ih,t) + u(ih-h,t)) \qquad \text{for } 0 < i < N.$$

If we write $v_i(t)$ for our approximation of $u(ih,t)$, that system can be written

$$v'_i(t) = \frac{1}{h^2}(v_{i+1}(t) - 2v_i(t) + v_{i-1}(t)) \qquad \text{for } 0 < i < N.$$

In this notation, $v_0(t) = u(0,t) = L(t)$ and $v_N(t) = u(Nh,t) = u(1,t) = R(t)$, i.e. the left and the right boundary conditions, respectively.

With $\mathbf{v}(t) = (v_1(t), v_2(t), \dots v_{N-1}(t))$ we can thus write the system as

$$\mathbf{v}'(t) = \frac{1}{h^2} \begin{pmatrix} v_2(t) - 2v_1(t) + L(t) \\ v_3(t) - 2v_2(t) + v_1(t) \\ \vdots \\ v_{N-1}(t) - 2v_{N-2}(t) + v_{N-3}(t) \\ R(t) - 2v_{N-1}(t) + v_{N-2}(t) \end{pmatrix} = \mathbf{f}(t, \mathbf{v}(t)). \tag{2}$$

Equation (2) is called the *semidiscretized* form of equation (1), and we now aim to solve it numerically. Figure 1 summarizes situation.

[1] As in the lectures: add the Taylor series for $u(x+h,t)$ and for $u(x-h,t)$ so that the odd powers of h drop out, and truncate above order h^2.

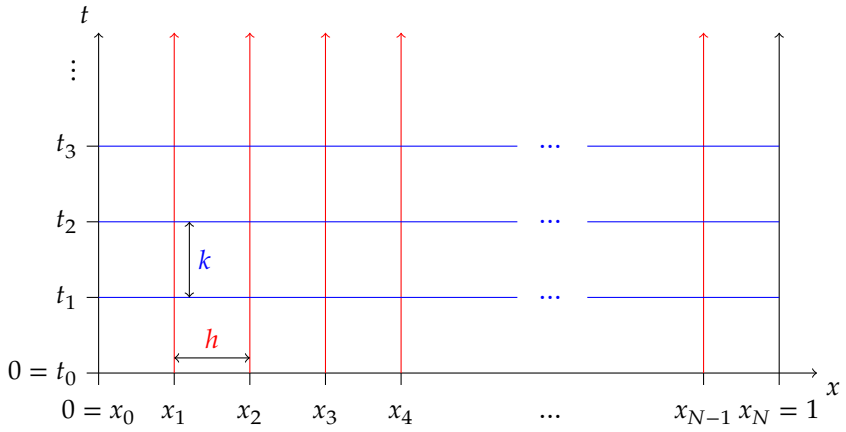
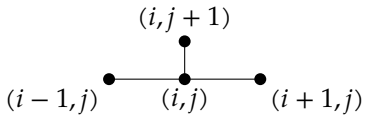


Figure 1: Central differences for the spatial second derivatives turns the heat equation (1) into a system of ODEs (2) along the red lines. This system is solved by stepping forward in time with a Runge–Kutta method, giving us approximations of the solution $u(ih, jk)$ at the nodes (where red and blue lines intersect). Note that in general $h \neq k$, even though the drawing indicates otherwise.

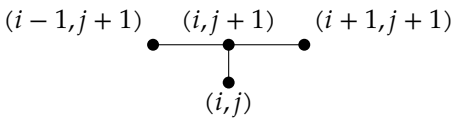
2 Solving equation (2)

Equation (2) can be solved using any Runge–Kutta method for ordinary differential equations. Forward Euler gives an explicit scheme with stencil[2]



,

while backwards Euler gives an implicit scheme with stencil



.

It turns out that in practice, it is very useful to solve equation (2) with another Runge–Kutta method, which Butcher table is

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array}.$$

Writing k for the temporal step size and now writing time steps with superscripts (i.e. $\mathbf{v}^j \approx \mathbf{v}(jk)$), this method applied to equation (2) gives the scheme

$$\mathbf{v}^{j+1} = \mathbf{v}^j + \frac{k}{2}\mathbf{f}(t_j, \mathbf{v}^j) + \frac{k}{2}\mathbf{f}(t_{j+1}, \mathbf{v}^{j+1}).$$

By insering the definition of $\mathbf{f}$ from equation (2), the above is written out as

$$\begin{pmatrix} v_1^{j+1} \\ v_2^{j+1} \\ \vdots \\ v_{N-2}^{j+1} \\ v_{N-1}^{j+1} \end{pmatrix} = \begin{pmatrix} v_1^{j} \\ v_2^{j} \\ \vdots \\ v_{N-2}^{j} \\ v_{N-1}^{j} \end{pmatrix} + \frac{k}{2h^2} \begin{pmatrix} v_2^j - 2v_1^j + L(jk) \\ v_3^j - 2v_2^j + v_1^j \\ \vdots \\ v_{N-1}^j - 2v_{N-2}^j + v_{N-3}^j \\ R(jk) - 2v_{N-1}^j + v_{N-2}^j \end{pmatrix} + \frac{k}{2h^2} \begin{pmatrix} v_2^{j+1} - 2v_1^{j+1} + L((j+1)k) \\ v_3^{j+1} - 2v_2^{j+1} + v_1^{j+1} \\ \vdots \\ v_{N-1}^{j+1} - 2v_{N-2}^{j+1} + v_{N-3}^{j+1} \\ R((j+1)k) - 2v_{N-1}^{j+1} + v_{N-2}^{j+1} \end{pmatrix}.$$

[2] *Beregningsmolekyl* or *stensil* in Norwegian.

Rearranging into a matrix equation and abbreviating $r = k/h^2$, we get the tridiagonal linear system

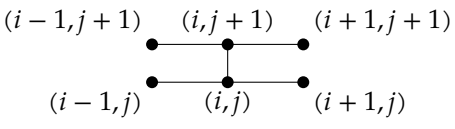
$$\begin{pmatrix} 1+r & -r/2 & & & \\ -r/2 & 1+r & -r/2 & & \\ & -r/2 & \ddots & \ddots & \\ & & \ddots & \ddots & -r/2 \\ & & & -r/2 & 1+r \end{pmatrix} \begin{pmatrix} v_1^{j+1} \\ v_2^{j+1} \\ \vdots \\ v_{N-2}^{j+1} \\ v_{N-1}^{j+1} \end{pmatrix} = \begin{pmatrix} v_1^j \\ v_2^j \\ \vdots \\ v_{N-2}^j \\ v_{N-1}^j \end{pmatrix} + \frac{r}{2} \begin{pmatrix} v_2^j - 2v_1^j + L(jk) + L((j+1)k) \\ v_3^j - 2v_2^j + v_1^j \\ \vdots \\ v_{N-1}^j - 2v_{N-2}^j + v_{N-3}^j \\ R(jk) + R((j+1)k) - 2v_{N-1}^j + v_{N-2}^j \end{pmatrix}. \tag{3}$$

Stepping in time (advancing from one blue line to the next in figure 1) according to equation (3) is the Crank–Nicolson method, with v_i^j the numerical approximation of $u(ih, jk)$.

2.1 Some observations

Crank–Nicolson amounts to solving the $(N-1)$-dimensional linear system in equation (3) for each time step, so the method is *implicit*. Note however that the matrix is strictly diagonally dominant, so an iteration technique like Jacobi or Gauss–Seidel can be employed to solve the system. Moreover, the matrix is the same for each time step, so exact methods can re-use a lot of previously done work (for example LU factorization) every time step. The fact that the matrix is tridiagonal also opens the door to a lot of other shortcuts.

The stencil of Crank–Nicolson is



,

nicely reflecting the fact that the RK method used is a mixture of forward and backward Euler.