Implementation of a stable numerical scheme to solve the 1D diffusion equation with variable coefficients and source terms subject to Neumann boundary conditions

Jake Aylmer

April 1, 2018

1 Problem statement

The one-dimensional diffusion equation for tracer $q(x,t)$, defined on $0 < x < L$ and $t \geq 0$, with source $S(x,t)$ and diffusivity $k(x)$ is given by

$$\frac{\partial q}{\partial t} - \frac{\partial}{\partial x}\left[k(x)\frac{\partial q}{\partial x}\right] = S(x,t), \tag{1}$$

which is subject to Neumann boundary conditions (zero flux of q at the boundaries)

$$k(0)\frac{\partial q}{\partial x}\Big|_{x=0} = k(L)\frac{\partial q}{\partial x}\Big|_{x=L} = 0 \tag{2}$$

and specified initial conditions

$$q(x,0) = q_0(x). \tag{3}$$

2 Analytic solution for a limiting case

Equation (1) can be solved analytically in the limit of no sources or sinks ($S(x,t) = 0$) and constant diffusivity $k(x) = k_0$. This is derived by using the method of separation of variables and expressing the general solution as an infinite sum of normal modes with coefficients found using Fourier analysis. The final result is given here and can be used as a test for the numerical solution to the full diffusion equation (1). The solution to

$$\frac{\partial q}{\partial t} = k_0\frac{\partial^2 q}{\partial x^2} \tag{4}$$

is given by

$$q(x,t) = a_0 + \sum_{n=1}^{\infty} a_n \exp\left(-k_0\left(\frac{n\pi}{L}\right)^2 t\right)\cos\left(\frac{n\pi}{L}x\right) \tag{5}$$

where the coefficients a_n are those found in the cosine series expansion of the initial conditions $q_0(x)$:

$$a_0 = \frac{1}{L}\int_0^L q_0(x)\mathrm{d}x \tag{6}$$

$$a_n = \frac{2}{L}\int_0^L q_0(x)\cos\left(\frac{n\pi}{L}x\right)\mathrm{d}x \quad n \geq 1. \tag{7}$$

2.1 Specific case for testing

Choose initial conditions

$$q_0(x) = q_{\max} x(L - x) \tag{8}$$

where $q_{\max}$ is a constant corresponding to the maximum of q_0 occuring at $x = L/2$. This allows for the a_n to be easily solved from equation (7):

$$a_0 = \frac{q_{\max} L^2}{6}, \quad a_n = -2q_{\max}\left(\frac{L}{n\pi}\right)^2 [1 + (-1)^n] \quad n \geq 1 \tag{9}$$

(i.e. $a_n = 0$ if n is odd). The solution is thus

$$q(x,t) = \frac{q_{\max} L^2}{6} - \frac{4q_{\max} L^2}{\pi^2} \sum_{n \text{ even}} \frac{1}{n^2} \exp\left(-k_0 \left(\frac{n\pi}{L}\right)^2 t\right) \cos\left(\frac{n\pi}{L} x\right). \tag{10}$$

In the limit $t \to \infty$, $q(x,t) \to q_{\max} L^2/6$.

3 Implementation of numerical solution

The numerical scheme is summarised here but not derived.[1] The x domain is discretised into N grid cells, labelled from $j = 0$ to $j = N - 1$, each of width $h = L/N$. q and S are then defined at the grid-cell centres, and expressed as N-element vectors $\mathbf{q} = [q(x_0), ..., q(x_j), ..., q(x_{N-1})]^{\mathrm{T}}$ and $\mathbf{S}$ defined similarly, where $x_j \equiv (j + 1/2)h$.[2] See figure (1).

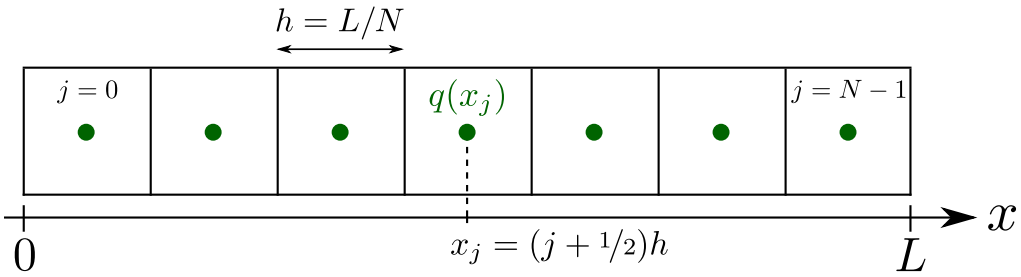

Figure 1: Schematic showing how the domain is discretised into N cells of equal widths h. q (green) is calculated at grid cell centres (as is S).

Equation (1) is re-written in the semi-discretised form

$$\frac{d\mathbf{q}}{dt} = \mathsf{A}\mathbf{q} + \mathbf{S} \tag{11}$$

where the $(N \times N)$ matrix A takes the place of the operator $\partial_x(k(x)\partial_x)$ in equation (1). A accounts for the boundary conditions of the problem: *here and in the code only Neumann boundary conditions are used.* The result is as follows for elements A_{ij} where i refers to the row index and j the column index, and $k_j \equiv k(x_j) = k((j + 1/2)h)$. For $i \neq 0$, $i \neq N - 1$:

$$h^2 \mathsf{A}_{ij} = \begin{cases} -(k_{j+1/2} + k_{j-1/2}) & i = j \\ k_{j\pm 1/2} & i = j \pm 1, \end{cases} \tag{12}$$

[1] See, for example, http://www.csc.kth.se/utbildning/kth/kurser/DN2255/ndiff13/Lecture3.pdf

[2] Technically, $q(x_j)$ should be written as, say, $\tilde{q}(x_j)$, to indicate that it is the integral average over the grid cell, but this subtlety is ignored for simplicity

and for the boundaries:

$$h^2\mathsf{A}_{00} = -k_{1/2}$$
$$h^2\mathsf{A}_{01} = k_{1/2}$$
$$h^2\mathsf{A}_{(N-1),(N-2)} = k_{N-3/2}$$
$$h^2\mathsf{A}_{(N-1),(N-1)} = -k_{N-3/2}. \tag{13}$$

Equation (11) is then discretised in time (to give the fully-discrete scheme) using the θ-method. θ is a parameter between 0 and 1 which determines whether the scheme is explicit or implicit and first or second order accurate. This results in

$$\mathbf{q}^{n+1} = [\mathsf{I} - \theta\Delta t\mathsf{A}]^{-1}\left[(\mathsf{I} + (1-\theta)\Delta t\mathsf{A})\,\mathbf{q}^n + \Delta t\mathbf{S}_\theta^n\right] \tag{14}$$

where Δt is the time-step, I is the N-dimensional identity matrix, superscript n refers to the time level and

$$\mathbf{S}_\theta^n = \theta\mathbf{S}^{n+1} + (1-\theta)\mathbf{S}^n. \tag{15}$$

The value of θ can be set in the code and corresponds to:

θ	Name	Type	Accuracy
0	Forward-Euler	Explicit	1^{st} order
$1/2$	Crank-Nicolson	Implicit	2^{nd} order
1	Backward-Euler	Implicit	1^{st} order

An example of the numerical scheme being used to solve the diffusion equation (4) with $k_0 = 2.5 \times 10^{-3}$ m^{2}s^{-1}, $L = 2.0$ m and initial conditions (8) with $q_{\max} = 4$ is shown in figure (2). This was carried out on a grid with $N = 20$ ($\Rightarrow h = 0.1$ m) and using time-step $\Delta t = 5$ s, for 6 time steps. The Forward Euler method ($\theta = 0$) is severely time-step restricted and is not plotted.

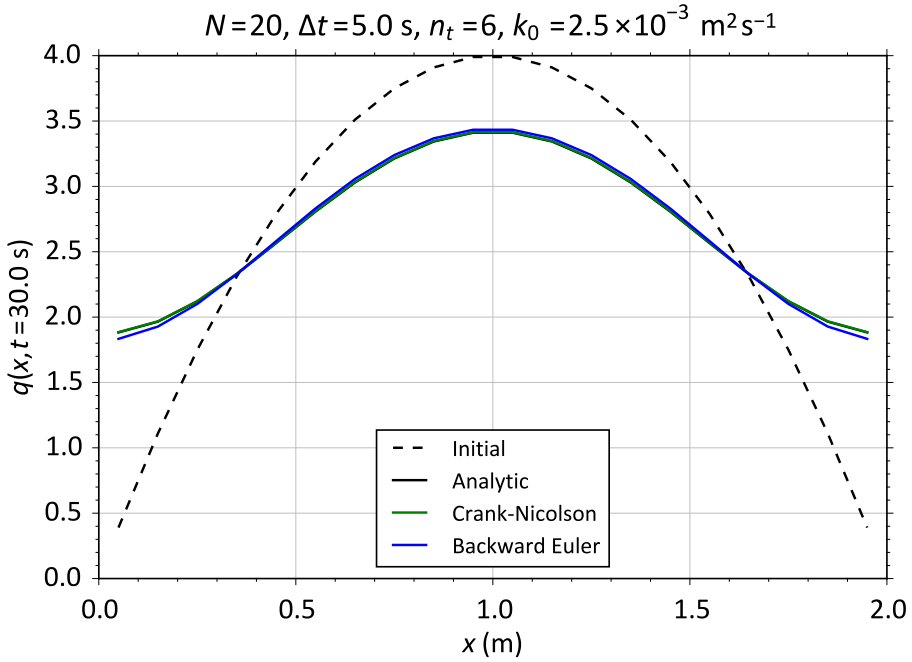

Figure 2: Example solution of the simplified diffusion equation (4) with initial conditions (8) (dashed-line), $L = 2.0$ m, $q_{\max} = 4.0$ and other parameters as listed at the top of the figure. The analytic solution (10) is plotted (black) but is obscured by the numerical solutions, which were calculated using the scheme implementation (14)-(15) setting $\mathbf{S} = 0$ and $k(x) = k_0$. The Crank-Nicolson solution (green) is more accurate than the Backward-Euler solution (blue) as expected (this can be seen by zooming in on the solutions). The Forward-Euler solution is not plotted because it requires a significantly smaller time-step for numerical stability.