AEM 4253/5253, Fall 2022, Homework 3, Due Friday December 9

Notes:

(1) Do not submit code with this HW. Upload a *single* PDF of your HW (either scanned from a legible handwritten version or a typed copy) to Canvas. All your plots must be computer-generated.
(2) Clearly label the axes on all your plots (and indicate the problem # in the caption).
(3) Use colors and/or symbols to differentiate different solutions on the plots.

$\boxed{\textbf{1.}}$ At time $t = 0$, a concentrated spot of dye is introduced into the centerline of a straight pipe with still fluid. The section we are looking at is 10 units long ($0 \leq x \leq 10$) and the dye is injected at position $x = x_0$. A reasonably good model for the concentration C is the diffusion equation

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2},$$

where D is the diffusion coefficient. If we assume that the initial spot (at $t = 0$) was a delta-function (this is an idealization), the exact solution at $t > 0$ is given by

$$C_{exact}(x,t) = \frac{1}{\sqrt{4\pi Dt}} \exp\left(\frac{-(x - x_0)^2}{4Dt}\right).$$

Assume that $x_0 = 5$ and $D = 0.1$. For the times we will be looking at, it is acceptable to use $C(0,t) = C(10,t) = 0$ as (fixed) boundary conditions.

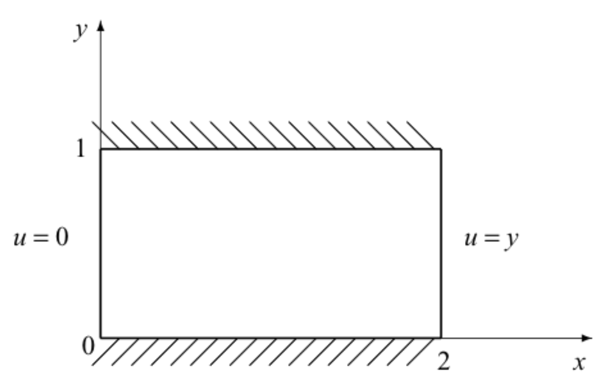
Write a program to solve the equation on an N cell mesh using a second order centered finite-difference scheme for the spatial term. Use time $t = 2$ as the starting point (evaluate the exact solution given above at $t = 2$ and use this as the initial condition) and time step till $t = 4$. Plot your numerical solution and the exact solution at $t = 2$ and $t = 4$ on a grid with $N = 200$ points using

a) RK4 time-stepping: report the stable time step.

b) The Crank-Nicolson method. Show that the scheme is stable for large CFL numbers. You should use a tridiagonal solver to solve the system of equations.

NOTE: The wikibooks entry: `https://en.wikibooks.org/wiki/Algorithm_Implementation/Linear_Algebra/Tridiagonal_matrix_algorithm` has source code with implementation of the Thomas algorithm in Fortran, C and Matlab.

$\boxed{\textbf{2.}}$ The steady state temperature distribution $u(x, y)$ in the rectangular copper plate below satisfies Laplace's equation,



$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0.$$

The upper and lower boundaries are perfectly insulated ($\partial u/\partial y = 0$); the left side is kept at 0° C, and the right side at $f(y) = y°$ C. The exact solution can be obtained analytically using the method of separation of variables and is given by

$$u(x,y) = \frac{x}{4} - 4\sum_{n=1}^{\infty} \frac{\sinh(n\pi x)\cos(n\pi y)}{(n\pi)^2 \sinh(2n\pi)}, \qquad n \text{ is odd.}$$

First write a program to compute the steady state solution to the second-order finite-difference approximation of the heat equation using the Jacobi iteration method. You should use N_x and N_y uniformly spaced cells in the horizontal and vertical directions, respectively. Use a first-order approximation for the Neumann boundaries.

- Now with $N_x = 21$ and $N_y = 21$ apply the Jacobi iteration to the discrete equations until the solution reaches steady state. To start the iterations, initialize the array with zeroes. To define "steady state", monitor the value of the change in solution between iterations:

$$res = \sqrt{\left\{\sum_{i=2}^{N_x-1}\sum_{j=2}^{N_y-1}[u^{(k+1}(i,j) - u^{(k)}(i,j)]^2\right\}/((N_x-2)(N_y-2))}$$

  and set a tolerance of 10^{-6}. Compare the numerical approximation to the analytical solution with a contour plot (use different line styles for the numerical and analytical isotherms).
- Repeat the previous case using the Gauss-Seidel iterative method.
- Repeat using SOR. First find the optimal value of ω for which convergence is fastest - do this by varying ω over the range $0 < \omega < 2$ and plotting the number of iterations it takes for convergence.