ME-474 Numerical Flow Simulation

Comments on the exercise: unsteady convection-diffusion

Fall 2021

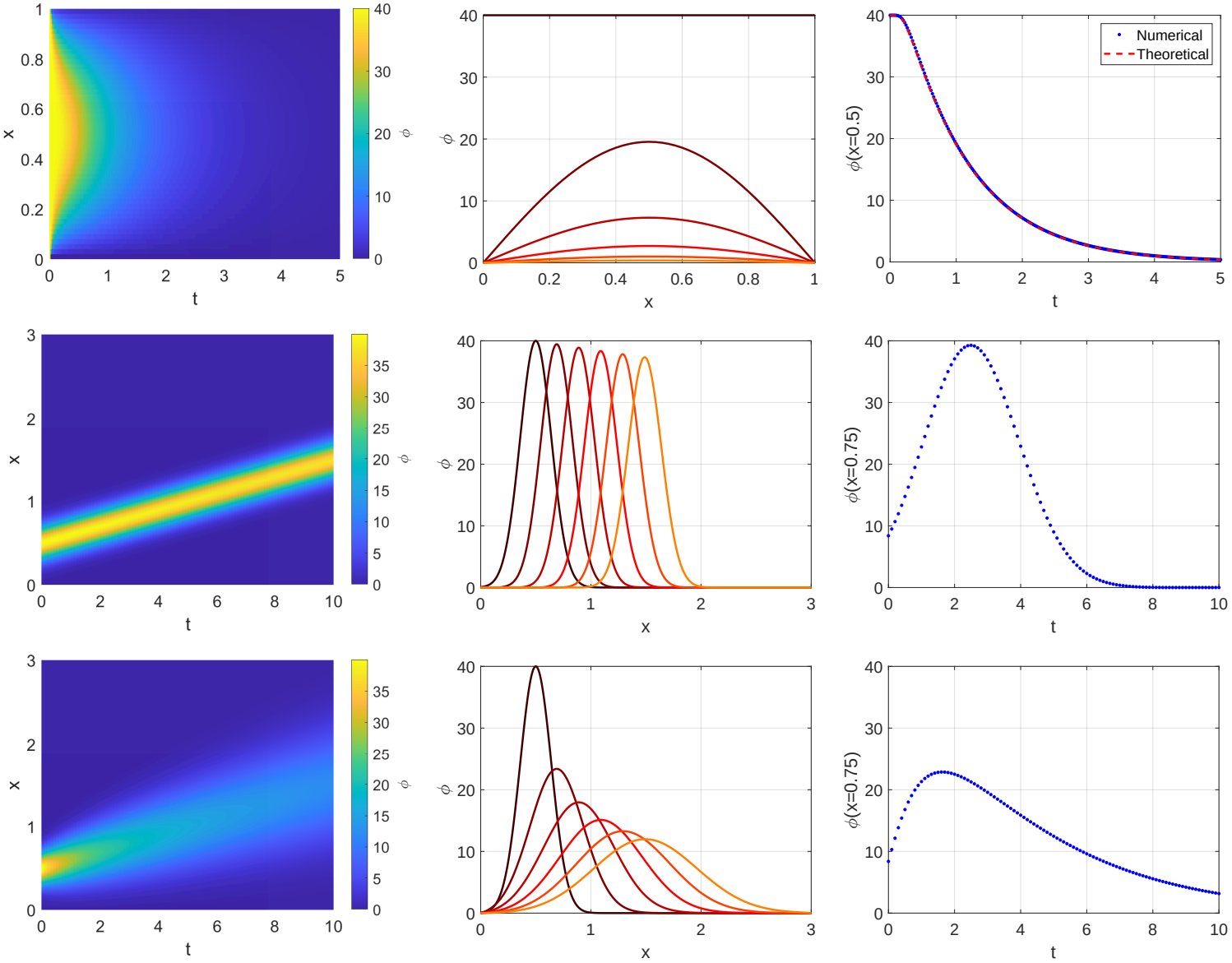

Figure 1: Top row: pure diffusion. Middle row: pure convection. Bottom row: convection-diffusion. Left column: Space-time plot of $\phi(x,t)$. Middle column: snapshots $\phi(x)$ at different time instants. Right column: time evolution $\phi(t)$ at a given location.

In this time-dependent problem, the solution ϕ depends on both x and t. There are several ways to implement that numerically. One of them is to store the solution in a 2-dimensional array: `Phi=zeros(nx,nt)`. Each column is a space-dependent "snapshot" (fixed time), and each row is a time-dependent "pointwise measurement" (fixed location). Storing the whole solution is possible when the number of mesh elements and time steps is reasonably small. Otherwise, instead of storing the whole history, it is more practical to store only the current solution (a 1-dimensional array of size `nx`) and to overwrite the previous solution.

Note that while the RHS b changes at every time step because it depends on the solution at the previous time step, the coefficients of the matrix do not change. This results from the linearity of the equation (the matrix would change at each iteration if the equation was nonlinear). Therefore, it is faster to define $\mathbf{A}$ once for all, before the temporal loop.

You can check that the Crank-Nicolson and implicit Euler schemes are stable for any Δt. However, they may produce oscillations if the condition on Δt and either $\rho(\Delta x)^2/\Gamma$ or $\Delta x/u$ is not verified (especially at the boundaries, if the initial condition that does not satisfy the boundary conditions). The explicit Euler scheme is unstable when Δt is too large.

Note that temporal discretization introduces some numerical diffusion: even when solving the pure convection case ($\Gamma = 0$), the initial condition is not only convected at speed u, it also diffuses as time increases. This is particularly visible here because the code uses upwind differencing (a first-order scheme) for the convective term.