Physics 427 – Introduction to Computational Physics
James B. Mertens
Homework 8 – due March 26, 2021

Problem 1: In class, we saw two examples of stability analysis for partial differential equations: the wave equation, and advection equation. Here we will look at the diffusion equation in one time and one space dimensions,

$$\partial_t f(t,x) = \partial_x^2 f(t,x)$$

with Eulerian integration in time and a centered second derivative. Written explicitly, the finite-difference version of this scheme is given by

$$f(t+\Delta t, x) = f(t,x) + \Delta t \frac{f(t,x+\Delta x) - 2f(t,x) + f(t,x-\Delta x)}{\Delta x^2}\,.$$

a) Given we can write the numerical solution for f as a combination of a true and error term, $f = f_{\text{true}} + \epsilon_f$, write an expression for the error $\epsilon_f(t+\Delta t, x)$ in terms of the error at an earlier time, $\epsilon_f(t, x+\Delta x)$, $\epsilon_f(t)$, and $\epsilon_f(t, x-\Delta x)$. You can assume the true part of the solution obeys the finite difference equation, canceling all of these terms, leaving you with a finite difference equation for ϵ_f.

b) Assuming the error takes the form

$$\epsilon_f(t,x) = E(t)e^{ikx}\,,$$

substitute this solution into the expression you derived in part a, simplify, and solve for $E(t+\Delta t)$. A useful identity is

$$e^{i\theta} + e^{-i\theta} = 2 - 4\sin^2(\theta/2)\,.$$

c) We would like the time-dependent part of the error to satisfy the inequality

$$|E(t+\Delta t)| \leq |E(t)|\,.$$

Substitute your expression from part b for $E(t+\Delta t)$. Is there any choice of Δt and/or Δx which can guarantee the inequality is satisfied, resulting in a stable method?

Problem 2: In class last week, we looked at the PDE_Integration notebook for solving the advection equation in one space and one time dimension. This notebook can be downloaded from Canvas.

a) Consider the case of a computational domain with *periodic* boundary conditions: we would like the value of the function at one end of the computational domain to match the value at the other.

In terms of the arrays storing our function, we will need to set function values at the boundaries (end) of the array to values just inside the boundary at the other end. In other words, we can use the conditions

```
f[0] = f[-2] # set first value in the array to the second-last
f[-1] = f[1] # set the last value in the array to the second
```

Implement this in the PDE_Integration notebook. Integrate for a longer time (e.g. up to `t_final = 120`) to verify the function "wraps around" the box.

b) Now consider how to implement periodic boundary conditions in 2 dimensions. In class, we examined the behavior of the wave equation in 2 dimensions; the PDE_integration notebook has been updated to include this code.

Determine an array slicing condition that can be used to implement periodic boundary conditions in this case, and so implement periodic boundary conditions in the notebook. You may need to integrate for a longer time in order to see the wave "pass through" the boundaries. Setting `t_final = 130` with `xs = np.linspace(-100, 100, 50)` works well for me. Because the integration method we are using is not always stable, you may find the integration routine breaks down if you integrate for too long, however this should not happen so quickly that you are unable to observe the wave passing through the periodic boundaries.

Problem 3: Complete the feedback survey on Canvas. This should be available early in the week, in the quizzes section of Canvas.