Homework 2: Learnable Finite Difference Stencils

ENM5320 Spring 2025, UPenn

Due date: Feb. 3^{rd} by midnight (11:59pm)

This is an **individual assignment**; you are encouraged to use any/all resources but you must attribute any resources used explicitly (collaborators/textbooks/journal papers/ChatGPT). Submit a handwritten or latexed report in Canvas, along with Jupyter notebooks, figures, or any other supporting files needed. Code should be documented sufficiently that we can run the code without modification and reproduce your results. Late submissions will follow the guidance outlined in the course syllabus.

This is your first open-ended assignment using PyTorch and numerical simulators. If you have gaps in your background on either on of those fronts, make sure you start the assignment early enough to get feedback at OH.

Instructions:

On the course github, you can find a Jupyter notebook providing step-by-step examples of how to use pytorch for basic learning tasks (`ENM5320 > Code >/PyTorchFDM.ipynb`, Link). You can click the link at the top of the git page to directly load into Colab. As we stepped through in class on 2/3, this code generates a trainable finite difference stencil; as always I strongly suggest you use Gemini to get a description of any aspects of the code that you don't understand. The objective of this assignment will be to adapt this code to reverse engineer a finite difference solver for the advection-diffusion equation and compare its performance to the traditional techniques discussed in class.

1. **Part 1. Generate a dataset.** The code provides an example of how to extract a stencil from a single initial condition $u(x,0) = f(x) = \sin(x-t)$. Generate a dataset consisting of N_{sol} solutions, $\mathcal{D} = \left\{ u_k = \sin(2^k(x-t)), \quad k = 0, 1, 2, ..., N_{sol} \right\}$.

2. **Part 2. Generate standard method comparison.** Solve using the explicit Euler method to a final time $T = 4\pi$. Generate a log-log plot of the dependence of the root-mean-square (RMS) error $||e|| = \sqrt{\frac{1}{N_{sol}} \sum_k ||u(x,T) - v(x,T)||^2}$, where u is the analytic solution and v is the finite difference solution. For this you will need to choose a resolution. To do this, select a reasonable number of timesteps N and experiment with varying the ratio $\lambda = k/h$. When you present your results, you should design your log-log plot to communicate the effect of λ and N.

3. **Part 3. Rerun with Lax-Friedrichs.** Rerun your code from the previous block using the Lax-Friedrichs scheme to introduce artificial viscosity. Comment in your report on any qualitative changes, and the effect of λ.

4. **Part 4. Learn a data-driven stencil.** Use the same dataset and the `PyTorchFDM.ipynb` code to learn a finite difference stencil on a given resolution. Once you have fit the stencil to the data, generate the same set of plots as from Parts 2,3 and compare the performance.