Homework assignment 2

August 30, 2019

To be handed in by 23:59 on Sunday September 8th. For all your codingexercises, I recommend that you use a Jupyter notebook which you can laterexport as PDF.

1. We want to replicate Figure 1.3 from the book, but for a *second* order derivative. For this, calculate the second order derivative of $f(x) = \cos(x)$ using a second order approximation:

$$f''(x) = [\cos(x + \Delta x) - 2\cos(x) + \cos(x - \Delta x)]/\Delta x^2 \tag{1}$$

and plot the error with respect to $f''(x) = -\cos(x)$ as in Fig 1.3. i.e. as a function of Δx. Mark the values of Δx where the error is dominated by truncation and roundoff.

2. Write down an explanation of the difference between hyperbolic, elliptical and parabolic PDEs, and why these differences matter when thinking about information travel and memory access.

3. Exercise 1.9 in the book. Learn how to use a FFT library or the built-in FFT method in your programming language of choice. There are various ways to define the normalization in the FFT, that can vary from one library to the next. To ensure that you are doing things correctly, compute the following transforms:

- $\sin(2\pi k_0 x)$ with $k_0 = 0.2$. The transform should have all of the power in the imaginary component only at the frequency 0.2.
- $\cos(2\pi k_0 x)$ with $k_0 = 0.2$. The transform should have all of the power in the real component only at the frequency 0.2.
- $\sin(2\pi k_0 x + \pi/4)$ with $k_0 = 0.2$. The transform should have equal power in the real and imaginary components, only at the frequency 0.2. Since the power is 1, the amplitude of the real and imaginary parts will be $1/\sqrt{2}$.

4. Modify the Explicit solver from Homework 1, Problem 4 to be an first order backwards Euler solver. To keep it simple, do not include any iteration, just base yourself on Equation 1.35. Begin by analytically calculating the Jacobian. Measure the convergence of the scheme.