MA/CS 615. Numerical PDE: Assignment I

January 27, 2026

1 Convergence of Fourier interpolants

Consider interpolating the following periodic function on the interval $x \in [-\pi, \pi]$

$$f(x) = \frac{\exp(a \cos x)}{2\pi I_0(a)},$$

where a is a given parameter that determines the smoothness of the function, and $I_0(a)$ is a Bessel function used for normalization purposes, available in MATLAB as *besseli(0, a)*. For the majority of this assignment fix $a = 3$, for example. Think about how the numerical/truncation errors depend on a. For your own benefit, you could also try some less smooth function and compare the results.

The goal of this exercise is to see whether and how fast the interpolation error converges to zero as the number of interpolation nodes increases, for the Fourier (trigonomentric) interpolant $\phi(x)$ on n equi-spaced nodes. For a given interpolant $\phi(x)$, we can evaluate the interpolant on a fine grid of points, for example, $\tilde{x} = \text{linspace}(-\pi, \pi, N+1)$ for $N = 1000$, and then compare to the actual function $f(x)$. We can also compute an estimate for the Euclidean norm of the interpolant error by summing the error over the fine grid

$$E_2[\phi((x))] = \left[h \sum_{i=0}^{N} |f(\tilde{x}_i) - \phi(\tilde{x}_i)|^2\right]^{1/2} \approx \left[\int_{-\pi}^{\pi} |f(x) - \phi(x)|^2 \, dx\right]^{1/2},$$

where $h = 2\pi/N$. *For the Fourier polynomial, you can write your own code for evaluating it at the fine grid $\tilde{x}$ by simply writing down the explicit form of the Fourier series, or, with some ingenuity, you can do it faster by doing an inverse FFT, as done in MATLAB's function* interpft. *Include code and explanation in the solution. Pay close attention to the ordering of the frequencies and use an odd number of points n and also the built-in function* fftshift *to make it more manageable; of course, if you can also make it work for even points that's great.*

1.1 Quick qualitative tests

For a given small n, say $n = 8$ if you got even points to work or $n = 9$ otherwise, plot the interpolant together with the function and see how good it is. Plot the error $\epsilon(x) = |f(x) - \phi(x)|$ of the interpolant for a larger n, say $n = 32$ or $n = 27$, and visually compare the accuracy in different regions of the interval, and comment on what you observe.

Plot the magnitude of the Fourier coefficients (i.e., the Fourier spectrum) $\left|\hat{f}\right|$ versus frequency for a large n [*Hint: Large means that the high-frequency components become smaller than roundoff so that further increasing n does not make a difference numerically.*]

1.2 Interpolation Error

Compute the estimated interpolation error E_2 of the Fourier interpolant for different numbers of nodes n, and then plot the error versus n.

Comment on how fast the error converges and why. Verify that your numerical results are consistent with theoretical predictions from class [*Hint: Spectral convergence is so fast that numerical roundoff will not permit really seeing the exponential decay well, but simply plotting an exponentially-decaying curve on the same plot or plotting the error on a log-linear scale will do*].

1.3 Approximating derivatives

An approximation to the first derivative $f'(x)$ can be obtained by simply differentiating the interpolant, $f'(x) \approx \phi'(x)$.

For the Fourier interpolant, compute $\phi'(x)$ and compare to the correct derivative $f'(x)$ on the same plot, for some n. Consider efficiency when writing the code and write down any formulas used in the report.

Compute error estimates for the derivative and plot how they depend on n, and compare to theoretical expectations.

1.4 Approximating integrals

An approximation to the integral of $f(x)$ can be obtained by integrating the interpolant instead.

For the Fourier interpolant, compute $\int_0^x \phi(t)\,dt$ for $x \in [-\pi, \pi]$ and compare to the correct answer $\int_0^x f(t)\,dt$ on the same plot, for some n. Explain how you got the "correct answer". Consider efficiency when writing the code and write down any formulas used in the report.

Compute error estimates for the integral and plot how they depend on n, and compare to theoretical expectations.

2 Spatial discretization of KdV

Consider the Korteweg de Vries (KdV) PDE

$$\partial_t u = \mathcal{K}\left[u(\cdot, t)\right] = -\partial_{xxx} u - 3\partial_x\left(u^2\right),$$

where $\mathcal{K}\left[u(\cdot, t)\right]$ denotes the functional on the right hand side, which only involves derivatives of x.

On an unbounded line, this has traveling wave solutions in the forms of "solitons"

$$u_{\mathrm{sol}}(x, t) = \frac{c}{2}\mathrm{sech}^2\left(\frac{\sqrt{c}}{2}(x - ct)\right), \tag{1}$$

where c is the speed of the soliton moving to the right. While the soliton solution (1) is not periodic, it decays exponentially for large x so we can pretend it is periodic. Here we will consider the KdV equation on a periodic domain $x \in [-L/2, L/2]$ with $L = 60$, and set $c = 1$. Note that direct differentiation and some algebra gives

$$\partial_t u_{\mathrm{sol}}(x, t = 0) = \frac{1}{2}\frac{c^{5/2}\sinh\left(1/2\sqrt{c}x\right)}{\left(\cosh\left(1/2\sqrt{c}x\right)\right)^3}.$$

In a series of assignments related to KdV we will consider pseudo-spectral methods for solving this PDE, in which we represent the solution as a Fourier series [*Notice: Because we are not on the standard interval the wavenumber k is not an integer but rather has units of inverse length.*]

$$u(x, t) = \sum_k \hat{u}_k(t)\, e^{ikx}.$$

Recall that a pseudo-spectral method approximates the r.h.s of the KdV equation in Fourier space as

$$\widehat{\mathcal{K}[u]} = \mathcal{F}\left(\mathcal{K}[u]\right) = \mathbf{F}(\hat{u}) = ik^3 \odot \hat{u} - 3ik \odot \mathcal{F}\left(\left(\mathcal{F}^{-1}\hat{u}\right)^{\odot 2}\right),$$

where $\mathcal{F}$ denotes a Fourier transform. This allows us to convert the KdV PDE into the system of ODEs

$$\frac{d\hat{u}(t)}{dt} = \mathbf{F}(\hat{u}(t)).$$

To explore how to smoothness of the solution in space affects the accuracy, in addition to the soliton wave you will consider a triangle wave

$$f(x) = \begin{cases} 10 + x, & x < 0 \text{ and } x \geq -10 \\ 10 - x, & x \geq 0, \text{ and } x \leq 10 \\ 0, & \text{otherwise} \end{cases}$$

and the sawtooth wave

$$g(x) = \begin{cases} 10 + x, & x < 0 \text{ and } x \geq -1 \\ 0, & \text{otherwise} \end{cases}.$$

Note: This problem works with functions and function norms, not with vectors and vector norms. Write down what formula you used for the different function norms in the report. In order to approximate function norms, you need to evaluate the solution on a finer grid, just like in Problem 1. So if you use N points for the Fourier transforms, evaluate the errors/functions on a grid of $M \gg N$ points using your interpft or the built-in interpft. Think about how to choose M in a way that your answers don't really depend on M, if that is possible, just like you want to plot a function with enough points so you cannot see the fact it is only sampled on a finite grid.

2.1 Error in initial condition

If we start with an initial condition $u(x, t = 0) = u_{\text{sol}}(x, t = 0)$, then the solution is given by the traveling soliton wave (1). However, we make an error already at $t = 0$ just by approximating this initial condition with a truncated Fourier series $\tilde{u}(x, t)$ with N Fourier modes. Notice that we studied this error in Problem 1 so reuse your codes for this part.

Compute how the discretization error in the initial condition depends on the number of Fourier modes N for different function norms ($L_{1/2/\infty}$) when the initial condition is $u_{\text{sol}}(x, t = 0)$. Do you see a difference between the different norms? To make the comparison fair use relative error

$$\epsilon = \frac{\|u - \tilde{u}\|}{\|u\|}.$$

Plot the error $u - \tilde{u}$ for the triangle and sawtooth waves, and also see whether you see a difference between the different norms for these non-smooth initial conditions. Explain any differences through the spectrum of u.

2.2 Pseudo-spectral approximation of $\mathcal{K}[u(\cdot)]$

Write code to evaluate $\mathbf{F}(\hat{u})$ for a given grid size N.

Compute the error in the approximation of $\mathcal{K}[u_{\text{sol}}(\cdot, t = 0)]$ in different function norms (is there a difference and why?) and see how the error changes with the number of points N. Is this discretization spectrally accurate?

How many points N do you need to obtain 9 digits of accuracy? Can you relate that to the spectrum of u?

What if the constant c and/or the length L of the periodic domain $x \in [-L, L]$ changed – how do you expect N to change and why? [*It is always a good idea to confirm your predictions numerically.*]

2.3 Aliasing error

Implement anti-aliasing and compare the numerical approximations of $\mathcal{K}\left[u_{\mathrm{sol}}\left(\cdot, t=0\right)\right]$ with and without anti-aliasing, and explain whether and when you see a noticeable difference and when you see no difference, and why.

Repeat for the triangle and square waves and comment on your observations. Note that for these functions the r.h.s. of the KdV equation does not have a classical interpretation so it is not really clear it makes sense to talk about an error in the approximation, however, we can certainly (naively) solve the system of ODEs and see what we get.