Numerical Methods II, Spring 2023
Assignment I: Fourier Interpolation and Approximation

Aleksandar Donev
Courant Institute, NYU, donev@courant.nyu.edu

Posted: Jan 8th 2023
Due: **5pm on Tuesday 2/7, 2023**

The optional parts of this assignment do not carry any points; only do optional things if you have time and interest for your own benefit.

In this homework, use the built in fft/ifft functions, not your own. You may write them for practice but part of the exercise here is to use existing optimized libraries (reading the documentation, etc.).

1 [65 points] Convergence of Fourier Interpolants

In this problem we 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.

[**Optional**] Think about how the numerical/truncation errors depend on a. For your own benefit, you should 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. [*Hint: Note that for Fourier transforms periodicity is already assumed so you should include only one of the points $x = 0$ and $x = \pi$, not both.*]

For a given interpolant $\phi(x)$, we can evaluate the interpolant on a fine grid of points, for example, $\tilde{x} = linspace(-\pi, \pi, N + 1)$ for $N = 1000$, and then compare to the actual function $f(x)$. We can also compute an estimate for the Euclidean norm of the interpolation 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.* ***Write your own fast interpft routine*** *assuming $N \geq n + 1$, and 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 [10 pts] Quick qualitative tests

[5 pts] 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 $\varepsilon(x) = |f(x) - \phi(x)|$ 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.

[5 pts] 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 [15 pts] Interpolation Error

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

[10 pts] 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 [20 pts] Approximating derivatives

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

[10 pts] 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.

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

1.4 [20 pts] Approximating integrals

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

[10 pts] 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.

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