Assignment 5.

Due Friday, Feb. 10.

Reading: Ch. 3 in the text. Material from course web page on fast Poisson solvers and the FFT.

1. Use the 9-point formula with the correction term described in Sec. 3.5 to solve

\[ u_{xx} + u_{yy} = f(x, y), \quad 0 < x, y < 1 \]

\[ u(x, 0) = u(x, 1) = u(0, y) = u(1, y) = 1. \]

Take \( f(x, y) = x^2 + y^2 \), and demonstrate numerically that your code achieves fourth order accuracy at the nodes. [Note: If you do not know an analytic solution to a problem, one way to check the code is to solve the problem on a fine grid and pretend that the result is the exact solution, then solve on coarser grids and compare your answers to the fine grid solution. However, you must be sure to compare solution values corresponding to the same points in the domain.]

2. The key to efficiency in the chebfun package, which you used in a previous homework exercise, is the ability to rapidly translate between the values of a function at the Chebyshev points, \( \cos(\pi j/n) \), \( j = 0, \ldots, n \), and the coefficients \( a_0, \ldots, a_n \), in a Chebyshev expansion of the function’s \( n \)th-degree polynomial interpolant: 

\[ p(x) = \sum_{j=0}^{n} a_j T_j(x), \]

where \( T_j(x) = \cos(j \arccos(x)) \) is the \( j \)th degree Chebyshev polynomial. Knowing the coefficients \( a_0, \ldots, a_n \), one can evaluate \( p \) at the Chebyshev points by evaluating the sums

\[ p(\cos(k\pi/n)) = \sum_{j=0}^{n} a_j \cos(jk\pi/n), \quad k = 0, \ldots, n. \]  \( (1) \)

These sums are much like the real part of the sums in the FFT,

\[ F_k = \sum_{j=0}^{n-1} e^{2\pi i j k/n} f_j, \quad k = 0, \ldots, n - 1, \]

but the argument of the cosine differs by a factor of 2 from the values that would make them equal. Explain how the FFT or a closely related procedure could be used to evaluate the sums in (1). To go in the other direction, and efficiently determine the coefficients \( a_0, \ldots, a_n \) from the function values \( p(\cos(k\pi/n)) \), what method would you use?