Homework 4
Math 128A
Due Tuesday, November 26th

1. (a) Let L_n represent the degree n Legendre polynomial. Show that for any polynomial p of degree less than n,
$$\int_{-1}^{1} L_n(x)p(x)\,dx = 0.$$
That is, the n^{th} degree Legendre polynomial is orthogonal to all polynomials of degree less than n.

   (b) Let $\tilde{L}_n$ represent the n^{th} degree monic Legendre polynomial. Show that
$$\left\|\tilde{L}_n\right\|_2^2 \leq \|\tilde{p}_n\|_2^2$$
for all monic polynomial of degree n, $\tilde{p}_n$, where the norm is
$$\|f\|_2^2 = \langle f, f\rangle = \int_{-1}^{1} (f(x))^2\,dx.$$
Hint: Let $w(x) = \tilde{p}_n(x) - \tilde{L}_n(x)$, and compute $\|\tilde{p}_n\|_2^2$.

   This result shows that the degree n monic Legendre polynomial is the degree n monic polynomial with smallest 2-norm. The results from both part (a) and (b) are important for high-order accurate integration methods, specifically Gaussian quadrature.

2. The zeros of the Chebyshev polynomials are the optimal points for interpolation on $[-1, 1]$ in the sense that they give the minimum maximum of the polynomial in the error bound. In practice we often use the extreme points of the Chebyshev polynomials because they include the endpoints and operations on the interpolant can be done quickly using the FFT.

   (a) For the function $f(x) = \exp\left(-(x-0.5)^2\right)$ on $[-1, 1]$ construct the polynomial interpolant P_n of degree n for $n = 4, 8, 12, 16$ using
      i. The $n+1$ equally spaced points $x_k = -1 + 2k/n$ for $k = 0 \ldots n$,
      ii. The $n+1$ extreme points of the degree n Chebyshev polynomial,
      iii. The $n+1$ zeros of the degree $n+1$ Chebyshev polynomial.

   For each n plot the approximation errors ($|P_n(x) - f(x)|$) of the three different interpolants on the same graph. Make a table of the maximum approximation error for each n for each interpolant. Discuss the results.

   (b) Repeat the previous part for $f(x) = (16x^2 + 1)^{-1}$

3. (a) Let f be a 2π-periodic function defined by the Fourier series
$$f(x) = \sum_{k=-\infty}^{\infty} c_k \exp(ikx).$$

Suppose that $f(x)$ is real valued, which means that $c_{-k} = \bar{c}_k$, where the overbar represents the complex conjugate, $\overline{a+bi} = a - bi$. For real valued functions, the Fourier series can be written as

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(kx) + b_k \sin(kx) .$$

Suppose you are given the complex coefficients c_k. Use that $\exp(i\theta) = \cos(\theta) + i\sin(\theta)$, and derive expressions for a_k and b_k in terms of c_k.

(b) Write a program which takes as input x and the complex Fourier coefficients c_k for $k = 0 \ldots m$ of a real valued function, f, and returns $f(x)$. Use the table of coefficients below and make a plot of $f(x)$.

k	c_k
0	0.0
1	$0.0407 - 0.0015i$
2	$0.1645 - 0.0167i$
3	$0.4382 - 0.0993i$
4	$-0.4112 - 0.7080i$
5	$-0.1492 - 0.9888i$
6	$-0.4125 + 0.7072i$
7	$-0.4447 - 0.0641i$
8	$-0.0242 + 0.1635i$

4. The file `noisy_signal.txt` contains pairs of points $(t_k, f(t_k))$ which represents 1000 equally spaced samples of the function f on the time interval $[0, 2\pi)$.

(a) Make a plot of the data, f vs. t. In Matlab, you can read in the data using the command `load`, and in Pytyon use the `loadtxt` command in the `numpy` library.

(b) Use the fft command to compute the discrete Fourier transform of the data. Let c_k be the complex-valued Fourier coefficients. Make a plot of $|c_k|$ vs. k for $k = 0, \ldots, 500$. Use log scales for both axes. In both Matlab and Python, you can evaluate the modulus of a complex number using the `abs` function. What can you conclude from this plot?

There are 1000 data points, and the fft will return 1000 complex valued Fourier coefficients. The discrete Fourier coefficients from the fft algorithm (both Matlab and Python) are ordered as

$$(c_0, c_1, \ldots, c_{500}, c_{-499}, c_{-498}, \ldots, c_{-2}, c_{-1}).$$

(c) Let f_s represent a smoothed version of the f with the high frequency components removed. To smooth the data, filter out the high frequency components by setting the discrete Fourier coefficients, c_k, to zero for $|k| > 10$. To generate the time samples of f_s, use the `ifft` command. Plot f_s vs. t.