Midterm Project, due March 8

In this project, we study different ways to interpolate a given set of points, determined by the Runge's function. We consider the following four different approaches: *Polynomial interpolation on equally spaced points*, *Polynomial interpolation on Chebyshev points*, *Piecewise cubic interpolation*, and *Trigonometric interpolation.*

(a) Given a set of n equally space points on the interval $[-1, 1]$, by

$$t_i = -1 + \frac{2(i-1)}{n-1}, \quad y_i = \frac{1}{1+25t_i^2}, \quad i = 1, 2, 3, ..., n,$$

determine a polynomial of degree $n-1$ which interpolates through these given n points. Use two ways to generate the interpolation polynomial.

First, use the monomial basis to find the interpolation polynomial $p_{n-1}(t) = x_1 + x_2t + ... + x_nt^{n-1}$, generate the Vandermonde matrix and solve the linear equations for the coefficients $x_i, i = 1, ...n$. To evaluate interpolation polynomial at any given point t, you need to write a function $yout = mypolyval(x, t)$, which allows an efficient recursive evaluation based on Horner's scheme. In this function, the input $x = [x_1, ..., x_n]$ are the polynomial coefficients, t is an arbitrary value, the output $yout = p_{n-1}(t)$.

Second, use the Lagrange basis to find the interpolation polynomial of the form $p_{n-1}(t) = y_1L_1(t) + ... + y_nL_n(t)$, where $L_i(t)$ are Lagrange polynomial functions. Write a function $yout = mylagrange(tn, yn, t)$, where $tn = [t_1, ..., t_n], yn = [y_1, ..., y_n]$ are data points, t is an arbitrary value, the output $yout = p_{n-1}(t)$. In this approach, there is no equations to solve, most effort is to define efficiently all the Lagrange polynomials (use 'for' loops and 'if' conditions).

Choose different values of n, e.g., $n = 5, 10, 15, 20$. For each chosen n, first draw curves of two interpolation polynomials in the same graph and in a separate graph (or subplot) draw the difference of two polynomials. They should agree. Next, draw the curve of the Runge's function and the curve of the interpolation polynomial in the same graph to compare, and in a different graph (or subplot) draw the difference of two curves. Do the two curves match better with the increase of n?

(b) Choose the n interpolation points as the Chebyshev points defined by

$$t_i = \cos\frac{(2i-1)\pi}{2n}, \quad y_i = \frac{1}{1+25t_i^2}, \quad i = 1, 2, 3, ..., n.$$

Compare the interpolation polynomial and the Runge's function, so they match better with the increase of n?

(c) Given the same set of equally spaced interpolate points as in Problem **(a)**, use the Matlab built-in function *spline* to find the piecewise cubic spline interpolation function. Compare the piecewise cubic spline and the Runge's function, so they match better with the increase of n?

(d) Use pen, paper, and computer to find the periodic cubic spline that interpolates the data $(0, 0), (0.5, 1), (1, 0)$. The periodic boundary condition is given by $p_1'(t_1) = p_2'(t_3), p_1''(t_1) = p_2''(t_3)$.

(e) To apply the trigonometric interpolation more conveniently, we use the following set of $N = 2M + 1$ equally spaced points on the interval $[0, 2\pi]$:

$$x_k = k\frac{2\pi}{N}, \qquad f_k = f(x_k), \qquad k = 0, 1, 2, ..., N-1,$$

where M is a positive integer and the function $f(x)$ is given by $f(x) = \dfrac{1}{1 + 25(x-\pi)^2}$.

The trigonometric interpolation is determined by $\psi(x) = \dfrac{A_0}{2} + \displaystyle\sum_{h=1}^{M} (A_h \cos hx + B_h \sin hx)$, where

$$A_0 = \frac{2}{N}\sum_{k=0}^{N-1} f_k, \quad A_h = \frac{2}{N}\sum_{k=0}^{N-1} f_k \cos\frac{2\pi hk}{N}, \quad B_h = \frac{2}{N}\sum_{k=0}^{N-1} f_k \sin\frac{2\pi hk}{N}, \quad h = 1, 2, \ldots, M.$$

Compare the curve of the trigonometric interpolation $\psi(x)$ with the curve of the function $f(x)$, for $M = 2, 5, 7$, respectively.

Does the interpolation become improved with the increase of M?