Center for Complex Networks

Shahid Beheshti University
Department of Physics

Advanced Numerical Simulation

Problem Set 2

(Due Date : 1404/08/16)

Problems

1. **Lagrange Interpolation**
   In a laboratory experiment, measurements of a signal believed to follow a sine-like pattern were collected at discrete points and saved in the file `data21.txt`. Each line of the file contains two columns: the position x_i and the corresponding measured value $f(x_i)$, which includes small experimental noise. Your task is to analyze this dataset using the **Lagrange interpolation** scheme.

   (a) Read the data from `data21.txt`. Implement the Lagrange interpolation formula directly:

   $$P(x) = \sum_{i=0}^{n} f_i \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}.$$

   Do not use any built-in interpolation functions.

   (b) Use your implementation to evaluate the interpolated values at

   $$x = 0.8, \quad 2.3, \quad 4.1.$$

   Compare your results to the true (noise-free) values of the function $f(x) = \sin(x)$. Compute and report the **absolute** error.

   (c) Plot the following on the same figure:
   - the interpolating polynomial $P(x)$,
   - the exact function $\sin(x)$,
   - and the measured data points.
   - the absolute interpolation error $|f(x) - P(x)|$

   Use the interval $x \in [0, 6]$.

   (d) Discuss the interpolation behavior near the edges of the dataset and test how the polynomial performs when **extrapolating** to $x = 6.1$. Compare the extrapolated value with the true $\sin(6.1)$.

2. **Bonus: Least-squares approximation**
   In this exercise, you will apply the **least-squares approximation** method to fit a polynomial model to fit the data set `data22.txt`, along with the corresponding symmetric points $f(-x) = f(x)$.

(a) Read the data from `data22.txt`. Fit the data using polynomials of degree $n = 2, 4, 6, 8, 10$ and 12. For each degree, compute the coefficients of the polynomial $P_n(x)$ that minimizes

$$S = \sum_{i=1}^{N} [f(x_i) - P_n(x_i)]^2 .$$

Implement your own least-squares fitting algorithm without relying on built-in polynomial fitting functions and compare it with `numpy.polyfit`.

(b) Compare your fitted polynomial for $n = 12$ with the well-known analytical approximation of the Bessel function:

$$f(x) \approx 1-2.2499997x^2+1.2656208x^4-0.3163866x^6+0.0444479x^8-0.0039444x^{10}+0.0002100x^{12}$$

Evaluate and discuss the accuracy of your fitted coefficients relative to the analytical ones.

(c) Plot the data points, the exact function $J_0(3x)$, and the least-squares approximations for different polynomial degrees on the same figure for $x \in [-1, 1]$. On a separate plot, show the absolute error $|f(x) - P_n(x)|$ for each case.

(d) Discuss how increasing the polynomial degree affects the fitting accuracy and stability of the coefficients. Which degree provides the best balance between accuracy and numerical stability?

3. **Numerical Calculus**

(a) Solve exercises 3.1, 3.12, 3.16 of Tao Pang's book. You may skip the theoretical part of exercise 3.16.