1. **Least squares fitting - m-th degree polynomial through N points.**

   (a) Find a system of equations for the coefficients of an m-th degree polynomial that fits a function \( f \) at given data points \( (x_k, f(x_k)), k = 1, \ldots, N \) in the least squares sense. Write the equations in matrix form. What is the size of the matrix?

   (b) Solve the system of equations using Matlab for \( x_k = \{0, 0.2, 0.35, 0.5, 1\} \), \( f(x) = e^{-x} \), and \( m = 3 \). Evaluate the resulting polynomial at \( x = 0.7 \). Compare the answer to the “exact” value of \( e^{-0.7} \). Use the MATLAB function “polyfit” to check the results.

2. **Lagrange interpolating polynomial.**

   (a) Plot the Lagrange interpolating basis functions \( l_1^{(3)}(x) \) and \( l_3^{(3)}(x) \) \((0 < x < 1)\), generated using \( x_k = \{0, 0.2, 0.5, 1\} \). Use markers to highlight points \( x_k \) on the plots.

   (b) Plot the Lagrange interpolating polynomial \( p_3(x) \) \((0 < x < 1)\) that passes through the points \( (x_k, f(x_k)) \), \( f(x) = e^{-x} \). Evaluate the resultant polynomial at \( x = 0.7 \). Compare the answer to the result in problem 1.

3. **Finite Difference Tables:** Let \( S_{N}^{k} \) denote the sum of the \( k \) th powers of the first \( N \) integers i.e.:

   \[
   S_{N}^{k} = \sum_{i=1}^{N} i^k
   \]

   Write a simple MATLAB program to evaluate these sums for a specified value of \( k \) for values of \( N \) from 1...\( k + 3 \). Now write MATLAB code to form the forward difference table (since the sample points are uniform). Notice that for each value of \( k \) the difference table terminates - why does this happen? For the special case \( k = 2 \) extract the differences from your table and use the Gregory-Newton divided difference formula to derive the formula:

   \[
   S_{N}^{3} = \sum_{i=1}^{N} i^2 = \frac{1}{6} N (N + 1)(2N + 1)
   \]

4. **Newton’s Divided Difference Formula:** Write a MATLAB routine \( yi=ndiff(x,y,xi) \) to determine the Newton divided difference polynomial interpolant of a function \( y = f(x) \) whose values at the vector of sample points \( x \) are given in the vector \( y \), while \( xi \) is the vector at which the desired interpolated values are requested. Illustrate your results with the function \( f(x) = (1 + x^4) \exp(-x) \) on the interval \([-1, 1]\) with \( N = 4 \) uniformly distributed points \( x=-1:(2/(N-1)):1 \). You can check your divided difference code using the MATLAB function \texttt{polyfit}, which should
give precisely the same results for the interpolating polynomial as Nddiff. Provide the following values

\[
\begin{array}{c|c|c|c}
  x & -0.6 & 0 & 0.4 \\
  p_N(x) & & & \\
\end{array}
\]