The following assignments are due on April 9 at 12:30pm. No homework will be accepted after that time. You should give me your homework in class or put it under my office door (WWH612). Do not use my mailbox in the WWH lobby.

Turn in a listing of your MATLAB programs as well as output from selected runs. Always use format long e in MATLAB.

1. Problem 6.2 from the text book.

2. Problem 6.5 from the text book.


4. Consider the two different approximations of the first derivative of \( f(x) \):

\[
\frac{f(x + h) - f(x)}{h} \quad \text{and} \quad \frac{f(x + h) - f(x - h)}{2h}.
\]

Use MATLAB and these formulas for a variety of \( h > 0 \) to compute the derivative of \( \sin(x) \) at \( x = 1 \). Find the value of \( h \) which gives the most accurate results for the two formulas. Discuss your findings. Can you make an educated guess on properties of the arithmetic that is used by MATLAB and your computer?

5. Recall that the function

\[ f(x) = \frac{1}{1 + x^2}, \quad x \in [-5, 5] \]

is used when discussing Runge’s phenomenon; cf. pp. 186–187. Plot the function and its polynomial interpolant for 5, 11, and 21 equidistant points always including the two end points of the interval. The interpolation polynomial can be computed by using POLYFIT; try help POLYFIT in MATLAB. Note that POLYFIT provides least squares solutions but that the least squares solution is the same as the polynomial interpolant in a special case; see section 2.9 of the text book.
6. Revise your interpolation points and use

\[ x_k = 5 \cos \left( \frac{(2k - 1)\pi}{2n + 2} \right) \quad k = 1, \ldots, n + 1. \]

Use the same function \( f(x) \) as above. Discuss your findings.

7. Try to replace POLYFIT with your own MATLAB program implementing Newton’s algorithm; see description below. Provide evidence that your program is correct.

Newton’s interpolation formula.

The following formula, attributed to Newton, provides the unique polynomial \( p_n(x) \) which satisfies the condition \( p_n(x_i) = f(x_i), i = 0, 1, \ldots, n \):

\[ p_n(x) := f(x_0) + \sum_{i=0}^{n-1} f[x_0, \ldots, x_{i+1}] \prod_{j=0}^{i} (x - x_j). \]

The interpolation points \( x_i \) are distinct and \( f(x_i) \) are given values. The divided differences \( f[x_0, \ldots, x_{i+1}] \) are given recursively by

\[ f[x_0, x_1] := \frac{f(x_0) - f(x_1)}{x_0 - x_1}, \quad \text{and} \quad f[x_0, \ldots, x_{i+1}] := \frac{f[x_0, \ldots, x_i] - f[x_1, \ldots, x_{i+1}]}{x_0 - x_{i+1}}. \]

Note that the sum of the first \( k + 1 \) terms on the right of the formula for \( p_n(x) \) provides the degree \( k \) polynomial which interpolates the values of \( f(x) \) at the first \( k + 1 \) points.

Values of \( p_n(x) \) can conveniently be computed by noticing that

\[ p_3(x) = f(x_0) + (x-x_0)(f[x_0, x_1] + (x-x_1)(f[x_0, x_1, x_2] + (x-x_2)f[x_0, x_1, x_2, x_3])), \]

etc., i.e., by factoring out common factors.