Homework 3, Due: Wednesday, 11/14

This assignment is due on **Wednesday, November 14**, by 11:59 PM. Your assignment should be well-organized, typed (or neatly written and scanned) and saved as a .pdf for submission on Canvas. You must show all of your work to receive full credit. For problems requiring the use of MATLAB code, remember to also submit your .m-files on Canvas as a part of your completed assignment. Your code should be appropriately commented to receive full credit.

Problems

$\boxed{1}$ Consider the data generated by $f(x) = e^{-x}$ at the nodes $x_0 = 0$, $x_1 = 0.25$, $x_2 = 0.75$ and $x_3 = 1.0$.

(a) (6 points) Construct a natural cubic spline $S(x)$ to approximate $f(x)$ using the data at the nodes specified above.

(b) (4 points) Integrate the spline $S(x)$ over [0,1] and compare the result to $\int_0^1 e^{-x}\,dx = 1-1/e$.

(c) (6 points) Use the derivatives of the spline $S(x)$ to approximate $f'(0.5)$ and $f''(0.5)$, and compare your approximations to the actual values.

$\boxed{2}$ (6 points) Write a MATLAB **function** that computes Newton's divided-difference coefficients given a set of data. Call your function `newtonsDD.m`. Clearly define the inputs and output of your function, and explain what each portion of your code does. (It may help to refer to Algorithm 3.2 on pg. 124 of the text.)

Test your code by computing the divided-difference coefficients that correspond to the following data:

x	$f(x)$
1.0	0.7651977
1.3	0.6200860
1.6	0.4554022
1.9	0.2818186
2.2	0.1103623

Note that this is the same data used in Example 1 on pg. 124 of the text, so your results should match those in Table 3.11! To receive full credit, you must provide your function .m-file along with a script file demonstrating the results of the test problem. Use the results of your table to write the divided-difference form of the 4th Lagrange interpolating polynomial that interpolates the given data.

$\boxed{3}$ Consider the following data:

x_i	0.2	0.3	0.6	0.9	1.1	1.3	1.4	1.6
y_i	0.050446	0.098426	0.33277	0.72660	1.0972	1.5697	1.8487	2.5015

(a) (6 points) Construct the least squares polynomials of degree 1, 2, and 3 that best fit the above data, along with their corresponding errors. For each case, clearly show the set up of the least squares approximation: Write out the form of the model function $\widehat{y}_i$ and the corresponding design matrix X. You may use MATLAB to help you compute the least squares approximations – as usual, just make sure that you describe and include any code(s) that you use.

(b) (4 points) Create a figure in MATLAB plotting your three least squares polynomials from part (a), along with markers plotting the data. Make use of the `hold on` and `hold off` commands in MATLAB to plot all three polynomials and the data points in the same figure.

$\boxed{4}$ Using the same data as in Problem 3 above, construct the least squares polynomial approximation of the form be^{ax} by:

(a) (6 points) Considering the logarithm of the approximating equation $y = be^{ax}$, which implies that $\ln(y) = \ln(b) + ax$. This allows you to apply linear least squares to the log model to solve for the parameters $\ln(b)$ and a! You can then transform back to find an approximation for b.

(b) (6 points) **BONUS:** Applying a nonlinear least squares algorithm. One built-in MATLAB function commonly used for nonlinear minimization is `fminsearch`, which uses the Nelder-Mead simplex (direct search) method. Use this function to solve the least squares problem at hand. The key is in defining the objective function, which here should be the sum of squared errors. How do your least squares estimates for a and b compare to your estimates in part (a)? Which approximation gives you the better fit? Plot your model solutions with parameters from parts (a) and (b) and compare to the data.

Note: For any of the above problems for which you use MATLAB to help you solve, you must submit your code/.m-files as part of your work. Your code must run in order to receive full credit. If you include any plots, make sure that each has a title, axis labels, and readable font size, and include the final version of your plots as well as the code used to generate them.