CS412, Fall 19
Prof. Amos Ron

Assignment #3
Due: October 28, 2019
(after that day assignments should be put in the mbox of Jihye Choi)

(1) In this question you construct (without the use of code; you may use a calculator, though) a polynomial interpolant to data via the divided differences approach. You then evaluate your polynomial and its first derivative using nested multiplication.

(a) Add four data points of your own choice to the given point (k,3), with k the last digit of your UW i.d.: the x-value of each of the points you add should be *negative.* Construct a divided difference table corresponding to these five points. Using the table, find the polynomial in Π_4 that interpolates your data. Write that polynomial in two different Newton forms. (Read the notes in order to find "the other form": we also covered it in class).

(b) Using nested multiplication evaluate each of the above forms at $x = 2$ and $x = 0$. Now, use nested multiplication in order to evaluate the derivative of each form at $x = 2$ (note that you are double checking yourself, since the two forms represent the same polynomial).

(c) Now replace one of your points by the point (5,6), and find the new polynomial interpolant. You are free to choose which point to delete, but you want to be efficient here, hence you wish to use the output of (a) to the extent that this is possible. Choose the point to be deleted in a way that minimizes the additional computations, and find the new polynomial that interpolates the new dataset.

(2) In this question you explore the error formula of polynomial interpolation. You will try to predict, based on this formula, whether polynomial interpolation provides a good fit to a given function f on the interval $[-1, 2]$. No code is involved in this question.

(a) You are given four functions, all defined on the interval $[-2, 1]$: $f(x) = e^x$, $g(x) = \sin x$, $h(x) = \log(x + 6)$, $k(x) = |x|^{3/2}$. Find formulæ that express the nth order derivatives of each of these functions. (If you cannot find a forumla that is valid for all n, find the derivatives for $n = 1, 2, 3, 4$). Check carefully the smoothness of each of these functions on the interval (i.e., whether a derivative of some order of the function of interest fails to exist at some point(s)).

(b) Try to bound the nth order derivative of each of your functions on the interval $[-1, 2]$ (e.g., if $m(x) = x^5$, and $n = 2$, then $m''(x) = 20x^3$, and on the interval $[-1, 2]$ this derivative is bounded (in absolute value) by 160. Note that your bound may depend on n). Divide the bound that you have by $n!$ (as in the error formula).

(c) Your new boss, I.F. Ail, plans to approximate each of the functions in (a) by interpolating it at some points (say, uniformly spaced) on the interval $[-1, 2]$. She plans to pick one of these functions, interpolate it at 3 points to begin with, and then gradually increase the number of interpolation points, with the hope that eventually the interpolant will converge to the original function. She solicits your opinion concerning the possible success of that process. Based only on the analysis you did in (b), what is your prediction concerning the hoped-for convergence? Your prediction can be one of the following three: 'convergence is quite certain', 'convergence is unlikely, or may be very slow', 'it is hard to tell' (try to avoid 'hard to tell'; Frida does not like it). Give some reasoning. Of course, there are four functions here, hence four predictions.

(3) In this question you compare numerically polynomial interpolation at equally spaced points, polynomial interpolation at the Chebyshev points, and cubic spline interpolation.

(a) First write a routine that does the following: given an interval $[a, b]$, a function F, and a number n, your code should produce the following three interpolants to the function F: (i) a polynomial interpolant to F at n equidistant points on $[a, b]$; (ii) a polynomial interpolant at n Chebyshev points of that interval; (iii) a cubic spline interpolant at n equidistant points. You are allowed to look at our class demo subdir, but you must document this fact (and you may not use the `matlab` library, such as `polyfit`). Furthermore, the polynomial interpolation must be done via a table of divided differences.

In order to produce the spline interpolant, you may use the `matlab` library. With x the set of your interpolation points, and y the values of F at these points, you may use the sequence

```
s=spline(x,y);
val=ppval(s,z);
```

The vector *val* then contains the values of your spline interpolant at the points specified in z. This is equivalent to the single command

```
val=spline(x,y,z);
```

(b) Now, run your code with different values of $n \leq 25$, with respect to each of the four functions that appear in Q. 3. Try to find values of n that allow you to make observations (of the sort: 'the spline interpolant is much worse than each of the polynomial interpolants', or 'There is a major improvement by switching to Chebyshev points', etc.). For each function, select two values of n for which you are able to make the most significant observations.

(c) Turn in your well-documented code, the eight outputs (two values of n for each of the four functions), and the observations you were able to make. Compare these observations with your predictions in the previous question. Here, 'output' must include a plot of the error functions, and some (2-3) of the divided difference tables that your code produced.

In contrast with previous assignments, we will examine in this question the quality of your code. Namely, writing a compact efficient code is desired.