CS3414. Homework Project set IV. Each problem is worth 10 points unless otherwise specified.

C&K = *Cheney and Kincaid* textbook. You can write your codes in C or C++ or Mathematica.

1 Problem 1.

From census data, the approximate population of the United States was 150.7 million in 1950, 179.3 million in 1960, 203.3 million in 1970, 226.5 million in 1980, 249.6 million in 1990, 281.4 million in 2000, and 309.3 million in 2010. Using an exact interpolation polynomial for these data points (see `interpol_pol1.nb`), find an approximate value for the population in 1985. Show the polynomial and the plot, including the orginal data points. Then use the *same* polynomial to estimate (*extrapolate*) the population in 2015 and 1920. Check these numbers against real data found on the web. What conclusion should be drawn?

2 Problem 2.

Approximate $(f(x) = arcsin(x)$ on the interval $-1/2, 1/2$ by an interpolating polynomial of degree 15, $P_{15}(x)$. Use equally spaced nodes. How many? Show the plot of the original function and the interpolation. Now determine how accurate the approximation is by numerical tests; you want to find maximum error, that is max. of $|f(x) - P_{15}(x)|$ on the entire interval, discretized to $h = 10^{-3}$

3 Problem 3.

Use `spline.math` discussed in class to calculate the 2^{nd} derivative S_3'' of the interpolating cubic-spline function, S_3. Using S_3'', figure out if "Mathematica" uses natural splines with `Cubic` option? If you are unable to beat *Mathematica*'s syntax and make it produce second derivatives within the code, then you can calculate them by computing appropriately close values of S_3 and generating the derivatives externally, see C&K, page 196 Eq. (20). For example, `Print[fit[0]]` will print the value of the spline function at $t = 0$, that is at the leftmost knot; `fit[0.01]` is 0.01 away, to the right. See `spline.math` for these examples. You need to show the equations used, the values of h, and the values of the second derivatives used to draw the conclusion.

4 Problem 4 (30 points).

Write your first or last name (at least three letters) on a piece of graphing paper. Use "continuous" cursive in which the pen never leaves the paper. You may find it useful to magnify the original, using a Xerox machine or a scanner. Place knots (for interpolation) on the name trace, at least five per letter. Read off (x,y) coordinates of the knots, and use them to prepare a natural cubic spline (parametric) approximation of your name. You can use *Mathematica*, a library routine, or your own code, if you like. Compare the result with the original **on the same scale**. Did you have enough knots? May need to add a few more to make the interpolation visually close to the original. Present both the original and your best interpolation in the report. What compression ratio of the input data have you achieved? Assume that the "input" is a high-quality `JPEG` image of your signature, image size 1" x 2" or so, and for the "compressed" data you need enough memory to to store all of the coefficients of the polynomials in the spline function you have just construced.