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. C&K 4, page 162.
2. C&K 11, page 163.
3. C&K 11, page 169.
4. C&K 5, page 177. (extra credit).
5. C&K 5, page 179.
6. C&K 3, page 413.
7. C&K 14, page 415.

8 (20 points). Modify `spline.math` discussed in class to calculate the 2^{nd} derivative S_3'' of the interpolating cubic-spline function, S_3. Plot the original function $f(x) = 1/(1+x^2)$, its interpolation S_3, and its S_3''. Using S_3'', figure out if "Mathematica" uses natural splines with `Cubic` option? Include the plots and a printout of the modified `spline.math`. If you are unable to beat *Mathematica*'s syntax and 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 190 Eq. (20). For example, `Print[fit[0]]` will print the value of the spline function at $t = 0$, that is at the leftmost knot; `Print[fit[0.01]]` is 0.01 away, to the right. See `spline.math` for these examples.

9 (30 points). Write your first or last name (at least three letters) on a piece of graphing paper. Use "continuous" script 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 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.