Math 448 – Numerical Analysis: Final Spring 2019

It would be great if you can finish this by the end of exam week, since grades are due early the next week. Submission can be a combination of handwritten notes, and printed Matlab code and pictures – either separately or copied into one big Word file.

1. The golden section method to minimize a function f on an interval $[a, b]$ is:
   Let $\phi = (1 + \sqrt{5})/2$, $c = b - (b - a)/\phi$, $d = a + (b - a)/\phi$, $fc = f(c)$, and$fd = f(d)$.
   While $b - a > tol$,
   if $fc < fd$, $b = d$, $d = c$, $fd = fc$, $c = b - (b - a)/\phi$, $fc = f(c)$
   else $a = c$, $c = d$, $fc = fd$, $d = a + (b - a)/\phi$, $fd = f(d)$
   end
   end

   (a) Write Matlab code that implements this method, with input a function name, a, and b. You should have a while loop that exits when the $b - a < 10^{-10}$. Make your version output the current interval at the end of the loop.

   (b) Apply this function to $f(x) = x^3 - 4x^2 + 2x - 1$ on $[1, 4]$. Verify that you have the correct solution by finding the minimum analytically using calculus. Yes, Calculus I is still, and always will be, useful!

2. Consider the system of equations $1 + x - y^2 = 0$ and $y - x^2 = 0$.

   (a) Rewrite the second as $y = x^2$, substitute into the second and show $x^4 - x - 1 = 0$. Find the four solutions using Aberth's method and Durand-Kerner's method. Verify that you have the same solutions that come from Matlab's `roots` function.

   (b) Set up this system of equations for solution using Newton's method, including the system of equations in your solution. As in class, solve with various initial guesses and verify you get the same four solutions as above. You will need complex initial guesses in some cases.

   (c) Explain why solving this system of equations is equivalent to minimizing $f(x, y) = (1 + x - y)^2 + (y - x^2)^2$.

   (d) Use Matlab's `fminsearch` to minimize $f(x, y) = (1+x-y)^2+(y-x^2)^2$ and verify you can find the same real solutions as above with appropriate initial guesses. Before you run the command, type `figure, hold on`, then include `plot(x,y,'k')` in the function so you get a picture of where the function evaluations are placed. What do you think is happening?

   (e) As in class, use the steepest descent code to minimize this function step by step. How well does it do?

3. Find linear, quadratic and cubic least squares approximations for the following data, and their sums of squares of errors S. I recommend using the theory from class to build a linear system of equations, then solve using Matlab's backslash command. Include in your solution a plot of the original data, and the various least squares approximations. Do you think the approximations are any good?

x	$f(x)$	x	$f(x)$	x	$f(x)$	x	$f(x)$
0.1	3.0801	5.2	13.3207	9.4	14.6359	15.6	8.5326
1.1	5.1347	6.1	16.7352	11.1	12.2725	16.1	9.4929
1.6	6.6853	6.6	15.5935	11.4	13.1354	17.6	7.4184
2.4	7.2045	7.1	15.8526	12.2	12.3529	17.9	4.5196
2.5	7.8503	8.2	16.2694	13.2	10.5963	19.1	4.8213
4.1	10.7652	9.1	13.9964	14.1	7.8457	20.0	3.3183

4. Consider the *piecewise* polynomial fit

$$y = \begin{cases} c_1 + c_2 x, & x \in [a, \xi], \\ c_3 + c_4 x, & x \in (\xi, b]. \end{cases}$$

The parameter ξ must be chosen in advance. If the $\{x_i\}$ are ordered, then j is chosen so that $x_j \le \xi$ and $x_{j+1} > \xi$. The least squares fit then minimizes

$$S = \sum_{i=1}^{j} (y_i - c_1 - c_2 x_i)^2 + \sum_{i=j+1}^{n} (y_i - c_3 - c_4 x_i)^2.$$

To ensure continuity at ξ, find c_3 as a function of c_1, c_2, c_4 and ξ, and substitute it in the expression for S. By setting $\partial S / \partial c_i = 0$, $i = 1, 2, 4$, show that the least squares piecewise linear fit to data is the solution of $A\mathbf{c} = \mathbf{r}$ where

$$A = \begin{pmatrix} n & \sum_{i=1}^{j} x_i + (n-j)\xi & \sum_{i=j+1}^{n} x_i - (n-j)\xi \\ \sum_{i=1}^{j} x_i + (n-j)\xi & \sum_{i=1}^{j} x_i^2 + (n-j)\xi^2 & \xi \sum_{i=j+1}^{n} (x_i - \xi) \\ \sum_{i=j+1}^{n} (\xi - x_i) & \xi \sum_{i=j+1}^{n} (\xi - x_i) & -\sum_{i=j+1}^{n} (x_i - \xi)^2 \end{pmatrix},$$

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_4 \end{pmatrix}, \quad \text{and} \quad \mathbf{r} = \begin{pmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{j} x_i y_i + \xi \sum_{i=j+1}^{n} y_i \\ \sum_{i=j+1}^{n} y_i (\xi - x_i). \end{pmatrix}.$$

For the data in question 3, find the piecewise linear fits with $\xi = x_i$, $i = 2, 3, \ldots, 23$, and choose the one that minimizes S. Plot this with the original data. I recommend writing a function that takes as input the x and y vectors as well as ξ, then calculates the various sums, sets up the system of equations, solves using backslash, then outputs a vector of $\mathbf{c}$ as well as S. Then put this in a loop that tests all the different cases, and see what the best S is.