CBE 20258 Algorithm Problem Set 3

Due in class 2/18/16

Problem 1). In the first project, you calculated the flow rates for a set of seven streams in a cell separator as a function of operating pressures, and then used the information to determine the allowable operating ranges. As you probably noticed, it actually took a few moments for the code to run, even if you had a reasonably efficient algorithm. This can cause some real problems if you ask slightly different questions!

Suppose your boss says, OK – we know the operating ranges for a particular combination of channel lengths. How does the operating region change if we modify the various values of each length L_i^*? If we do a combinatoric study for 10 values of each of 7 lengths, that would involve resolving the code a whopping 10^7 times: it would take about a year to execute! You could do much better by recognizing that, without loss of generality, you can take L_1^* to be fixed at unity, and then use symmetry to ignore variations in the mirror image streams (this kills off another 3 variables). Still, it takes 10^3 combinations to explore even this reduced set: meaning about an hour of execution time! Thus, it pays to use numerical analysis to create a much more elegant solution to the problem!

a. The problem you were investigating involved the solution of the expression $\underset{\approx}{A}\,\underset{\sim}{Q} = \underset{\sim}{b}$, where $\underset{\sim}{b}$ was the column vector:

$$\underset{\sim}{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ p_1^* \\ 1 \\ 1 \\ p_2^* \end{bmatrix}$$

Explicitly obtain the expression for the flow rate Q_i in terms of p_1^* and p_2^* and the elements of A^{-1}. (Note: no numbers or values of L_i^*, just leave it in the form $\left(\underset{\approx}{A}^{-1}\right)_{ij}$, etc., where the indices are identified)

b. These expressions are all hyper-planes (e.g., a particular Q_i is a linear function of the two variables p_1^* and p_2^*). The valid and invalid regions in p_1^* and p_2^* space are bounded by lines where the various Q_i are zero. Develop an analytic expression for each of these lines (e.g., where $Q_i = 0$) in terms of the elements of A^{-1}.

c. The vertices of the valid regions are just the intersection points of these lines (where $Q_i = Q_j = 0$). Develop an expression for the intersection point of the ith and jth line. There would be a total of 7*6=42 of these, but leave it in i,j form (single expression).

You could go back to your graph for a single test case and figure out exactly *which* of these intersections matter to get the two little "valid triangles" you highlighted in your project, but I'm not asking you to do that here. If you had to figure out, say, how the area of allowable operating pressures (the area of the triangles) depended on the channel segment lengths, this is the way you would approach it!

Problem 2). On the first algorithm assignment you determined the optimum depth of the second probe in a two-probe system for calculating the temperature gradient at the surface. Suppose your boss says that this isn't good enough, and suggests adding in a third probe (e.g., one at a depth h and the second at a depth 2h). Take the expected temperature profile to be about $T_0 + A*\exp(b*x)$ where $A = 2°C$ and $b = 0.5\ cm^{-1}$. The probes have a random error of ±0.1°C.

a. Redo the calculations for the two probe system, this time calculating the random experimental error using the correct error propagation formula used in class and the expected temperature distribution given above. What is the optimum value of h, and what is the best we can do calculating the derivative?

b. For your three probe system you will not want to weight the probes evenly. Determine the optimum weighting of the three probes for getting the surface derivative (Hint: you need to keep another term in your Taylor series expansion for the algorithm error, and try to get the quadratic term exactly!)

c. Determine the random experimental error for this new three probe formula as a function of h, and determine both the optimum h and the minimum derivative measurement error.

Problem 3). You are using a falling-ball rheometer to measure the viscosity of a liquid. The fluid is sufficiently viscous that the Stokes sedimentation equation described in class is assumed to apply (you will learn all about that next term). The experiment consists of measuring how long (t) it takes a sphere of radius (a) to fall a distance (L). The sphere density is (ρ_s) and the fluid density is (ρ_f). The velocity $U_s = L/t$. We have the measurements (regarded as independent):

$$U_s = \frac{2}{9}\frac{\left(\rho_s - \rho_f\right) g a^2}{\mu}$$

	value	1σ
a	$100\mu m$	$1\mu m$
ρ_s	$1.18\, g/cm^3$	$0.01 g/cm^3$
ρ_f	$0.98\, g/cm^3$	$0.002\, g/cm^3$
g	$980.6 cm/s^2$	0
L	$2 cm$	$0.01 cm$
t	$230\, s$	$0.5\, s$

a. What is the 2σ error bound for the measured viscosity (μ)? Report this in units of g/cm-s.

b. Which measurement had the largest contribution to the error in the final result (e.g., which measurement would you want to spend some time and money improving)?

Hint: It is much easier if you do this one with variables using the computer. I would write a short function which takes in all the parameters as a vector and spits out the viscosity. You can then take the derivative using the finite difference method as discussed in class and calculate the error. You can also see which term has the largest contribution to the overall error (e.g., part b). You can do it all using pencil and paper, of course: your choice.

Problem 4). In the last project you found that the average Notre Dame high temperature excursion last year was 0.937 degrees above the historic averages, and the standard deviation was 9.938 degrees.

a. If we assume that all the days are independent, what is the probability of the average excursion being this large or higher?

b. What is wrong with the answer to part a, and about what should the correct value be? State your (new) assumptions.

c. For an extra credit point, get a precise value for part b using the results of your project. This will require a few lines of matlab code.