CBE 20258 Algorithm Problem Set 1

Due in class 1/26/17

Problem 1). Setting up Systems of Equations in Matrix Form:

a. Suppose you have a specified number of cans of red, blue, and yellow paint. You don't like these colors, however, and want to combine them to form cans of purple, orange, and green paint (which are composed of even mixtures of the appropriate primary colors). If the number of cans of the three primary colors is given by a 3x1 column vector $\underset{\sim}{b}$ and the number of cans of the new colors is given by a 3x1 column solution vector $\underset{\sim}{x}$, set up the problem in the form $\underset{\approx}{A}\,\underset{\sim}{x} = \underset{\sim}{b}$, clearly identifying $\underset{\approx}{A}$, $\underset{\sim}{x}$, and $\underset{\sim}{b}$.

b. If you had 3 cans of Red, 1 cans of Blue, and 4 cans of Yellow, how many of each of the new colors can you make? You can solve this easily by hand, or use the computer if you wish.

c. What is wrong with your solution if you had 5 cans of Yellow?

Problem 2). Algorithm vs. Numerical Error: You are trying to design a system to measure the temperature gradient of a slab of material normal to its surface (e.g., the normal derivative of the temperature evaluated at x = 0). You have one probe at the surface itself (x = 0), and can place a second at a depth h (x = h). The probes only measure the temperature to a precision of ± 0.1°C, the first derivative is expected to be of O(1 °C/cm), and the second derivative should be of O(2 °C/cm^2). About how deep should you place the second probe? What is the expected precision of your measured gradient? Show your work and assumptions!

Problem 3). Overflow and Algorithm Issues:

a. Over the past couple of years the lottery has reached amazing heights, in part due to the decrease in the individual odds of any set of numbers winning from 1/175M to 1/292.2M done in the fall of '15 (actually a great marketing move…). The all-time record payout drawing was on 1/13/16 and was worth \$1.586B (lump sum was \$983M, and after tax value was roughly \$590M). About \$1B in tickets were sold at \$2 each. Calculate the probability that there would be exactly 0, 1, 2, 3, and 4 winners of the jackpot. Assume that everyone uses the "computer option" to pick their numbers at random. Hint: Although you could use the example program from class (lecture 2), it takes a little while to run and is very inefficient for such large numbers! You can do it with a calculator if you recognize that for small k and large n, (n-k)! ~ n!/n^k. The "1-p" exponentials are dealt with using the first term of the Taylor Series: $\ln(1-p) \approx -p$ for small p.

b. What is the expected return (after taxes) of buying just one of the 500 million tickets sold? Note that your return goes down if more than one of the tickets "hits", as was the case last January.

c. Interestingly, research has shown that if people pick their own numbers rather than letting the computer do it, the choices are highly non-random. In particular, people usually pick numbers from 0-31 (corresponding to birthdays) while the lottery numbers go from 0-69. In view of this, *qualitatively* answer the following questions:

1) Does the probability of any individual's lottery ticket winning change?

2) Does the probability of getting multiple winners change (e.g., do you expect the payout to a winning ticket decrease due to splitting the prize)?

3) Does the probability of no one winning the lottery change?

4) For financial reasons, why are you better off letting the computer pick the numbers than picking birthdays?

5) Can you think of an even better strategy which would lead to a higher expected return (it's still a bad investment, though…)?

Problem 4) Ill-Conditioned Equations: A classic example of ill-conditioned equations occurs in acid-base equilibria. Suppose you are diluting 60g of acetic acid (CH_3COOH) with water to make up a 1 liter solution. The goal is to determine the pH (the negative of the log10 of the hydrogen ion concentration [H+]) of the solution. There are four species present: [H+], [OH-], [CH_3COOH], and [CH_3COO-]. These are governed by the four balance equations:

1) Mass balance: [CH_3COOH] + [CH_3COO-] = Total Molarity of Acetic Acid

2) Ion balance (electroneutrality): [H+] = [OH-] + [CH_3COO-]

3) Water Equilibria: [H+]*[OH-] = Kw = 10^{-14} M^2

4) Acid Dissociation: [H+]*[CH_3COO-]/[CH_3COOH] = Ka = 1.76×10^{-5} M

It is possible to just rearrange these (non-linear) equations so that the right hand side of each is zero, and then solve it as a system of non-linear equations $\underset{\sim}{f}\left(\underset{\sim}{x}\right)=0$. Matlab is very good at finding the roots to systems of equations, and the command we will use later this term to do this is "fsolve". If you do this, however, you will find that you often get weird and incorrect answers, including negative concentrations of some of the ions! This is because the problem as written is ill-conditioned (variables and equations differ widely in magnitude). So:

Recast these equations so that they are well-conditioned (all variables $\underset{\sim}{x}$ and equations $\underset{\sim}{f}$ are of comparable magnitude).

For an extra credit point, solve for the pH using Matlab… Hint: Try playing around with taking the log of the equilibrium equations and working with the log of the concentrations…

What you should get from these questions:

Pretty much all of the algorithm questions asked in this course are designed to reveal (or reinforce) ways of getting a “word problem” to a form you can actually plug numbers into. For many of you (OK, most of you) this is a lot harder than actually coding things up and solving them on the computer. Remember, people were solving problems with slide rules not too long ago, and using just pencil and paper not long before that! The ability to translate words into math is critical to you as an engineer…

1. This is a pretty straightforward example of setting up a system of equations, where the (usual) error is getting the variables mixed up. The key is to do the mass balance correctly and -then- put it down in matrix form. In most engineering problems, recasting the words into equations is the hardest part!

2. This is a problem combining both algorithm errors and random errors. Usually you want the non-random algorithm errors to be of the same order or smaller than the random errors (which could be zero, after all – you know the algorithm errors won’t be!). It is also a pretty basic problem in experimental design: how do you design a system to minimize overall error. Later this semester (after we’ve done lots of error propagation) you will learn how to determine which experimental measurements should get the most attention (e.g., $ spent on improvement) by doing a sensitivity analysis.

3. This is a fun probability analysis problem, demonstrating both overflow/underflow issues, and more importantly the use of approximate analytic expressions (e.g., Stirling’s Formula) in asymptotic limits. We’ll use this sort of approximation quite a bit next semester in transport to make problems more tractable.

This problem also illustrates why the lottery is a bad investment even when other people (by already having lost) have pumped up the payout to ridiculous levels.

4. Conditioning equations is very important when solving them numerically, as (due to round off errors) even tiny errors can totally screw up the answer. There are an infinite number of ways to condition equations, but the key is to try to work with variables which are always O(1) in magnitude, equations are O(1) in magnitude, and “non-physical solutions” are eliminated. In this problem, that means concentrations can’t be negative (note that Matlab wouldn’t care, and can’t tell the difference between a concentration just above zero and just below – but you can!).

Hint: Try working with the log (or log10) of the concentrations and some of the massaged equations.