# Applied Calculus

## Homework 7

## Due in class, November 24, 2015

1. Suppose $y = a \ln(bx)$. Show that the error in this expression propagates according to

$$\Delta y \approx a \frac{\Delta x}{x} .$$

2. (The context is Example 19.12 in MLS on radio-carbon dating, which I repeat here in parts.) All living things contain small amounts of radioactive carbon $C^{14}$. The ratio of $C^{14}$ to stable $C^{12}$ is constant in the atmosphere. Once death occurs, $C^{14}$ is no longer taken in, and the amount of $C^{14}$ in the organism begins to decay according to

$$A(t) = A_0 \, \mathrm{e}^{-kt} ,$$

where $A(t)$ is the activity at time $t$ and $A_0$ is the initial activity.

   (a) The half life of $C^{14}$ is 5760 years. Find $k$.

   (b) You have a sample with carbon ratio $A(t)/A_0 = 0.1$. How old is the sample?

   (c) What is the uncertainty in the age of the sample if you can determine the carbon ratio $A(t)/A_0$ only with a relative accuracy of 10%, assuming that the value for $k$ is exact?

3. If you measure $x = 10 \pm 1$ and $y = 15 \pm 2$, what should you report for

$$z = \frac{xy}{y - x}$$

together with its uncertainty?

4. Suppose $u = x\,y\,z$. Find an expression for the *relative* uncertainty of $u$ in terms of the *relative* uncertainties of $x$, $y$, and $z$. (Cf. example from class on Thursday!)

5. Find $f(x)$ for each of the following given $f'(x)$.

   (a) $f'(x) = x$

   (b) $f'(x) = x^n$ with $n \neq -1$

(c) $f'(x) = x^{-1}$

(d) $f'(x) = \mathrm{e}^{rx}$

(e) $f'(x) = (1+x)^2$.

(2 points each)

6. Use Scientific Python to verify the conclusion of Problem 4 computationally.

   A commonly used technique to estimate the propagation of error (and other random processes) in complex models is the method of *Monte-Carlo simulation*. The idea is the following: You run the computation many times with randomly perturbed input values, and look at the statistics of the output. The advantage is that you can treat the model as a black box, the disadvantage is that you need a large number of computations to get good statistics, which might be expensive when the model is complex. This exercise demonstrates the idea behind the Monte-Carlo technique in a simple setting.

   In Scientific Python, you can create arrays (ordered sets of numbers) and perform computations with each of its members "simultaneously" just as if it were a single computation. Proceed as follows.

   - `normal(m,s,N)` will produce an array of length `N` of random numbers which are drawn from a normal distribution centered at `m` with standard deviation `s`.

     Thus, to test the propagation of uncertainty for a quantity specified as $x = 7.3 \pm 0.01$, you would create a sample via `x=normal(7.3,0.01,10000)` where, in this example, we have chosen a sample size of 10 000.

     You can proceed likewise for the other variables $y$ and $z$.

   - You can now compute $u = x\,y\,z$ for all samples at once via `u=x*y*z`. The resulting variable `u` is still an array with 10 000 components!

   - The correct concept for uncertainty in statistical terms is the standard deviation. Thus, the relative uncertainty of $x$ expressed in Scientific Python is `std(x)/mean(x)`. Corresponding expressions hold for the relative uncertainties of $y$, $z$, and $u$.

   - Now compute the relative uncertainty of $u$ in two different ways. First, use your formula from Problem 4. Second, compute the relative uncertainty directly from your array `u`. Are they close? Comment on any differences.

   - Finally, destroy the independence of the errors in $x$, $y$, and $z$ by using the same array `x` for all three. What do you get? Explain!

   If you are ambitious, you may also try the following:

   - Instead of normally distributed random numbers, which is the most natural assumption in the absence of information to the contrary, you may take random number that follow different distributions. For example, `rand` produces uniformly

distributed random numbers, which you can scale appropriately and use in a similar way. You will find that the conclusions above do not change—standard deviations and independence are all that matters.

- Test the conclusion of Problem 3. Here, the relative errors are quite large and there is a potential problem with small denominators in this expression, so you would expect that the Monte-Carlo approach might differ significantly from the estimate given by the the error propagation formula. The the latter is based on the linear approximation of the function and may not always be trusted.