Math 56 Compu & Expt Math, Spring 2013: HW2 debrief

April 14, 2013

Please indicate who you work together on homework. Remember the write-up should be in your own words (ie own LaTeX). Careful explaining: "reduce the accuracy" is opposite sense from "reduce the error"—try to be unambiguous.

1. 2+2+3+2+1 = 10 pts Some of you lost points for forgetting that we want *relative* error throughout this question (after all, the naive implementation already has small absolute error)!

   (a) You didn't need full ε analysis here, just to realise how rounded down. Classic catastrophic cancellation.

   (b) Please if you use wikipedia and find Newton's generalized binomial, which is one name for what you're using, then explain *what* the symbols you use mean, eg $\begin{pmatrix} \alpha \\ k \end{pmatrix}$, $(-r)^k$, etc. Best not to use these fancy symbols at all. `Sage` can help you check your answer.

   (c) See Kyutae's detailed analysis. However, if you explained which were dominant (the addition and the sqrt caused the ε which end up dominating), fine too. It you didn't include one of those epsilons, you were underestimating the error by a factor 3/2, so lost a point.

   (d) Hanh had a nice argument to deduce the number of terms.

   (e) Must quote relative difference (abs diff is misleadingly small). Eg see Kunyi.

2. 3+2+4 = 9 pts

   (a) See eg Nate.

   (b) geometric understanding is key, eg see Tom.

   (c) Effort for coding here. Don't forget to do enough iterations that all pts in the grid converge—this is at least 10 even at low resolution. Also, your code is faster if you *vectorize*, i.e. carry out the Newton step formula on all points in the grid simultaneously. Eg if f and f' are defined funcs that accept vector or array inputs:

   ```
   u = linspace(-2,2,1000);
   [U V] = meshgrid(u,u); Z = U + 1i*V;
   for i=1:20, Z = Z - f(Z)./fp(Z); end
   imagesc(u,u, angle(Z));
   ```

   A neat way to convert the complex z values into a real number indicating which root they were nearest, was to use `angle`, eg see Kyutae who did some nice zooms.

3. 2+2+3 = 7 pts. See my code `exptaylor.m`. You can also use `polyval` to evaluate polynomials (eg Taylor series) as Hanh shows. Also check out her beautiful plots showing convergence over a range of x, and the "repeated-squaring" algorithm to bring x close to zero so a short Taylor series is accurate.

   (a) See Tom's nice explanation. Note you needed Taylor's *theorem* with a bound on e^q for unknown point $q \in (0, x)$. When I ask for a "bound" I usually mean upper bound, as here. Whether you stopped at x^{49} or x^{50} is ok, since saying 50 terms is ambiguous about whether to include the 0th term. In former case, relative error is bounded by `1/factorial(50)*10^50` which you should evaluate to 3.3×10^{-15}.

(b) As John explains, rel err is $O(1)$ no matter how high you take n.

(c) Condition # is $|x|$, see eg Kunyi, so it's an algorithm fault not an ill-conditioned problem. Really you should check that it's not a problem that summing from small to large can cure. But, what is the largest term? It's around $k = 20$, of size 10^7, so roundoff explains why we have $O(1)$ rel err on an answer size 10^{-9}. Catastrophic cancellation. Only Tom explained this correctly.

4. 2+2+2+2 = 8 pts

   (a) Eg see John.

   (b) Surprisingly, κ only large for $x \approx 1$, since as the graph crosses zero, relative errors are v sensitive to x. Large and small x are fine.

   (c) As Kunyi showed, it's best to simplify the formula for roots vs c. Then you think of this as your $f(c)$ to put in the κ formula. So, finding nearby roots of a polynomial is ill-conditioned, and κ is about the inverse of the root separation (generally true).

   (d) Eg see Ben. This is really now best seen as an application of the error theorem from lecture 6.

5. 4 pts. Beautiful example of transition from Taylor-series (ie 2nd-order) dominated error to catastrophic cancellation dominated error. Make sure you know how to predict the best h (around 10^{-5}) and the error it gives (around 10 digits). See Kyutae's nice addition of the predicted lines to the loglog plot—amazing how well they fit.

6. 3+2+3 = 8 pts

   (a) Lesson: never compare floating-point numbers to see if they are equal! (possible exception is if they haven't been changed from being set to a definite number like zero).

   Maybe suggestions to fix, but this most accurate is to make integer loop and compute x from it:

   ```
   for i=0:10, x = i/10; end
   ```

   Incrementing x repeatedly would accumulate more round-off errors.

   (b) Note "error" here just means difference between left and right sides. It's very close! (according to wikipedia, this example of a near-Fermat-breaker was constructed by the show's writer David Cohen!)

   (c) See John's detailed flop count. All got this; basic idea is $O(n^2)$ vs $O(n^3)$.