Math 56 Compu & Expt Math, Spring 2013: Homework 3 Debrief

April 22, 2013

Please study this, and all the debriefings. I consider them part of the required reading for the course. You will learn a lot, especially if you redo the HW you lost points on.

1. 3+3+2 = 8 pts
   (a) The 3 rows of the linear system are the conditions that: eliminates the coefficient of $f(x)$, equates the coefficient of $f'(x)$ to unity, and eliminates the coefficient of $f''(x)$. The "stencil" is solved to be $[-3\ 4\ \ -1]/2h$
   (b) It's best to test your formula to check it, e.g. like this:
   `f=@(x)sin(x); fp=@(x)cos(x); h=1e-5; x=1; (-3*f(x)+4*f(x+h)-f(x+2*h))/2/h - fp(x)`
   This gives error of about `3e-11`. Note I chose around the best $h = O(\varepsilon_{\text{mach}}^{1/3})$
   (c) Taylor's theorem with bounds on the $f'''(q)$ terms. Most of you had problems staying rigorous here, forgetting the absolute value signs. It was ok if you presented it in $O(h^2)$ form if you showed why the bound involving a sum of two f''' values tends to a constant (basically, you need f''' continuous). See Hanh's solution.

2. 3+3 = 6 pts. *Note: a style tweak: the word "flop" or "flops" is only used when counting numbers of floating-point operations, not when discussing rounding error. I don't know why; it's just numerical culture.*
   (a) This was as in lecture (there we did $x_1 + x_2$).
   (b) This is hard (good practise understanding the concept) but simple once you see it the right way. Eg John. If you wrote out the whole relative error expression it was hard to see how to choose the two epsilons to make it work, but you can.

3. 2+3+3 = 8 pts
   (a) The eigvals of A are complex conjugate pair (typical of mostly rotation matrices) of equal magnitude, 2, which must be less than $\|A\|$ (why?)
   (b) You notice a shortcut that the smallest eigenvalue of $A^T A$ tells you the reciprocal of $\|A^{-1}\|$. Thus $\kappa(A)$ is the sqrt of ratio of largest to smallest eigenvalue of $A^T A$.
   (c) See anyone's picture: the ellipse longest semi-axis is $\|A\|$, and shortest is $1/\|A^{-1}\|$ (why? max growth under A^{-1} must be min growth under A), so $\kappa(A)$ is the ratio of major to minor axes.

4. 3+3+3+2 = 11 pts. The point of this question is really that $\kappa(A)$ controls (via the Backwards Stability Theorem) only the *worst-case* errors, not even the typical error.
   (a) Shows matrices with same norm may have wildly different $\kappa(A)$. In fact I chose `A1` to have *singular values* `10.^(-(0:99)/7)` which makes the ratio of largest to smallest 1.39×10^{14}.
   (b) Here `bvec` was chosen to be unit magnitude, in the direction of `A1`'s largest ellipse semi-axis (growth output direction). However, perturbations of $\mathbf{b}$ are amplified by the reciprocal of the smallest growth factor of A, which was the reciprocal of the tiny 10^{-14}. Thus relative errors are huge. Consistent with theorem.

(c) Isn't it fascinating that now the relative error due to random perturbations of $\mathbf{b}$ is only around 10^{-14}, and that moreover this is typical of random RHS vectors? This is subtle: a random RHS vector $\mathbf{b}$ has $A^{-1}\mathbf{b}$ very large, $O(\|A^{-1}\|)$, so absolute errors in $\mathbf{x}$ are relatively small.

However, if instead it were the $\mathbf{x}$ that were chosen randomly, and used to generate $\mathbf{b} = A\mathbf{x}$, then the linear system solve generically gives behavior as in (b), because the growth factor of such an $\mathbf{x}$ is $O(1)$.

BONUS See discussion above, and draw yourself a sphere mapping under A to a very elongated ellipse. The clue is that $\mathbf{x}$ is size 1 in (b) (so that `bvec` aligns with output ellipse semi-major axis), but the largest possible size 1.39×10^{14} in (c) (so that `cvec` aligns with semi-minor axis).

Exactly the same issue comes up in Midterm 1, 2013, bonus for #4, explained in the solutions.

(d) See eg Ben proof.

5. 7 pts. Nothing too taxing here, just fun. With side length less than 10^{-3}, you got 7 levels of recursion since $(1/3)^7 < 10^{-3} < (1/3)^6$. It was incredibly slow to run though, and I don't know why. Maybe we could use `profiler` on the code? Tom and Nate did the full Koch snowflake for fun!