Math 116 Numerical PDEs: Homework 2–debriefing

January 25, 2012

To show large and small numbers on the same plot, it is essential to use a log scale (otherwise you can't communicate anything about the small ones!) Each plot should communicate effectively to an audience—this is an essential professional skill as a scientist / applied mathematician. This means wisely-chosen axes, labels, comments or discussion relating to class material if asked for. Often in numerical analysis a convergence plot *is needed, which clearly shows algebraic ($N^{-\alpha}$) or exponential (K^{-N}) convergence.*

1. [4 pts] For a), way I has maximum error $2\varepsilon_{mach}$ but way II only ε_{mach}, or you can explicitly show how the rounding to the numbers $1 + 2.2 \times 10^{-16}$ or $1 + 4.4 \times 10^{-16}$ occurs.

2. [4 pts: 2 + 2] Stability in b) was the tricky bit: if you examine $\tilde{f}(x) = [1 + x(1 + \varepsilon_1)](1 + \varepsilon_2)$ you will see it is *already* in the form of $f(\tilde{x})(1 + \varepsilon_4)$ for $\tilde{x} = x(1 + \varepsilon_1)$, and with $\varepsilon_4 = \varepsilon_2$. So the proof is almost trivial this way, once you see it. See Katie's or Lin's solution.

   However, attempting to expand the formula for relative error was messier, and people forgot to check that there exists an ε^4 such that this relative error is small for annoying cases like $x = -1$ (Brad and Jeff).

   Note that these stability tests are *uniform* wrt x. Ie, the constant in $O(\varepsilon_{mach})$ cannot depend on x. Some of you said, more loosely, that *for each* x the epsilons are $O(\varepsilon_{mach})$, which is merely pointwise convergence. Check difference: pointwise vs uniform.

3. [4 pts] Showing condition number of the "polynomial roots" problem is infinite, is enough here, since you know that the condition number of the entire "eigenvalues of diagonal matrix" problem is 1.

   A key point is that $O(\sqrt{\varepsilon_{mach}})$ errors only arise when roots are equal—no-one mentioned this. (Multiple p-fold degeneracy even worse, $O(\varepsilon_{mach}^{1/p})$).

   You only had to do pencil+paper work here, but coding it is possible too. This is actually hard to demonstrate in practice via MATLAB, since *e.g.* `A = diag(1 + 1e-14*[-1 1]); roots(poly(A))-1` gives exactly zero (no error), due to rounding of the 3 poly coeffs to integers $(1, -2, 1)$. Some perturbation about things not represented exactly is needed, *e.g.* `A = diag(sqrt(2) + 1e-14*[-1 1]); roots(poly(A))-sqrt(2)` which gives errors around 10^{-8}, *i.e.* $O(\sqrt{\varepsilon_{mach}})$ as expected.

4. [4 pts] See Katie's soln.

   BONUS: $\sup_{x\in[-1,1]} l_{n/2}(x)$ blows up exponentially, roughly as 2^n asymptotically as $n \to \infty$. We discussed a heuristic reason for this in class, based on $\phi(\pm 1)$ being $\ln 2$ greater than $\phi(0)$.

5. [7 pts: 2+2+3] See Brad solution.

   A semilogy plot of abs val here would be very informative for the interpolant error plots: this would show the variation of interpolation error with x; this is precisely the $\phi(z)$ potential function from lecture (smaller by $\ln 2$ in middle than ends, leading to exponential difference in size of $\prod_j (x - x_j)$ and hence pointwise interpolation error. (No-one did this but trivial to try!)

   For the convergence plot, a semilogy is *essential* otherwise you can't see how small the error is getting!

   The BONUS is to realise that it is rounding error from summing exponentially-growing Lagrange basis functions which causes the sudden change from exponential convergence to divergence at around $n = 27$. So this is another good reason why Chebychev nodes are better since the basis funcs then never exceed 1 in size.

6. [5 pts: 2 for code tweak, rest for investigation I requested] [Sorry about typo: by #6b I meant #5b].

   See eg Brad's.

   I was looking for discussion on each of the points I listed, so even if your codes were good, I didn't give you full marks if discussion was missing. "Runge phenomenon" means: asymptotic divergence (blow up) of sup of interp error as $n \to \infty$ for a function with nearby poles such as the Runge function does at $\pm i/5$. So, really you should check a sequence of increasing n, as you did at end of #5, and make a convergence plot. Error can then go down to 10^{-15}, as good as could hope for. (Eg Vipul showed for the entire func).

   It's rather peculiar that for $(1 + 25x^2)^{-1}$, 26 Chebychev nodes gives worse error (0.013) than for 25 nodes (0.008). There is an even-odd oscillation here, and I don't know why.

   Finally, you can predict the convergence rate of this Chebyshev interpolation, via the largest *Berstein ellipse* that doesn't touch the nearby poles. See Trefethen's ATAP book.