Math 56 Compu & Expt Math, Spring 2013: HW5 Debriefing

May 7, 2013

1. 3+4 = 7 pts

   (a) Derivative has nth Fourier coeff in times that of the value. $e^{\sin x}$ is entire; convergence of f' v. similar to that of f, is super-exponential. You all got this. (Notice how much better it is as a way to approximate the derivative than finite-differencing is.)

   (b) The convergence in man error should be 2nd-order algebraic, see eg John, Many of you got only 1st-order convergence. I did not have time to debug your codes to find out why - come and talk if stuck here.

   $|\sin^3 x| \in C^2$ but $\notin C^3$. Eg see Hanh analysis, simplest. The theorem you proved in HW4 #1(f) says convergence in f is at least $O(|n|^{-k})$ if $f \in C^k$. Applying that would give 2nd-order. Two effects (I treated as BONUS) are missing in this analysis; and these effects actually cancel out: i) the derivative would converge one order worse than the value (since the Fourier coeffs are $O(|n|)$ bigger), ii) the theorem is not acutally optimal, and you can in fact get a whole extra order of convergence: error in value is $O(|n|^{-k+1})$ if $f \in C^k$ and of "bounded variation". Check out Kyutae's bonus-worthy analysis of this!

2. 7 pts.

   The points here are for getting the code to work (small error norm, should be at least 10^{-11}), and timing it. As you all found, FFTW is around 1000 times faster—a great reason to program in C, and think very hard about memory movement, cache, CPU design, etc. See wikipedia for Cooley–Tukey algorithm.

3. 4+2 = 6 pts.

   (a) $\hat{f}_{654}$ is the complex conjugate of $\hat{f}_{-654}$, which must hold since the signal vector $\mathbf{f}$ is real-valued. (We did this in lecture—prove it as an exercise).

   (b) See my `audiofft.m` example from lecture. If m is the index of the peak coefficient, and $T = N/f_{\text{sam}}$ is the total sample time, where $N = 2^{16}$ is the sample vector length and $f_{\text{sam}} = 44100$ is the sampling freq. So the peak freq is $m/T = 440$ Hz (this is A above "middle C"). See Kyutae.

   (c) You probably cannot hear any tone above the noise (at least I cannot!) To compute amplitude, `abs(ft(654+1))/N` returns the amplitude of *one* complex exponential mode, but you need to double it since there is the negative freq mode present too (they combine to give a real-valued sine wave), to give amplitude of 0.0129. Meanwhile the mean square overall amplitude is `sqrt(mean(abs(f).^2))` which is 0.226. Thus the signal is amplitude about 5% of the overall.

4. 2+2+2+2+2 = 10 pts

   (a) Eg see Tom's working which shows the structure nicely.

   (b) Here the last entry is wrapped around and added to the first.

   (c) As you all discussed, $N_1 + N_2 - 1$ is sufficient.

   (d) Kyutae has a good explanation of how the sum is simplified.

   (e) Some great LaTeX use here to show the expanded double sum. Or see Hanh or Ben's concise proof.

5. 6 pts.

   This was fun and easy. You needed to zero-pad out to length $N_1 + N_2 - 1$ as above.

   Unfortunately if `wavwrite` is given data outside the range $[-1, 1]$ it will "clip" it to these maximum values, which sounds horrible. I didn't penalize this. Use `y = y/max(abs(y));` before writing out.

   Kunyi did a multiple-echo version!

6. 2+3+2 = 7 pts

   (a) Just reading in here. Don't forget `axis equal` to view correctly.

   (b) I was not worried about zero-padding here (in fact I used the periodic convolution to blur it in the first place). John's deconvolved picture was upside-down; this is due to using `ifft2` instead of `fft2`, since recall that (in the 1D case) F^2 is N times a permutation matrix that reverses the order of the elements.

   (c) iid Gaussian white noise with unit standard deviation given by `randn(512,512)`. I chose the noise level 3×10^{-5} so that the noise nearly swamps the image after dividing by this aperture's FFT. Your explanations that dividing by small numbers amplifies noise were good.

BONUS One way to improve this (as Hanh realised) is just kill (set to zero) the Fourier coefficients that get divided by really small coefficients of the aperture. E.g. cut-off below 10^{-4} in the aperture coefficient magnitudes (although this is a subjective choice based on the observed noise level; Hanh chose 10^{-3}). The few missing data don't affect the image quality too much, and the noise is much reduced. This is called *Tikhonov regularization*, very useful in inverse problems like this one.