Math 56 Compu & Expt Math, Spring 2013: HW4 Debriefing

May 1, 2013

1. 2+2+2+3+2+3 = 14 pts
   (a) Don't forget to integrate here! Eg seen Ben.
   (b) The easiest idea was to pick a low-order polynomial such as $a+x$ and solve for a to make it true. Also see Kunyi's $\cos x$. In general you can use Gram-Schmidt from Math 22. Choices involving $1/x$ are not in the space L^2, i.e. not square-integrable. (I meant non-zero function, since the zero function is also orthogonal—thanks Kyutae!)
   (c) Trickiest bit was remembering that the sum over $n \in \mathbb{Z}$ splits into the $n = 0$ term plus *twice* the sum over natural numbers. See eg Kunyi.
   (d) Note the domain $(-\pi, \pi)$ differs from the standard one, and the one in the fourier worksheet. $m = 0$ has to be dealt with separately. Note from this series you can prove $\sum_{k=1}^{\infty} k^{-4} = \pi^4/90$.
   (e) Differentiation brings down a factor in in the nth Fourier coefficient. Main idea: bounded function $\Rightarrow$ bounded coefficients via the projection formula, or by Parseval (the latter gets you $o(1/|n|)$, as Hanh did). It also is possible to argue that if f convergent at any x, then since $|e^{inx}| = 1$ the coefficients must form a bounded sequence (see eg Jon).
   (f) Super-algebraic, which is faster than any algebraic order, but slower than exponential (viz $e^{-c\sqrt{n}}$).
2. 3+3+2+2+2 = 12 pts.
   (a) Several had trouble with spotting that there are curves where $mj = const$, forming hyperbolae. See Kyutae.
   (b) F^2 is N times a permutation matrix which reflects the order cyclically about 0. See Ben's nice LaTeXhere.
   (c) Identity times N^2.
   (d) See Tom. Any eigenvalue of F must be a 4th root of N^2 since this is what all eigenvalues of F^4 are.
   (e) Unit condition number is common to all multiples of a unitary matrix.
3. 4+3+4 = 11 pts.
   (a) Since $f \in C^\infty$, super-algebraic decay of Fourier coefficients. Actually, since f is entire, we even expect (and see) super-exponential!
   (b) Tricky step here is realizing that $\tilde{f}_m$ for $m = N/2$ up to $N-1$ are actually telling you (to a good approximation) the Fourier coefficients $\hat{f}_m$ for $m = -N/2$ up to -1.
   (c) As Kyutae explains, the N required for interpolation to $\varepsilon_{\text{mach}}$ is twice the Fourier index at which the coefficients have decayed to around $\varepsilon_{\text{mach}}$. This is the Nyquist sampling theorem in action.
4. 3+4+2 = 9 pts. This was a hard one, since you have to think clearly and stop algebra from being too messy.
   (a) See Tom.

(b) For double sums you need different index labels, as in worksheet. The goal is to get some function of f plus the sum of squared magnitudes of the c_n. Hanh realized the nice trick that you can cancel out the coefficients $\hat{f}_n$ for $|n| \le N/2$ at the start.

(c) Once you've realized that setting all c_n to zero minimizes error, you can read off this (best) error as $\|f\|^2 - \sum_{|n|\le N/2} |\hat{f}_n|^2$. See Kyutae. You could also simply get this from Parseval as the squared L_2-norm of function built from the "tail" of the series, i.e. $|n| > N/2$.

5. 3+5 = 8 pts. Ideally you should repeat timing tests multiple times, since your CPU is also busy doing other stuff sometimes.

(a) You generally find library (the BLAS) around 20-1000 factor faster than naive Matlab loop! Depends on your machine. Tells you to exploit linear algebra libraries whenever possible.

(b) Fastest is $2^{13} = 8192$. Slowest is usually one of the 10 primes in the interval, which you can find with

```
for i=8100:8200, if numel(factor(i))==1, disp(i); end, end
```

Some found $8131 = 47 \times 173$ is slowest. Fastest is around 10-20 times the slowest in this range. See eg Hanh's timing plot. Some of you interpreted the number of factors as *unique* factors. Since ambiguous, I accepted this. But the point was that having many prime factors means that they're all *small*, which is good for the FFT algorithm. Any more requires digging into FFT algorithms (see the FFTW which Matlab uses).