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

due 10am Thursday April 18th

1. Here you learn how to "roll your own" finite difference formulae. Let's say you have access to f at only x, $x+h$, and $x+2h$, and want a 2nd-order accurate approximation to $f'(x)$. Note that this is at the leftmost point of the three; e.g. at the extreme end of a grid of values.

   (a) Set $f'(x) \approx af(x) + bf(x+h) + cf(x+2h)$, expand the right-hand side via Taylor series about x, then write out the three rows of a linear system resulting from equating powers of h^0, h^1 and h^2. Write your linear system in matrix-vector notation.
   [LaTeX hint: `\left[\begin{array}{lll} x & y & z \\ w ... \end{array}\right]`]

   (b) Solve the system either by hand or computer, hence write your new finite difference formula. How do you know the solution is unique?

   (c) Give a *rigorous* upper bound on the error of this formula (in exact arithmetic, i.e. ignore rounding).

2. Stability.

   (a) Show whether subtraction $x_1 - x_2$ is backwards stable (with respect to the two input data) under the rules of floating point.

   (b) In a worksheet you found that $1 + x$ as done by the rules of floating point arithmetic is not backward stable. Show whether $1 + x$ is *stable* or not.

3. Here's a new formula for matrix 2-norm:

$$\|A\| = \sqrt{\lambda_{\max}(A^TA)}, \qquad \text{where } \lambda_{\max}(A^TA) \text{ is the largest eigenvalue of the matrix } A^TA.$$

   (a) Let $A = \left[\begin{array}{cc} 1 & -1 \\ 2 & 2 \end{array}\right]$. Use the new formula to compute by hand $\|A\|$. How does it compare to the size of the largest eigenvalue of A? (for which you can use `eig`)

   (b) Use this to compute the matrix condition number $\kappa(A)$. Is it well-conditioned?

   (c) Take 100 points $\mathbf{x} \in \mathbb{R}^2$ equi-spaced on the unit circle, and plot them, and $A\mathbf{x}$ for each. What geometric property does $\kappa(A)$ measure of the ellipse produced?

4. Download the two 100×100 matrices `A1` and `A2` from the HW page, and use `textread` to read them into Matlab (you will need to `reshape` them).

   (a) Compare their matrix 2-norms and condition numbers. What worst-case relative errors do you expect for solving linear systems with matrix `A1`? With `A2`? (Use our backward stability theorem, and assume standard double precision.)

   (b) Let's focus on $A =$ `A1`, and load in the RHS $\mathbf{b} =$ `bvec` from the HW page. Solve $A\mathbf{x} = \mathbf{b}$. Then perturb $\mathbf{b}$ by a random vector of norm $\varepsilon_{\text{mach}}$ to get $\tilde{\mathbf{b}}$ (this emulates rounding error applied to the RHS), and solve again $A\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$. What relative norm change $\|\tilde{\mathbf{x}} - \mathbf{x}\|/\|\mathbf{x}\|$ results? Does this match your prediction from (a)?

   (c) Repeat (b) except using the RHS $\mathbf{c} =$ `cvec` from the HW page. Surprising? Is it consistent with (a)? Repeat for random unit-norm RHS vectors—do they behave more like $\mathbf{b}$ or like $\mathbf{c}$?

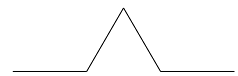
BONUS Explain the different behaviors [hint: $\|\mathbf{x}\|$], deducing how the directions of $\mathbf{b}$ and $\mathbf{c}$ relate to long and short axes of the ellipse of the image of the unit sphere under A.

(d) Given $A \in \mathbb{R}^{M \times P}$ and $B \in \mathbb{R}^{P \times N}$, prove a bound on $\|AB\|$ in terms of the norms of the individual matrices. [Hint: HW2 6(c).]

5. Recursion, and some "turtle" drawing in the complex plane.

(a) Make a function `y = koch(z,s)` which given complex numbers z and s returns $y = z + s$ and adds the line segment from z to y to the current figure (followed by `hold on`).

(b) Make a driver which uses four calls to `koch` to draw the generator for the Koch curve:



Each segment in the generator is length 1/3, and the angles are integer multiples of $\pi/3$. Here's how to do it using the stopping point y as the starting point for the next segment each time:

```
z = 0;
y = koch(z,1/3);
y = koch(y,1/3*exp(1i*pi/3));  ...
```

(c) Incorporate something like (b) into `koch` so that it draws a generator composed of four Koch curves unless $|s| < 10^{-3}$, in which case it reverts to the original simple line segment. As before, the y returned should be the final pen position. The call `koch(0,1)` should then produce a the Koch curve fractal—include a plot [Hint: it's a bit slow. Also, `axes equal`]