Math 56 Compu & Expt Math, Spring 2014: Homework 2

due 10am Thursday April 10th

As usual, include your code, even if one line. This can be in the same folder (needn't be in the PDF file)

1. Let's evaluate the function[1] $f(x) = \sqrt{1+x^2} - 1$ for small x, to high relative accuracy, using standard double-precision arithmetic.
   (a) What is the relative error in a naive implementation of $f(x)$ for $x = 10^{-8}$, and why? (You don't need full ε analysis here).
   (b) Write the first four nonzero terms of the Taylor series for f about zero. [Hint: binomial]
   (c) Use the rules of floating point, and just the first nonzero Taylor term, to estimate, for general small x, the *relative* error of the naive implementation. You may drop lower-order terms, eg $O(\varepsilon^2)$. Then for (roughly) what x_c is this 12 digits or better for all $x > x_c$?
   (d) For $|x| < x_c$, you'll switch over to Taylor series. How many terms do you need in the series to get 12 digit accuracy for all $|x| < x_c$? [Since x is so small, you may assume that the error is dominated by the first omitted term, equivalently that the unknown point in Taylor's theorem is very close to the origin.]
   (e) At $x = x_c$ the two methods should agree; check their relative difference. [If it's not around 12 digits, debug!]
2. Newton's iteration for rootfinding also works for complex numbers. Let's explore roots of $f(z) = z^3 - 1$.
   (a) Clearly $z = 1$ is the only real root. What root does the starting point $z_0 = i$ settle on under the Newton iteration? Find the third root, and plot all three as points in the complex plane. Explain their angles.
   (b) Using your brain, give all solutions to $z^5 = 2$, in $re^{i\theta}$ notation.
   (c) Back to our cubic f. When a starting point z_0 settles on a root, we say that z_0 is in the *basin* of that root. Write a code to explore in which of the three basins each point in the complex plane lies, coloring that point differently for each basin. Do this by setting up a square array of starting values, and doing each Newton iteration elementwise on this whole array at once (i.e. *vectorized* operation). You need only a few iterations until all pixels have converged. Choose the resolution high enough to see a beautiful fractal. [Hint: look at the suggested commands in 5b from HW1, also consider `imagesc(g,g,u)` where `u` is a square array of real values and `g` is the list you sent to `meshgrid`. What is a good way to convert the array of complex iterates into an array of real numbers classifying which roots they found?]

   BONUS produce zoomed plots exploring the fractal nature!
3. Recall the relative condition number is called κ.
   (a) What is κ for evaluating $\sin^{-1} x$ with respect to the input x at $x = 0.999999$?
   (b) What is κ for evaluating $\ln x$, and when is it large?
   (c) What is κ, with respect to input data c, for finding the roots of $x^2 - 2x + c = 0$? If $c = 1 - 10^{-16}$ how far apart are the roots, and what is κ?

[1] An application being: what is the length of the part of the line from $(0, 0)$ to $(1, x)$ lying outside the unit circle? This f also came up in research of Mike O'Neil at NYU.

(d) What relative error is it reasonable to demand from a backwards-stable algorithm to compute $\sin(10^{10})$?

4. Using Taylor's theorem rigorously prove that the error for the following finite difference formula for $f''(x)$ is $O(h^2)$ as $h \to 0$:

$$\frac{f(x+h) - 2f(x) + f(x-h)}{h^2}.$$

Verify this numerically by using the formula to approximate the second derivative of $\sin(5x)$ evaluated at $x = 1$. To do this, plot on a log-log scale the relative error of the finite difference formula as a function of h. Explore the full range of $10^{-16} \leq h \leq 1$, and comment.

5. Oddball questions.

(a) What should this Matlab code do, what *does* it do, and why?

```
x=0; while x~=1, x = x + 0.1; end
```

What is the right code to make this loop go a precise number of times?

(b) In the Simpson's episode *Treehouse of Horror VI*, Homer has a nightmare in which the equation

$$1782^{12} + 1841^{12} = 1922^{12}$$

flies past him. If true, this would contradict Fermat's last theorem (just proven when the episode aired). Give the relative error between left and right sides, and the number of binary digits needed to handle integers of this size correctly. [Adapted from Greenbaum-Chartier book, Ex. 5.13.]

(c) Let A and B be two dense matrices of size $n \times n$. How many flops does it take to compute AB? If $\mathbf{x}$ is a column vector length n, which of the *two* possible ways to compute $AB\mathbf{x}$ is better and why?