Assignment 1, due September 20

Corrections: [none yet]

1. We want to design a special floating point format that uses the fewest bits so that any x with $|x| \leq 400$ and $|x| > .1$ is represented to .05% relative accuracy (the relative error is not more than $5 \cdot 10^{-4}$). We need one sign bit, q exponent bits, p fraction bits and an exponent offset e_0. (These are car velocities in km/h. No car (hopefully) goes faster than $400\,\text{km/hr} \approx 240\,\text{mi/hr}$, and $.1\,\text{km/hr} \approx .06\,\text{mi/hr}$ is a "rolling stop".) Note that the range of speed exponents is not symmetric, so you may be able to save a bit by choosing the offset carefully.

2. We want to evaluate $f(x) = e^x - 1$ accurately in double precision floating point when x is close to zero. For this problem, assume that exponentiation and multiplication and addition are done with relative accuracy at least ϵ_{mach}, the machine precision of double precision arithmetic.

   (a) Estimate the relative accuracy of the direct code `f = np.exp(x)-1.` Assume that `np.exp(x)` evaluates e^x to full double precision floating point accuracy.

   (b) Estimate the relative accuracy of the Taylor series approximation `f = x + .5*x*x`. Assume that the arithmetic is done with double precision floating point accuracy and that the error in the Taylor approximation is $f(x) - (x + .5x^2) = (1/6)x^3$. (This is an accurate estimate of the error for the very small x we have in mind.)

   (c) Consider the hybrid approximation

   ```
   if ( np.abs(x) < epsilon ):
       f = x + .5*x*x
   else:
       f = np.exp(x)-1.
   ```

   Within the range of normalized numbers, the error for this approximation is largest around $x = \pm\epsilon$. For $|x| < \epsilon$ the Taylor expansion becomes more accurate as $|x|$ decreases. For $|x| > \epsilon$, the direct formula becomes more accurate as $|x|$ increases. Find the ϵ that minimizes this largest error. What is the worst case accuracy, given in the number of correct digits of the answer?

3. The *logistic function* with positive *length parameter* r is

$$s(x,r) = \frac{e^{x/r}}{1 + e^{x/r}} \; .$$

As a function of x, this function switches from $s \approx 0$ when x is a large negative number to $s \approx 1$ when x is a large positive number. It is used in *logistic regression*, which is one of the tools used in data science for classification. The parameter r controls the range of x values where the transition from 0 to 1 happens. The logistic function is used to create *soft threshholds*, which are an alternative to *hard threshholds*. Hard threshholds are equal to zero (on one side of some criterion) or one (on the other side). Evaluate the condition number of the problem of evaluating $s(x, 1)$ (set $r = 1$ for simplicity). Figure out whether it is well conditioned for large negative and positive x. The inverse logistic function problem is find x so that $s(x) = y$, for a given y between 0 and 1. Figure out whether the inverse logistic problem is well conditioned for y near 0 and y near 1.

4. The Fibonacci recurrence relation is

$$f_{n+1} = f_n + f_{n-1} \ .$$

The *initial conditions* are the values f_0 and f_1. Once the initial conditions are specified, the rest of the numbers f_n are determined. The *target values* are f_L and f_{L+1}, where L is some positive integer. We are interested in Fibonacci sequences of length L. *The* Fibonacci numbers are produced using initial conditions $f_0 = 1$ and $f_1 = 1$.

(a) Show that any Fibonacci sequence has the form

$$f_n = a_1 z_n^n + a_2 z_2^n \ .$$

The coefficients a_1 and a_2 depend on the initial conditions but the *roots* z_1 and z_2 do not. Hint: The sequence $f_n = z^n$ satisfies the Fibonacci recurrence if and only if z is one of the roots of a given quadratic *characteristic polynomial*. Show it is always possible to find a_1 and a_2 so that $f_0 = a_1 + a_2$ and $f_1 = a_1 z_1 + a_2 z_2$. The roots z_1 and z_2 are positive. Suppose $z_1 > z_2$ The larger root is the *golden mean*.

(b) Show that $f_{n+1}/f_n \to z_1$ as $n \to \infty$ unless $a_1 = 0$. In particular, show that *the* Fibonacci numbers have this property.

(c) Show that the condition number of computing the target values from the initial conditions is bounded as $L \to \infty$ unless $a_1 = 0$, when the condition number grows exponentially with L.

(d) Consider initial conditions with $a_1 \neq 0$ and consider the problem of computing the initial conditions from the target values. Show that the condition number of this problem grows exponentially with L.

5. Write a code in Python 3 related to Problem 4. The code should start with initial conditions (given in the code), compute target values, then reverse the recurrence to re-compute the initial conditions from the target values

$$f_0, f_1 \quad \overset{\text{floating point}}{\longrightarrow} \quad \widehat{f}_L, \widehat{f}_{L+1} \quad \overset{\text{floating point}}{\longrightarrow} \quad \widehat{f}_0, \widehat{f}_1$$

Print the error $f_0 = \widehat{f_0}$ as a function of L.

(a) Start with $f_0 = \sqrt{2}$ and $f_1 = e$. Show that the errors are roughly explained by the condition number analysis of Problem 4.

(b) Repeat the exercise with *the* Fibonacci numbers. Why is there no error until L reaches a given point? What determines the L value where error starts?