Homework 3

Due: 2:00pm Feb. 18th, 2016

Each problem is worth 10 points.

Exercise 1 [Floating point arithmetic]: Let a and b be two positive floating-point numbers with the same exponent. Explain why the computed difference $a \ominus b$ (the IEEE rounded difference) is always exact using IEEE arithmetic.

Exercise 2 [Conditioning, stability]: Let a, b, c be floating point numbers. The solution to

$$ax^2 + bx + c = 0$$

is given by the quadratic formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Assume $a \sim \mathcal{O}(1)$ and $b \sim \mathcal{O}(1)$. If a, b, c are accurate to full relative double precision, but $c \approx 0$, what goes wrong, if anything, in calculating the roots x_+ and x_-? (Think relative precision).

$$x_+ = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \qquad x_- = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

Is there another method that can avoid any problems? Can the quadratic formula be fixed?

Exercise 3 [LU factorization]: Write a function in Matlab that will compute the PLU-decomposition of an $n \times n$ matrix A using partial pivoting:

$$A = PLU.$$

The calling sequence of the function should be:

```
[pmat,lmat,umat] = partial_pivot_lu(amat)
```

Turn in your code, along with a printout of the output of the following Matlab commands in *short e* format (see Matlab preferences):

```
rng(7, 'twister')
n = 6;
amat = randn(n);
[pmat,lmat,umat] = partial_pivot_lu(amat)
```

You must also email me (`oneil@nyu.edu`) a copy of your code that I can execute.

- continued -

Exercise 4 [Complexity]: Estimate the number of operations required for Gaussian elimination (LU-factorization) with partial pivoting on an $n \times n$ matrix with half-bandwidth m (see Section 7.2.4 in the text for more info). You can use the fact that the half-bandwidth of the U factor is at most doubled by using partial pivoting.