Math 128a - Homework 1 - Due Feb 7 at the beginning of class
Corrections made on Feb 1 to Questions 1.6 and 4.3

1) In class we saw an example showing that in decimal floating point arithmetic, the computed value of xmid=(xlower+xupper)/2 is not necessarily between xlower and xupper, which would be a problem for the logic in bisection (in 3 decimal digit arithmetic, try xlower = .997 and xupper = .999). We will show that this is impossible in IEEE arithmetic, which is binary. In other words, we will show that in IEEE arithmetic xmid = $fl(fl(\text{xlower} + \text{xupper})/2)$ is in the interval [xlower,xupper], assuming overflow does not occur when adding xlower and xupper. (Here $fl(a \text{ op } b)$ means the floating point result of the operation a op b.

Part 1. Using the fact that IEEE arithmetic is correctly rounded, show that it is monotonic, that is if a, b, c, d and x are IEEE floating point numbers then

$$a \le b \text{ and } c \le d \quad \text{implies} \quad fl(a + c) \le fl(b + d)$$
$$a \le b \text{ and } 0 < x \quad \text{implies} \quad fl(a/x) \le fl(b/x)$$

(Similar facts hold for subtraction and multiplication, but we will not need these here.)

Part 2. Show that $fl(2 * x) = 2 * x$ exactly, assuming overflow does not occur.

Part 3. Show that $2 * \text{xlower} \le fl(\text{xlower} + \text{xupper}) \le 2 * \text{xupper}$.

Part 4. Conclude that $\text{xlower} \le fl(fl(\text{xlower} + \text{xupper})/2) \le \text{xupper}$.

Part 5. Where does this argument fail for correctly rounded decimal arithmetic?

Part 6. What happens if xlower and xupper are adjacent IEEE floating point numbers?

2) Suppose x is the exact answer to a problem, and $\widehat{x}$ is our approximate answer. In class we defined the absolute error in $\widehat{x}$ as $|x - \widehat{x}|$ and the relative error in $\widehat{x}$ as $|x - \widehat{x}|/|x|$. In this problem we will explore some simple properties of these error measures.

Write the base β expansion of $x > 0$ as $x = .x_1 x_2 \cdots x_n \cdot \beta^{e_x}$, and the base β expansion of $y > 0$ as $y = .y_1 y_2 \cdots y_n \cdot \beta^{e_y}$. We will say that x and y *agree to their leading d base β digits* if $|x - y| < \frac{1}{2}\beta^{\max(e_x, e_y) - d}$. For example, .1230 and .1226 agree to 3 decimal digits, as do 1.00 and .996, or .1233 and .1237.

Part 1. Suppose you print out $\widehat{x}$ as a base β number. Show that if the relative error $|x - \widehat{x}|/|x| < 1$, then the leading $\lfloor \log_\beta \frac{|x|}{|x - \widehat{x}|} \rfloor - 1$ nonzero base β digits of $\widehat{x}$ are correct, i.e. x and $\widehat{x}$ agree to that many digits. ($\lfloor x \rfloor$ is the *floor of x*, the largest integer less than or equal to x.)

Part 2. Suppose you have solved your problem and gotten $\widehat{x}$, and also a bound $e_{abs} \geq |x - \widehat{x}|$ on the absolute error (perhaps using rounding error analysis as described in class). You would like a bound $e_{rel} \geq |x - \widehat{x}|/|x|$ on the relative e rror. One obvious candidate is $e_{rel} = e_{abs}/|x|$, but of course you can't compute this because you don't know x (otherwise we wouldn't need an error bound!). So instead you try $e_{rel} = e_{abs}/|\widehat{x}|$. Show that it is ok to use $e_{abs}/|\widehat{x}|$ instead of $e_a bs/|x|$ by showing that

$$\frac{\frac{|x-\widehat{x}|}{|\widehat{x}|}}{1 + \frac{|x-\widehat{x}|}{|\widehat{x}|}} \leq \frac{|x - \widehat{x}|}{|x|} \leq \frac{\frac{|x-\widehat{x}|}{|\widehat{x}|}}{1 - \frac{|x-\widehat{x}|}{|\widehat{x}|}}$$

Conclude that if $e_{rel} \leq .1$, then the actual relative error satisfies $.8e_{rel} \leq |x - \widehat{x}|/|x| \leq 1.2e_{rel}$.

3) Let $1 + r = \prod_{i=1}^{n}(1 + \delta_i)$, where $|\delta_i| \leq \epsilon < 1$.

Part 1. Show that if $n\epsilon < 1$, then $|r| \leq n\epsilon/(1 - n\epsilon)$.

Part 2. Show that if $n\epsilon \leq .1$, then $r \leq 1.2n\epsilon$.

Part 3. In IEEE double precision, how big can n be and satisfy $n\epsilon \leq .1$?

Part 4. If you compute $p = \prod_{i=1}^{n} x_i$ in floating point arithmetic, and no over/underflow occurs, and $n\epsilon \leq .1$, about how many leading decial digits of the computed value of p are correct when using IEEE double precision arithmetic with $n = 10$? $n = 100$? $n = 1000$? $n = 10000$?

4) Suppose $x > 0$. Here are two Matlab algorithms for computing e^{-x}:

Algorithm 1: Compute e^{-x} using a Taylor expansion

```
s = 1; t = 1; i = 1;
while (abs(t) > eps*abs(s))
      ... stop iterating when adding t to s does not change s
      t = -t*x/i;
      s = s + t;
      i = i + 1;
end
result1 = s;
```

Algorithm 2: Compute e^{-x} as $1/e^x$, using a Taylor expansion for e^x

```
s = 1; t = 1; i = 1;
while (abs(t) > eps*abs(s))
      ... stop iterating when adding t to s does not change s
      t = t*x/i;
      s = s + t;
      i = i + 1;
end
result2 = 1/s;
```

Part 1. Run these two algorithms for x = 1:20, tabulating the relative errors and number of iterations to converge for each.

Part 2. Prove that the relative error of result2 is, as you observe, bounded by $(3i - 2)\epsilon$, i.e. very accurate. You may assume the error from terminating the Taylor expansion is smaller than round off error, and you may ignore terms proportional to ϵ^2. Confirm that $(3i - 2)\epsilon$ bounds the relative errors in your table above.

Part 3. Prove that the relative error of result1 is bounded by $3(i - 1)\epsilon e^{2x}$, i.e. it grows quickly with x, so that Algorithm 1 is much less accurate than Algorithm 2. You may make the same assumptions as before. Confirm that $3(i - 1)\epsilon e^{2x}$ bounds the relative errors in your table above.

Part 4. The computer implementation for e^x takes the same time for large and small arguments; i.e. it does not use a simple Taylor expansion, which would require more terms for larger arguments. Sketch an algorithm for e^x that does not take longer for large x. Use the fact that $e^x = 2^y$ where $y = x \cdot \log_2 e$, write $y = y_{int} + y_{frac}$ as a sum of an integer and a fraction less than 1, and use the fact that $2^y = 2^{y_{int}} \cdot 2^{y_{frac}}$ is to be rounded to a floating point number. How many term of a Taylor expansion of $2^{y_{frac}}$ are needed so that the remaining terms contribute less than ϵ to the relative error?