CSC336: Problem Set 1

Due on Friday, Oct 12, 2018

Zhongtian Ouyang

Problem 1

Errors

We assume π equals to 3.14159265358979 in this question

a)
Absolute Error $= 3.14 - \pi = -0.00159265358979 \approx -1.59 \times 10^{-3}$
Relative Error $= \frac{3.14-\pi}{\pi} \approx -5.07 \times 10^{-4}$

b)
Absolute Error $= 3.14159 - \pi = -0.00000265358979 \approx -2.65 \times 10^{-3}$
Relative Error $= \frac{3.14159-\pi}{\pi} \approx -8.45 \times 10^{-7}$

c)
Absolute Error $= 3.141592654 - \pi = 0.00000000041021 \approx 4.10 \times 10^{-10}$
Relative Error $= \frac{3.14-\pi}{\pi} \approx 1.31 \times 10^{-10}$

Problem 2

Rounding

(a)

$$5.35 \cdot 10^0 + 2.46 \cdot 10^{-2} = 5.3746 \approx 5.37 \cdot 10^0$$

(b)

$$4.53 \cdot 10^1 - 6.38 \cdot 10^{-1} = 44.662 \approx 4.47 \cdot 10^1$$

(c)

$$5.65 \cdot 10^1 + 5.23 \cdot 10^{-4} = 5.650523 \approx 5.65 \cdot 10^0$$

(d)

$$6.54 \cdot 10^4 - 8.73 \cdot 10^6 = -8.6646 \cdot 10^6 \approx -8.66 \cdot 10^6$$

(e)

$$5.21 \cdot 10^8 \times 4.25 \cdot 10^{-5} = 22.1425 \cdot 10^3 = 2.21425 \cdot 10^4 \approx 2.21 \cdot 10^4$$

(f)

$$-4.32 \cdot 10^7 \times 3.25 \cdot 10^3 = -14.04 \cdot 10^{10} = -1.404 \cdot 10^{11} \approx -inf$$

(g)

$$5.41 \cdot 10^{-5} \times 4.27 \cdot 10^{-5} = 23.1007 \cdot 10^{-10} = 2.31007 \cdot 10^{-9} \approx 2.31 \cdot 10^{-9}$$

(h)

$$-6.52 \cdot 10^{-6} \times 4.75 \cdot 10^{-6} = -30.97 \cdot 10^{-12} = -0.3097 \cdot 10^{-10} \approx -0.310 \cdot 10^{-10}$$

(i)

$$-6.46 \cdot 10^{-7} \times 1.32 \cdot 10^{-6} = -8.5272 \cdot 10^{-13} = 0.0085272 \cdot 10^{-10} \approx 0.01 \cdot 10^{-10}$$

(j)

$$3.82 \cdot 10^{-6} \times 1.25 \cdot 10^{-7} = 4.775 \cdot 10^{-13} = 0.004775 \cdot 10^{-10} \approx 0.00 \cdot 10^{-10} = 0$$

Problem 3

Conditioning

a)
$f(x) = tan(x) \rightarrow f'(x) = sec^2(x)$
The condtion number is:

$$\frac{xf'(x)}{f(x)} = \frac{x \cdot sec^2(x)}{tan(x)} = \frac{x}{sin(x)cos(x)}$$

When x is close to 0:

$$\lim_{x \to 0} \frac{xf'(x)}{f(x)} = \lim_{x \to 0} \frac{x}{sin(x)cos(x)}$$

Because both the numerator and denominator evaluates to zero, we can apply l'Hopital's Rule:

$$\lim_{x \to 0} \frac{x}{sin(x)cos(x)} = \lim_{x \to 0} \frac{1}{cos^2(x) - sin^2(x)} = \lim_{x \to 0} \frac{1}{cos(2x)} = 1$$

Since 1 is means that we don't lose accuracy, this is well-conditioned.

When x is close to $\pi/2$:

$$\lim_{x \to \pi/2} \frac{xf'(x)}{f(x)} = \lim_{x \to \pi/2} \frac{x}{sin(x)cos(x)} = \infty$$

this is ill-conditioned.

b)

```
interval = 1e-8;
results1 = [];
for i = (0-2*interval):interval:(0+2*interval)
    results1 = [results1,tan(i)];
end
fprintf('%e ',results1);
fprintf('\n');

results2 = [];
val = pi/2;
for i = (val-5*interval):interval:(val-interval)
    results2 = [results2,tan(i)];
end
fprintf('%e ',results2);
fprintf('\n');
```

The output from the program:

```
-2.000000e-08 -1.000000e-08 0.000000e+00 1.000000e-08 2.000000e-08
2.000000e+07 2.500000e+07 3.333333e+07 5.000000e+07 1.000000e+08
```

When x is close to 0, for example, suppose $x = 1.0 \cdot 10^{-8}$ and $\hat{x} = 2.0 \cdot 10^{-8}$, then $y = 1.0 \cdot 10^{-8}$ and $\hat{y} = 2.0 \cdot 10^{-8}$. The relative change in x = $\frac{\hat{x}-x}{x} = 1$ and the relative change in y = $\frac{\hat{y}-y}{y} = 1$. This support the conclusion that tan(x) is well-conditioned when x is close to 0.

When x is close to $\pi/2$, for example, suppose $x = \pi/2 - 2.0 \cdot 10^{-8}$ and $\hat{x} = \pi/2 - 1.0 \cdot 10^{-8}$, then $y = 5.0 \cdot 10^{7}$ and $\hat{y} = 1.0 \cdot 10^{8}$. The relative change in x = $\frac{\hat{x}-x}{x} \approx 6.366198 \cdot 10^{-9}$ and the relative change in y = $\frac{\hat{y}-y}{y} = 1$. This support the conclusion that tan(x) is ill-conditioned when x is close to $\pi/2$.

Problem 4

2017 final

Q2)
When x is a small but positive value, the true value of cos(x) is so close to 1 such that in Matlab, cos(x) is rounded to 1. so F evaluates to $(1-1)/x^2$, which is equals to zero. To fix this problem, we can rewrite the statement in the following way.

$$F = \frac{(1-cos(x))}{x^2} = \frac{(1-cos(x))(1+cos(x))}{x^2 \cdot (1+cos(x))} = \frac{1-cos^2(x)}{x^2 \cdot (1+cos(x))} = \frac{sin^2(x)}{x^2 \cdot (1+cos(x))}$$

Q3)
Assume $|x| \geq |y|$, the statement can be rewritten as:

$$G = \sqrt{x^2+y^2} = |x| \cdot \frac{1}{|x|}\sqrt{x^2+y^2} = |x|\sqrt{\frac{x^2}{|x|^2}+\frac{y^2}{|x|^2}} = |x|\sqrt{1+(\frac{|y|}{|x|})^2}$$

The value of $(\frac{|y|}{|x|})^2$ must between zero and one. So as long as x, y, and the final result can be expressed in IEEE Double-precision floating-point number, the intermediate values will not overflow or underflow.

if $|y| \geq |x|$, we can follow similar steps and get:

$$G = \sqrt{x^2+y^2} = |y| \cdot \frac{1}{|y|}\sqrt{x^2+y^2} = |y|\sqrt{(\frac{|x|}{|y|})^2+1}$$

Below is an example code in Matlab using max and min function.

```
function result = f(x_in,y_in)
    x = abs(x_in);
    y = abs(y_in);
    if x == 0 && y == 0
        result = 0;
    else
        nums = [x y];
        result = max(nums) * sqrt(1 + (min(nums)/max(nums))^2);
    end
end
```

Problem 5

Matlab

The function exp1:

```
function result = exp1(x)
    prev = -1;
    cur = 0;
    cycle = 0;

    while cur ~= prev
        prev = cur;
        cur = cur + (x^cycle)/factorial(cycle);
        cycle = cycle + 1;
    end

    result = cur;
end
```

The program used to test x with value from -25 to 25:

```
exp1s = [];
exps = [];
errors = [];
xs = [];

for x = -25:25
    exp1_result = exp1(x);
    exp_result = exp(x);
    relative_error = (exp1_result - exp_result)/exp_result;

    xs = [xs x];
    exp1s = [exp1s exp1_result];
    exps = [exps exp_result];
    errors = [errors relative_error];
end

A = [xs; exp1s; exps; errors];
fprintf('%6s %12s %12s %12s\n', 'x', 'exp1(x)', 'exp(x)', 'error');
fprintf('%6d %12.4e %12.4e %12.4e\n', A);
```

The printed result from running the program:

x	exp1(x)	exp(x)	error
−25	8.0866e−07	1.3888e−11	5.8226e+04
−24	3.7628e−07	3.7751e−11	9.9664e+03
−23	6.8990e−09	1.0262e−10	6.6229e+01
−22	−3.1820e−08	2.7895e−10	−1.1507e+02
−21	2.7590e−08	7.5826e−10	3.5387e+01
−20	4.1736e−09	2.0612e−09	1.0249e+00
−19	2.5538e−09	5.6028e−09	−5.4420e−01
−18	1.5984e−08	1.5230e−08	4.9480e−02
−17	4.1442e−08	4.1399e−08	1.0196e−03
−16	1.1257e−07	1.1254e−07	2.8526e−04
−15	3.0591e−07	3.0590e−07	1.0354e−05
−14	8.3152e−07	8.3153e−07	−8.6112e−06
−13	2.2603e−06	2.2603e−06	−1.2997e−06
−12	6.1442e−06	6.1442e−06	6.1218e−08
−11	1.6702e−05	1.6702e−05	7.6483e−08
−10	4.5400e−05	4.5400e−05	−7.2342e−09
−9	1.2341e−04	1.2341e−04	−5.4918e−10
−8	3.3546e−04	3.3546e−04	−1.4773e−10
−7	9.1188e−04	9.1188e−04	1.2606e−11
−6	2.4788e−03	2.4788e−03	−7.2503e−13
−5	6.7379e−03	6.7379e−03	2.1369e−13
−4	1.8316e−02	1.8316e−02	1.4396e−14
−3	4.9787e−02	4.9787e−02	8.3623e−16
−2	1.3534e−01	1.3534e−01	4.1018e−16
−1	3.6788e−01	3.6788e−01	3.0179e−16
0	1.0000e+00	1.0000e+00	0.0000e+00
1	2.7183e+00	2.7183e+00	0.0000e+00
2	7.3891e+00	7.3891e+00	−2.4040e−16
3	2.0086e+01	2.0086e+01	−3.5376e−16
4	5.4598e+01	5.4598e+01	5.2056e−16
5	1.4841e+02	1.4841e+02	−1.9150e−16
6	4.0343e+02	4.0343e+02	0.0000e+00
7	1.0966e+03	1.0966e+03	−6.2201e−16
8	2.9810e+03	2.9810e+03	−1.5255e−16
9	8.1031e+03	8.1031e+03	0.0000e+00
10	2.2026e+04	2.2026e+04	−3.3033e−16
11	5.9874e+04	5.9874e+04	0.0000e+00
12	1.6275e+05	1.6275e+05	−3.5764e−16
13	4.4241e+05	4.4241e+05	−1.3157e−16
14	1.2026e+06	1.2026e+06	1.9361e−16
15	3.2690e+06	3.2690e+06	0.0000e+00
16	8.8861e+06	8.8861e+06	0.0000e+00
17	2.4155e+07	2.4155e+07	3.0845e−16
18	6.5660e+07	6.5660e+07	0.0000e+00
19	1.7848e+08	1.7848e+08	−1.6698e−16
20	4.8517e+08	4.8517e+08	−2.4571e−16
21	1.3188e+09	1.3188e+09	−3.6156e−16
22	3.5849e+09	3.5849e+09	2.6602e−16
23	9.7448e+09	9.7448e+09	1.9573e−16

```
    24    2.6489e+10    2.6489e+10    0.0000e+00
    25    7.2005e+10    7.2005e+10    0.0000e+00
```

b)
When x is positive, the values are very accurate. But when x is negative, the approximations become more and more inaccurate as x gets smaller. The reason is that when x is positive, all terms are positive, we are accumulating to the final value; However, when x is negative, the even terms of the series is positive, while the odd terms of the series is negative. Therefore, the final result could be smaller than the intermidiate values. And when x gets smaller, the final result become closer to 0, while the intermediate values becomes larger. In lectures, we have shown that the if the intermediate values are much larger than the final result, the final result could be very inaccurate.

c)

```
function result = exp2(x_in)
    x = abs(x_in);
    prev = -1;
    cur = 0;
    cycle = 0;

    while cur ~= prev
        prev = cur;
        cur = cur + (x^cycle)/factorial(cycle);
        cycle = cycle + 1;
    end

    if x_in >= 0
        result = cur;
    else
        result = 1/cur;
    end
end
```

The program for testing is almost identical to the one in part a), so I don't repeat it here.
We can see from the results below that all of the approximations are accurate now.

```
     x       exp2(x)         exp(x)          error
   -25    1.3888e-11    1.3888e-11   -1.1633e-16
   -24    3.7751e-11    3.7751e-11    0.0000e+00
   -23    1.0262e-10    1.0262e-10   -2.5190e-16
   -22    2.7895e-10    2.7895e-10   -1.8534e-16
   -21    7.5826e-10    7.5826e-10    4.0909e-16
   -20    2.0612e-09    2.0612e-09    2.0066e-16
   -19    5.6028e-09    5.6028e-09    0.0000e+00
```

−18	1.5230e−08	1.5230e−08	0.0000e+00
−17	4.1399e−08	4.1399e−08	−3.1969e−16
−16	1.1254e−07	1.1254e−07	0.0000e+00
−15	3.0590e−07	3.0590e−07	0.0000e+00
−14	8.3153e−07	8.3153e−07	−2.5466e−16
−13	2.2603e−06	2.2603e−06	1.8737e−16
−12	6.1442e−06	6.1442e−06	4.1358e−16
−11	1.6702e−05	1.6702e−05	0.0000e+00
−10	4.5400e−05	4.5400e−05	2.9851e−16
−9	1.2341e−04	1.2341e−04	−2.1963e−16
−8	3.3546e−04	3.3546e−04	1.6160e−16
−7	9.1188e−04	9.1188e−04	7.1338e−16
−6	2.4788e−03	2.4788e−03	0.0000e+00
−5	6.7379e−03	6.7379e−03	2.5746e−16
−4	1.8316e−02	1.8316e−02	−3.7885e−16
−3	4.9787e−02	4.9787e−02	2.7874e−16
−2	1.3534e−01	1.3534e−01	2.0509e−16
−1	3.6788e−01	3.6788e−01	−1.5089e−16
0	1.0000e+00	1.0000e+00	0.0000e+00
1	2.7183e+00	2.7183e+00	0.0000e+00
2	7.3891e+00	7.3891e+00	−2.4040e−16
3	2.0086e+01	2.0086e+01	−3.5376e−16
4	5.4598e+01	5.4598e+01	5.2056e−16
5	1.4841e+02	1.4841e+02	−1.9150e−16
6	4.0343e+02	4.0343e+02	0.0000e+00
7	1.0966e+03	1.0966e+03	−6.2201e−16
8	2.9810e+03	2.9810e+03	−1.5255e−16
9	8.1031e+03	8.1031e+03	0.0000e+00
10	2.2026e+04	2.2026e+04	−3.3033e−16
11	5.9874e+04	5.9874e+04	0.0000e+00
12	1.6275e+05	1.6275e+05	−3.5764e−16
13	4.4241e+05	4.4241e+05	−1.3157e−16
14	1.2026e+06	1.2026e+06	1.9361e−16
15	3.2690e+06	3.2690e+06	0.0000e+00
16	8.8861e+06	8.8861e+06	0.0000e+00
17	2.4155e+07	2.4155e+07	3.0845e−16
18	6.5660e+07	6.5660e+07	0.0000e+00
19	1.7848e+08	1.7848e+08	−1.6698e−16
20	4.8517e+08	4.8517e+08	−2.4571e−16
21	1.3188e+09	1.3188e+09	−3.6156e−16
22	3.5849e+09	3.5849e+09	2.6602e−16
23	9.7448e+09	9.7448e+09	1.9573e−16
24	2.6489e+10	2.6489e+10	0.0000e+00
25	7.2005e+10	7.2005e+10	0.0000e+00