Numerical Analysis (CS 450)

Homework Set 1, Bill Karr

Problem 1: (15 points)

Consider the function $f : \mathbb{R}^2 \to \mathbb{R}$ defined by $f(x) = x - y$. Measuring the size of the input (x, y) by $|x| + |y|$, and assuming that $|x| + |y| \approx 1$ and $x - y \approx \varepsilon$, show that $\text{cond}(f) \approx \varepsilon$. What can you conclude about the sensitivity of subtraction?

Solution & Proof. To compute the condition number of f, we need to examine

$$\text{cond}(f) = \frac{\dfrac{|f(x+\Delta x, y+\Delta y) - f(x,y)|}{|f(x,y)|}}{\dfrac{|(\Delta x, \Delta y)|}{|(x,y)|}} = \frac{\dfrac{|\Delta x - \Delta y|}{|x-y|}}{\dfrac{|\Delta x| + |\Delta y|}{|x| + |y|}} \approx \frac{\dfrac{|\Delta x - \Delta y|}{\varepsilon}}{\dfrac{|\Delta x| + |\Delta y|}{1}} = \frac{1}{\varepsilon}\frac{|\Delta x - \Delta y|}{|\Delta x| + |\Delta y|}$$

Using the triangle inequality, we see that $|\Delta x - \Delta y| \leq |\Delta x| + |\Delta y|$ and equality can be achieved, for example, if $\Delta x > 0$ and $\Delta y < 0$. Thus, $\text{cond}(f) \approx \frac{1}{\varepsilon}$ for $x - y \approx \varepsilon$.

Thus, subtraction of two numbers is very sensitive when the two numbers are relatively large, but have relatively small difference. It's easy to have a large relative error. □

Problem 2: (15 points)

For computing the midpoint m of an interval $[a, b]$, which of the following two formulae is preferable in floating-point arithmetic?

(i) $m = (a + b)/2$

(ii) $m = a + (b - a)/2$

Why? When? Devise examples for which the "midpoint" given by the formula lies outside of the interval $[a, b]$. Specify the floating point system used, specify a and b in floating-point representation. Write down all intermediate results for both cases. Point out the steps where the problem occurs.

Solution. Neither formula is the "preferable" formula. They both work in certain circumstances. Sometimes one works and the other doesn't. Let us work with in a binary floating-point system ($\beta = 2$) with $p = 2$, $U = 4$, and $L = -4$ where anything that overflows is considered infinite.

Let $a = -(0.1)_2 \times 2^4$ and $b = (1.1)_2 \times 2^4$. Then, $a + b = (1.0)_2 \times 2^4$. We divide by two and obtain $m = (1.0)_2 \times 2^3$. However, $b - a = (1.1)_2 \times 2^5$ which is larger than overflow and thus infinite. We compute $m = a + (b - a)/2 = a + \infty = \infty$ which is larger than b. Thus, the second formula fails.

On the other hand, if $a = (0.1)_2 \times 2^4$ and $b = (1.1)_2 \times 2^4$, then $a + b = (1.1)_2 \times 2^5$ which is infinite and $m = \infty$ using the first formula which is larger than b. However, the second formula doesn't have the same issue. We get $m = a + (b - a)/2 = (0.1)_2 \times 2^4 + (1.0)_2 \times 2^4/2 = (1.0)_2 \times 2^4$. □

Problem 3: Bessel recurrence vs. floating point (20 points)

(a) Write a program that tests the accuracy to which the values returned from `scipy` obey

$$J_{n+1}(z) = (2n/z)J_n(z) - J_{n-1}(z). \tag{1}$$

This should yield roughly machine precision in each case.

(b) Write a program that uses the values for $J_0(z)$ and $J_1(z)$ (obtained from `scipy`) and the so-called 'recurrence relation' (1) to compute the values of $J_2(z), ..., J_{50}(z)$.

Print these values for each n, and also print the relative error compared to the value returned by `scipy` and report your results.

For your experiments, fix $z = 20$.

(c) You should find that the results from (1) rapidly start losing precision around $n = 30$. Identify the reason for this loss of precision.

(d) Observe that (1) can be rearranged to compute J_{n-1} from J_{n+1}:

$$J_{n-1}(z) = (2n/z)J_n(z) - J_{n+1}(z). \tag{2}$$

Do you believe using (2) to compute $J_0, ..., J_{48}$ from J_{49} and J_{50} will encounter loss of precision? Why?

Solution. (a) See Table 1.

(b) See Table 2.

(c) On Table 2, we see that the accuracy blows up for the larger values of n. As n grows, we see that $J_n(20)$ becomes smaller in magnitude. thus, the recurrence relation is using two larger numbers to compute a smaller number by taking a difference. We showed in the first problem that when $|x| + |y|$ is much bigger than $|x - y|$, subtraction becomes sensitive and prone to error. All of this error continues to propagate as we use the recurrence relation to compute $J_n(z)$ for larger values of n.

(d) See Table 3. Here, since we started with small numbers for $J_{50}(20)$ and $J_{49}(20)$, the actual error starts out small relative to the size of $J_n(20)$ for smaller n values, thus the propagated error is small relative to the values of the Bessel function that we're computing.

□

Table 1: Bessel Recursion Accuracy

n	$J_n(20)$	$J_n(z) - ((2n/z)J_{n-1}(z) - J_{n-2}(z))$	% error
2	0.167024664341	2.77555756156e-17	1.73103040998e-14
3	0.0668331241758	0.0	0.0
4	-0.160341351923	0.0	0.0
5	-0.0989013945604	0.0	0.0
6	0.130670933555	-6.93889390391e-18	1.25964630952e-14
7	0.151169767982	0.0	0.0
8	-0.0550860495637	1.38777878078e-17	1.87870435183e-14
9	-0.184221397721	-2.77555756156e-17	2.21820557913e-14
10	-0.0738689288408	0.0	0.0
11	0.125126254648	6.93889390391e-18	1.13091785556e-14
12	0.186482558024	-2.77555756156e-17	2.33258509034e-14
13	0.061356303376	-2.77555756156e-17	1.35960069907e-14
14	-0.11899062431	0.0	0.0
15	-0.204145052548	-1.8323016715e-17	2.25633726574e-12
16	-0.146397944003	-2.77555756156e-17	1.91180645573e-14
17	-0.000812069055154	2.77555756156e-17	1.19071633614e-14
18	0.14517984042	5.55111512313e-17	2.21080831413e-14
19	0.233099813727	5.55111512313e-17	2.53635513253e-14
20	0.251089842916	0.0	0.0
21	0.218861903522	2.77555756156e-17	2.50878255518e-14
22	0.164747773775	-2.77555756156e-17	4.10689455014e-14
23	0.110633644029	1.38777878078e-17	3.64737607094e-14
24	0.0675828786855	-1.38777878078e-17	6.96357762909e-14
25	0.0380486890792	1.73472347598e-18	1.77353447714e-14
26	0.0199291061966	1.73472347598e-18	3.83465294446e-14
27	0.00978116579257	-8.67361737988e-19	4.37898765054e-14
28	0.00452380828487	-3.25260651746e-19	3.94648436394e-14
29	0.00198073574809	1.62630325873e-19	4.97396255084e-14
30	0.000824178234983	1.89735380185e-19	1.52993447482e-13
31	0.000326963309857	4.06575814682e-20	9.01842801363e-14
32	0.000124015363604	3.38813178902e-20	2.15238955394e-13
33	4.50827809534e-05	4.23516473627e-21	8.00712895667e-14
34	1.57412573519e-05	-1.05879118407e-21	6.18004041909e-14
35	5.2892425727e-06	1.05879118407e-21	1.97615269075e-13
36	1.71324313802e-06	-5.29395592034e-23	3.26784982006e-14
37	5.35784096556e-07	1.19114008208e-22	2.51188206646e-13
38	1.62001199928e-07	2.31610571515e-23	1.72154758836e-13
39	4.74202231856e-08	8.27180612553e-25	2.23347728163e-14
40	1.34536258586e-08	2.06795153138e-24	2.08833590054e-13
41	3.7035550769e-09	1.55096364854e-25	6.02548368075e-14
42	9.90238941374e-10	7.75481824268e-26	1.19114529287e-13
43	2.57400688594e-10	3.23117426779e-27	2.01499862811e-14
44	6.51038818615e-11	-5.65455496862e-27	1.46899655411e-13
45	1.60356152243e-11	-7.06819371078e-28	7.84357303505e-14
46	3.84926360297e-12	4.79627430374e-28	2.32956380393e-13
47	9.01144628754e-13	3.78653234506e-29	8.24295069619e-14
48	2.05887226426e-13	1.4199496294e-29	1.41784359854e-13
49	4.59366128055e-14	3.54987407349e-30	1.66294964838e-13
50	1.001485376e-14	1.38050658414e-30	3.10153763163e-13

Table 2: Estimating $J_{n+1}(z)$ using $J_0(z)$ and $J_1(z)$

n	$J_n(20)$ (`scipy`)	(using recurrence relation)	Relative error
0	0.167024664341	0.167024664341	0.0
1	0.0668331241758	0.0668331241758	0.0
2	-0.160341351923	-0.160341351923	1.73103040998e-16
3	-0.0989013945604	-0.0989013945604	1.40319435024e-16
4	0.130670933555	0.130670933555	2.1240818337e-16
5	0.151169767982	0.151169767982	1.8360533317e-16
6	-0.0550860495637	-0.0550860495637	1.25964630952e-16
7	-0.184221397721	-0.184221397721	1.5066423314e-16
8	-0.0738689288408	-0.0738689288408	3.75740870366e-16
9	0.125126254648	0.125126254648	2.21820557913e-16
10	0.186482558024	0.186482558024	2.97674762828e-16
11	0.061356303376	0.061356303376	3.39275356668e-16
12	-0.11899062431	-0.11899062431	0.0
13	-0.204145052548	-0.204145052548	0.0
14	-0.146397944003	-0.146397944003	0.0
15	-0.000812069055154	-0.000812069055154	2.25633726574e-14
16	0.14517984042	0.14517984042	1.91180645573e-16
17	0.233099813727	0.233099813727	0.0
18	0.251089842916	0.251089842916	2.21080831413e-16
19	0.218861903522	0.218861903522	7.60906539759e-16
20	0.164747773775	0.164747773775	1.68473145218e-15
21	0.110633644029	0.110633644029	3.76317383278e-15
22	0.0675828786855	0.0675828786855	8.62447855529e-15
23	0.0380486890792	0.0380486890792	2.2978469247e-14
24	0.0199291061966	0.0199291061966	7.10284918167e-14
25	0.00978116579257	0.00978116579257	2.5840397332e-13
26	0.00452380828487	0.00452380828487	1.0840563874e-12
27	0.00198073574809	0.00198073574808	5.16063694616e-12
28	0.000824178234983	0.00082417823496	2.75365946494e-11
29	0.000326963309857	0.000326963309804	1.63089434293e-10
30	0.000124015363604	0.000124015363472	1.06394506542e-09
31	4.50827809534e-05	4.50827806109e-05	7.59740964285e-09
32	1.57412573519e-05	1.57412564221e-05	5.90704255823e-08
33	5.2892425727e-06	5.28923993972e-06	4.97799916786e-07
34	1.71324313802e-06	1.71323537901e-06	4.52884120459e-06
35	5.35784096556e-07	5.35760348919e-07	4.43231451601e-05
36	1.62001199928e-07	1.61925842207e-07	0.00046516767109
37	4.74202231856e-08	4.71726830267e-08	0.00522013905183
38	1.34536258586e-08	1.26130849915e-08	0.0624769022028
39	3.7035550769e-09	7.57039941061e-10	0.795591013137
40	9.90238941374e-10	-9.66062922138e-09	10.7558567107
41	2.57400688594e-10	-3.93995568266e-08	154.067021855
42	6.51038818615e-11	-1.51877553768e-07	2333.84943117
43	1.60356152243e-11	-5.98486168997e-07	37323.3079143
44	3.84926360297e-12	-2.42161297292e-06	629111.713814
45	9.01144628754e-13	-1.00566109119e-05	11159819.958
46	2.05887226426e-13	-4.28331361304e-05	208041737.605
47	4.59366128055e-14	-0.000186975815288	4070300440.42
48	1.001485376e-14	-0.000835953195723	83471333258.1
49	2.13468524255e-15	-0.00382559952418	1.79211410091e+12
50	4.4510392847e-16	-0.0179094844728	4.02366353726e+13

Table 3: Estimating $J_n(z)$ using $J_{50}(z)$ and $J_{49}(z)$

n	$J_n(20)$ (`scipy`)	(using recurrence relation)	Relative error
0	0.167024664341	0.167024664341	3.15735366857e-15
1	0.0668331241758	0.0668331241758	1.2458900863e-15
2	-0.160341351923	-0.160341351923	3.11585473797e-15
3	-0.0989013945604	-0.0989013945604	1.82415265531e-15
4	0.130670933555	0.130670933555	3.39853093392e-15
5	0.151169767982	0.151169767982	2.38686933121e-15
6	-0.0550860495637	-0.0550860495637	4.91262060714e-15
7	-0.184221397721	-0.184221397721	2.86262042967e-15
8	-0.0738689288408	-0.0738689288408	1.1272226111e-15
9	0.125126254648	0.125126254648	3.54912892661e-15
10	0.186482558024	0.186482558024	2.53023548403e-15
11	0.061356303376	0.061356303376	5.6545892778e-16
12	-0.11899062431	-0.11899062431	3.84876539906e-15
13	-0.204145052548	-0.204145052548	2.85516146804e-15
14	-0.146397944003	-0.146397944003	2.08548920446e-15
15	-0.000812069055154	-0.000812069055154	1.48330810784e-13
16	0.14517984042	0.14517984042	3.25007097474e-15
17	0.233099813727	0.233099813727	2.85771920674e-15
18	0.251089842916	0.251089842916	2.65296997696e-15
19	0.218861903522	0.218861903522	2.53635513253e-15
20	0.164747773775	0.164747773775	2.35862403306e-15
21	0.110633644029	0.110633644029	2.38334342742e-15
22	0.0675828786855	0.0675828786855	2.25879200258e-15
23	0.0380486890792	0.0380486890792	2.18842564257e-15
24	0.0199291061966	0.0199291061966	2.08907328873e-15
25	0.00978116579257	0.00978116579257	2.12824137257e-15
26	0.00452380828487	0.00452380828487	2.10905911945e-15
27	0.00198073574809	0.00198073574809	1.97054444274e-15
28	0.000824178234983	0.000824178234983	1.97324218197e-15
29	0.000326963309857	0.000326963309857	1.65798751695e-15
30	0.000124015363604	0.000124015363604	1.52993447482e-15
31	4.50827809534e-05	4.50827809534e-05	1.20245706848e-15
32	1.57412573519e-05	1.57412573519e-05	1.29143373237e-15
33	5.2892425727e-06	5.2892425727e-06	1.4412832122e-15
34	1.71324313802e-06	1.71324313802e-06	1.23600808382e-15
35	5.35784096556e-07	5.35784096556e-07	1.38330688352e-15
36	1.62001199928e-07	1.62001199928e-07	9.80354946017e-16
37	4.74202231856e-08	4.74202231856e-08	9.76843025847e-16
38	1.34536258586e-08	1.34536258586e-08	9.83741479063e-16
39	3.7035550769e-09	3.7035550769e-09	8.93390912652e-16
40	9.90238941374e-10	9.90238941374e-10	8.35334360215e-16
41	2.57400688594e-10	2.57400688594e-10	8.033978241e-16
42	6.51038818615e-11	6.51038818615e-11	5.95572646434e-16
43	1.60356152243e-11	1.60356152243e-11	6.04499588433e-16
44	3.84926360297e-12	3.84926360297e-12	6.29569951762e-16
45	9.01144628754e-13	9.01144628754e-13	4.48204173432e-16
46	2.05887226426e-13	2.05887226426e-13	3.67825863779e-16
47	4.59366128055e-14	4.59366128055e-14	2.74765023206e-16
48	1.001485376e-14	1.001485376e-14	1.57538177616e-16
49	2.13468524255e-15	2.13468524255e-15	0.0
50	4.4510392847e-16	4.4510392847e-16	0.0

Unfortunately, I had a difficult time getting used to Python because I have almost no programming experience and ran out of time to complete the rest of this assignment. I'm getting a study group together so that this doesn't happen again.