Krisha Sharma

CSE 13s

Winter 2022 Long

Assignment 2: Numerical Integration WRITEUP

Introduction:

Assignment 2 is focused on the concept of numerical integration; however, it is split into two parts. Our first task in this assignment is to implement a small library of math functions, sin(x), cos(x), e^x, $\sqrt{x}$, and log(x). We are not allowed to use the <math.h> library to help with the first task. Each of the functions written for the first task will also halt computation using $\varepsilon = 10^{14}$. The second part of this assignment is to write a dedicated program, called integrate, that links with my implemented math library and computes numerical integrations of various functions using the composite Simpson's ⅓ rule.

This write-up will first look at the math library functions I made. It will show the difference between the outputs generated by the math library when subtracted from the outputs of the <math.h> library.

The write-up will then look at the way I tested my integrate.c, more specifically, the way I tested my getopt loop. I will show you photos of my generated values and compare them to the final value taken from the table of approximate values in the assignment 2 instruction pdf.

Finally, the write-up will look at the role partitions (or intervals) play in computation accuracy using graphs and plots.

Error handling and citations are also mentioned in this write-up along with the assignment conclusion and takeaways.

Testing my own math library:

Sin(x) v.s sin(x):

```
Sin Difference:
DIFFERENCE = (-0.000000000000387)
DIFFERENCE = (-0.000000000000427)
DIFFERENCE = (-0.000000000000830)
DIFFERENCE = (-0.000000000001525)
DIFFERENCE = (0.000000000003927)
DIFFERENCE = (0.000000000024823)
DIFFERENCE = (-0.000000000065718)
DIFFERENCE = (0.000000000044709)
DIFFERENCE = (0.000000000206630)
DIFFERENCE = (-0.000000000905125)
DIFFERENCE = (0.000000004620359)
DIFFERENCE = (0.000000003832500)
DIFFERENCE = (-0.000000005788987)
DIFFERENCE = (-0.000000025536073)
DIFFERENCE = (0.000000021302848)
DIFFERENCE = (0.000000053020817)
DIFFERENCE = (0.000000488838396)
DIFFERENCE = (-0.000003669047594)
DIFFERENCE = (0.000011091536127)
DIFFERENCE = (0.000030801786859)
```

Explanation: The image above shows the difference between the output calculations of my math library functions and the official math library <math.h> for Sin(x) and sin(x). In order to show these outputs, I created a test file that subtracts the output of my Sin(x) function from the <math.h> library sin(x) function. As you can see the difference between the two starts out small but becomes larger later on.

Cos(x) v.s cos(x):

```
Cos Difference:
DIFFERENCE = (-0.000000000000113)
DIFFERENCE = (0.000000000000121)
DIFFERENCE = (0.000000000000400)
DIFFERENCE = (-0.000000000003027)
DIFFERENCE = (-0.000000000001023)
DIFFERENCE = (0.000000000010979)
DIFFERENCE = (-0.000000000012208)
DIFFERENCE = (-0.000000000008258)
DIFFERENCE = (0.000000000503319)
DIFFERENCE = (-0.000000000020418)
DIFFERENCE = (-0.000000003358351)
DIFFERENCE = (0.000000004426488)
DIFFERENCE = (0.000000020331028)
DIFFERENCE = (0.000000086240764)
DIFFERENCE = (-0.000000146553240)
DIFFERENCE = (-0.000000463007626)
DIFFERENCE = (-0.000000714496369)
DIFFERENCE = (0.000001466311978)
DIFFERENCE = (-0.000003379999122)
DIFFERENCE = (-0.000035073343094)
```

Explanation: The image above shows the difference between the output calculations of my math library functions and the official math library <math.h> for Cos(x) and cos(x). In order to show these outputs, I created a test file that subtracts the output of my Cos(x) function from the <math.h> library cos(x) function. As you can see the difference between the two starts out small but becomes larger as larger numbers are computed.

Exp(x) v.s exp(x):

```
Exponent Difference:
DIFFERENCE = (-0.000000000003638)
DIFFERENCE = (0.000000000014552)
DIFFERENCE = (-0.000000000116415)
DIFFERENCE = (-0.000000000058208)
DIFFERENCE = (0.000000000465661)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (-0.000000001862645)
DIFFERENCE = (0.000000011175871)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (-0.000000029802322)
DIFFERENCE = (0.000000178813934)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000953674316)
DIFFERENCE = (-0.000001907348633)
DIFFERENCE = (-0.000011444091797)
DIFFERENCE = (0.000030517578125)
DIFFERENCE = (-0.000030517578125)
DIFFERENCE = (0.000427246093750)
DIFFERENCE = (-0.000976562500000)
DIFFERENCE = (0.000000000000000)
```

Explanation: The image above shows the difference between the output calculations of my math library functions and the official math library <math.h> for Exp(x) and exp(x). In order to show these outputs, I created a test file that subtracts the output of my Exp(x) function from the <math.h> library exp(x) function. As you can see the difference is very small and at a certain point becomes 0.

Sqrt(x) v.s sqrt(x):

```
Square Root Difference:
DIFFERENCE = (-0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000001)
DIFFERENCE = (-0.000000000000001)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000001)
```

Explanation: The image above shows the difference between the output calculations of my math library functions and the official math library <math.h> for Exp(x) and exp(x). In order to show these outputs, I created a test file that subtracts the output of my Exp(x) function from the <math.h> library exp(x) function. As you can see the difference between the two is 0.

Log(x) v.s log(x):

```
Log Difference:
DIFFERENCE = (-0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000001)
DIFFERENCE = (-0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (-0.000000000000000)
DIFFERENCE = (-0.000000000000000)
DIFFERENCE = (-0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (0.000000000000007)
DIFFERENCE = (0.000000000000002)
DIFFERENCE = (0.000000000000000)
DIFFERENCE = (-0.000000000000000)
DIFFERENCE = (-0.000000000000000)
DIFFERENCE = (-0.000000000000000)
```

Explanation: The image above shows the difference between the output calculations of my math library functions and the official math library <math.h> for Log(x) and log(x). In order to show these outputs, I created a test file that subtracts the output of my Log(x) function from the <math.h> library log(x) function. As you can see the difference is 0.

Testing Integrate.c:

Getopt loop testing:

This section will show the output of each case a-j as generated in my terminal. The values generated in the command line go up to partition 100 and increment by 2 each time. The final value generated at partition 100 should be within 5 decimal places of the value provided in the table given in the assignment 2 instruction pdf.

Integral	Low	High	Value
$\sqrt{1-x^4}$	0	1	0.87401918476405
$1/\log(x)$	2	3	1.118424814549702
e^{-x^2}	-10	10	1.772453850905508
$\sin(x^2)$	$-\pi$	π	1.545303425380133
$\cos(x^2)$	$-\pi$	π	1.131387027213366
$\log(\log(x))$	2	10	3.952914142858876
$\sin(x)/x$	-4π	4π	2.984322451168924
e^{-x}/x	1	10	0.2193797774265986
e^{e^x}	0	1	6.316563839027766
$\sqrt{\sin^2(x)+\cos^2(x)}$	0	π	3.141592653589797

Table of approximated values

The values that are calculated in each of these use the integrated functions from the math library I created during part 1 of this assignment. How close the values are computed depends on the number of intervals selected. In the pictures, the interval amount is 100.

-a: sets the function to integrate $\sqrt{1 - x^4}$

```
krishasharma@atlas:~/cse13s-w22/krvsharm/asgn2$ ./integrate -a -p 0.0 -q 1.0 -n 100
sqrt(1 - x^4) ,0.000000,1.000000,100
2,0.812163891034571
4,0.852988388966857
6,0.862714108378597
8,0.866720323920874
10,0.868814915051592
12,0.870069316562543
14,0.870889853274920
16,0.871460984998172
18,0.871877298507962
20,0.872191782249259
22,0.872436190030983
24,0.872630588931354
26,0.872788217901426
28,0.872918127357466
30,0.873026692206613
32,0.873118518838732
34,0.873197009883946
36,0.873264727547650
38,0.873323634093882
40,0.873375255032606
42,0.873420792324722
44,0.873461204472862
46,0.873497264192597
48,0.873529600606012
50,0.873558730560244
52,0.873585082181643
54,0.873609012804904
56,0.873630822772167
58,0.873650766162204
60,0.873669059211622
62,0.873685886982568
64,0.873701408685138
66,0.873715761958244
68,0.873729066337231
70,0.873741426081444
72,0.873752932494288
74,0.873763665838065
76,0.873773696923108
78,0.873783088433507
80,0.873791896038551
82,0.873800169328883
84,0.873807952608497
86,0.873815285567619
88,0.873822203856656
90,0.873828739577633
92,0.873834921706512
94,0.873840776457364
96,0.873846327597446
98,0.873851596720659
100,0.873856603485616
```

-b: sets the function to integrate $1/log(x)$

```
[krishasharma@atlas:~/cse13s-w22/krvsharm/asgn2$ ./integrate -b -p 2.0 -q 3.0 -n 100
1/log(x) ,2.000000,3.000000,100
2,1.119726823210828
4,1.118531919383142
6,1.118447490672188
8,1.118432184839580
10,1.118427872969802
12,1.118426300202316
14,1.118425620027460
16,1.118425288077487
18,1.118425110762833
20,1.118425009174675
22,1.118424947623066
24,1.118424908584828
26,1.118424882864861
28,1.118424865365316
30,1.118424853126043
32,1.118424844358943
34,1.118424837946436
36,1.118424833168927
38,1.118424829550780
40,1.118424826770223
42,1.118424824605005
44,1.118424822898732
46,1.118424821539471
48,1.118424820445883
50,1.118424819558026
52,1.118424818831160
54,1.118424818231492
56,1.118424817733224
58,1.118424817316461
60,1.118424816965716
62,1.118424816668827
64,1.118424816416170
66,1.118424816200069
68,1.118424816014355
70,1.118424815854043
72,1.118424815715076
74,1.118424815594132
76,1.118424815488479
78,1.118424815395853
80,1.118424815314373
82,1.118424815242469
84,1.118424815178821
86,1.118424815122317
88,1.118424815072017
90,1.118424815027120
92,1.118424814986945
94,1.118424814950909
96,1.118424814918509
98,1.118424814889315
100,1.118424814862953
```

-c: sets the function to integrate e^{-x^2}

```
[krishasharma@atlas:~/cse13s-w22/krvsharm/asgn2$ ./integrate -c -p -10.0 -q 10.0 -n 100
e^(-x^2) ,-10.000000,10.000000,100
2,13.333333333333334
4,3.333333333518505
6,4.444510868171221
8,1.679536360954478
10,2.715508970557557
12,1.387484429801185
14,2.153319009201224
16,1.535257984583655
18,1.933791643785089
20,1.672428604536991
22,1.832163064103619
24,1.738617791720135
26,1.790714680857582
28,1.763074153933371
30,1.777039826274746
32,1.770319615882527
34,1.773399263286717
36,1.772055222037857
38,1.772613837882937
40,1.772392732931804
42,1.772476074877838
44,1.772446158842136
46,1.772456385056269
48,1.772453056228841
50,1.772454088106732
52,1.772453783512982
54,1.772453869130833
56,1.772453846214070
58,1.772453852055011
60,1.772453850637429
62,1.772453850965029
64,1.772453850892941
66,1.772453850908046
68,1.772453850905032
70,1.772453850905604
72,1.772453850905502
74,1.772453850905518
76,1.772453850905515
78,1.772453850905516
80,1.772453850905517
82,1.772453850905516
84,1.772453850905516
86,1.772453850905516
88,1.772453850905516
90,1.772453850905516
92,1.772453850905515
94,1.772453850905516
96,1.772453850905516
98,1.772453850905517
100,1.772453850905516
```

-d: sets the function to integrate $sin(x^2)$

```
[krishasharma@atlas:~/cse13s-w22/krvsharm/asgn2$ ./integrate -d -p -3.14159 -q 3.14159 -n 100
sin(x^2) ,-3.141590,3.141590,100
2,-0.901188478494456
4,2.164336299481835
6,-1.703713511855713
8,0.240882345611053
10,1.538238204657321
12,1.813469164403954
14,1.756593490402679
16,1.673541188532240
18,1.620029549735504
20,1.590064061882550
22,1.573389990620765
24,1.563771265409878
26,1.557958620931474
28,1.554284149317647
30,1.551866520105470
32,1.550220015800047
34,1.549065063023984
36,1.548234093793509
38,1.547622960857268
40,1.547164824564060
42,1.546815559151859
44,1.546545298252873
46,1.546333374990344
48,1.546165203790372
50,1.546030307631778
52,1.545921040123408
54,1.545831739843113
56,1.545758159816428
58,1.545697075801815
60,1.545646013001399
62,1.545603052576201
64,1.545566692801372
66,1.545535748184405
68,1.545509275317236
70,1.545486517792206
72,1.545466864872134
74,1.545449820189910
76,1.545434977837954
78,1.545422003953268
80,1.545410622425956
82,1.545400603727760
84,1.545391756120301
86,1.545383918691295
88,1.545376955804869
90,1.545370752653699
92,1.545365211674581
94,1.545360249644540
96,1.545355795316845
98,1.545351787487476
100,1.545348173406214
```

-e: sets the function to integrate $cos(x^2)$

```
krishasharma@atlas:~/cse13s-w22/krvsharm/asgn2$ ./integrate -e -p -3.14159 -q 3.14159 -n 100
cos(x^2) ,-3.141590,3.141590,100
2,2.298193436767280
4,-3.170419006148042
6,0.509531245309856
8,2.499810087146109
10,2.125732055829642
12,1.557553752857736
14,1.275870085724610
16,1.173822504143681
18,1.141530721850013
20,1.132066413671838
22,1.129613677720505
24,1.129245687873320
26,1.129452871884962
28,1.129775377313934
30,1.130079421732041
32,1.130333424263712
34,1.130536936012547
36,1.130697704481669
38,1.130824401878318
40,1.130924551798392
42,1.131004162507227
44,1.131067877621029
46,1.131119241723426
48,1.131160952768093
50,1.131195068465235
52,1.131223165452247
54,1.131246458970838
56,1.131265891919374
58,1.131282200937633
60,1.131295965519964
62,1.131307644656451
64,1.131317604311681
66,1.131326138155173
68,1.131333483299535
70,1.131339832326700
72,1.131345342536878
74,1.131350143107414
76,1.131354340668701
78,1.131358023673642
80,1.131361265841439
82,1.131364128888130
84,1.131366664702569
86,1.131368917090232
88,1.131370923177407
90,1.131372714547878
92,1.131374318166659
94,1.131375757134817
96,1.131377051308601
98,1.131378217808911
100,1.131379271443455
```

-f: sets the function to integrate $log(log(x))$

```
[krishasharma@atlas:~/cse13s-w22/krvsharm/asgn2$ ./integrate -f -p 2.0 -q 10.0 -n 100
log(log(x)) ,2.000000,10.000000,100
2,3.733749130395906
4,3.912566798993677
6,3.940066422561948
8,3.947604503051998
10,3.950342756791001
12,3.951526183455259
14,3.952102926711472
16,3.952410135404280
18,3.952585435402779
20,3.952691128620414
22,3.952757797155897
24,3.952801467254214
26,3.952831003913765
28,3.952851539362861
30,3.952866162804845
32,3.952876797457523
34,3.952884676476137
36,3.952890611400720
38,3.952895148812154
40,3.952898664547326
42,3.952901421906437
44,3.952903608494219
46,3.952905360052997
48,3.952906776189734
50,3.952907930943780
52,3.952908880006709
54,3.952909665730844
56,3.952910320653025
58,3.952910870006367
60,3.952911333532514
62,3.952911726804004
64,3.952912062199392
66,3.952912349629478
68,3.952912597083132
70,3.952912811041155
72,3.952912996792636
74,3.952913158678655
76,3.952913300281363
78,3.952913424571759
80,3.952913534025855
82,3.952913630716640
84,3.952913716387259
86,3.952913792509599
88,3.952913860331432
90,3.952913920914545
92,3.952913975165732
94,3.952914023862094
96,3.952914067671805
98,3.952914107171190
100,3.952914142858881
```

-g: sets the function to integrate $sin(x)/x$

```
[krishasharma@atlas:~/cse13s-w22/krvsharm/asgn2$ ./integrate -g -p -12.56 -q 12.56 -n 100
sin(x)/x ,-12.560000,12.560000,100
2,-nan
4,-nan
6,5.585045795861373
8,-nan
10,-nan
12,2.955113033662460
14,-nan
16,-nan
18,-nan
20,-nan
22,-nan
24,2.983167352805697
26,-nan
28,-nan
30,-nan
32,-nan
34,-nan
36,-nan
38,2.984154865183168
40,-nan
42,-nan
44,-nan
46,2.984247358470287
48,2.984259797378912
50,2.984269856276262
52,-nan
54,-nan
56,-nan
58,-nan
60,-nan
62,-nan
64,-nan
66,-nan
68,-nan
70,-nan
72,-nan
74,-nan
76,2.984315387431409
78,-nan
80,-nan
82,-nan
84,-nan
86,2.984319419709648
88,-nan
90,2.984320465176303
92,2.984320905974562
94,-nan
96,2.984321656256559
98,-nan
100,2.984322264736323
```

-h: sets the function to integrate e^{-x}/x

```
[krishasharma@atlas:~/cse13s-w22/krvsharm/asgn2$ ./integrate -h -p 1.0 -q 10.0 -n 100
e^(-x)/x ,1.000000,10.000000,100
2,0.556284267861316
4,0.312985875410002
6,0.255853141670985
8,0.236448956291270
10,0.228383162621719
12,0.224552251760750
14,0.222549605421031
16,0.221423546736741
18,0.220752845439824
20,0.220334210727354
22,0.220062536786067
24,0.219880330813143
26,0.219754627016734
28,0.219665750423167
30,0.219601545676045
32,0.219554273387523
34,0.219518873331517
36,0.219491958099266
38,0.219471211791168
40,0.219455020747126
42,0.219442241132959
44,0.219432049381755
46,0.219423843995733
48,0.219417179876214
50,0.219411723662118
52,0.219407222882853
54,0.219403484346454
56,0.219400358796688
58,0.219397729886238
60,0.219395506160778
62,0.219393615169341
64,0.219391999093568
66,0.219390611473641
68,0.219389414734077
70,0.219388378298429
72,0.219387477141466
74,0.219386690669093
76,0.219386001845747
78,0.219385396510090
80,0.219384862834976
82,0.219384390898743
84,0.219383972342933
86,0.219383600097541
88,0.219383268159330
90,0.219382971412082
92,0.219382705480162
94,0.219382466608688
96,0.219382251565050
98,0.219382057557633
100,0.219381882168480
```

-i: sets the function to integrate e^{e^x}

```
[krishasharma@atlas:~/cse13s-w22/krvsharm/asgn2$ ./integrate -i -p 0.0 -q 1.0 -n 100
e^e^x ,0.000000,1.000000,100
2,6.445641188183025
4,6.328542481322285
6,6.319177751524516
8,6.317423230998102
10,6.316922418702334
12,6.316738549801645
14,6.316658736577262
16,6.316619694821807
18,6.316598808228739
20,6.316586828929244
22,6.316579565099837
24,6.316574955433230
26,6.316571917038206
28,6.316569849025224
30,6.316568402239443
32,6.316567365657411
34,6.316566607329294
36,6.316566042263693
38,6.316565614266671
40,6.316565285312762
42,6.316565029132143
44,6.316564827235216
46,6.316564666387459
48,6.316564536969907
50,6.316564431893226
52,6.316564345865255
54,6.316564274888753
56,6.316564215911642
58,6.316564166580026
60,6.316564125061491
62,6.316564089917176
64,6.316564060008069
66,6.316564034425669
68,6.316564012440203
70,6.316563993461537
72,6.316563977009454
74,6.316563962690892
76,6.316563950182291
78,6.316563939215935
80,6.316563929569170
82,6.316563921055971
84,6.316563913520188
86,6.316563906830204
88,6.316563900874624
90,6.316563895558795
92,6.316563890801996
94,6.316563886535141
96,6.316563882698912
98,6.316563879242185
100,6.316563876120799
```

-j: sets the function to integrate $\sqrt{sin^2(x) + cos^2(x)}$

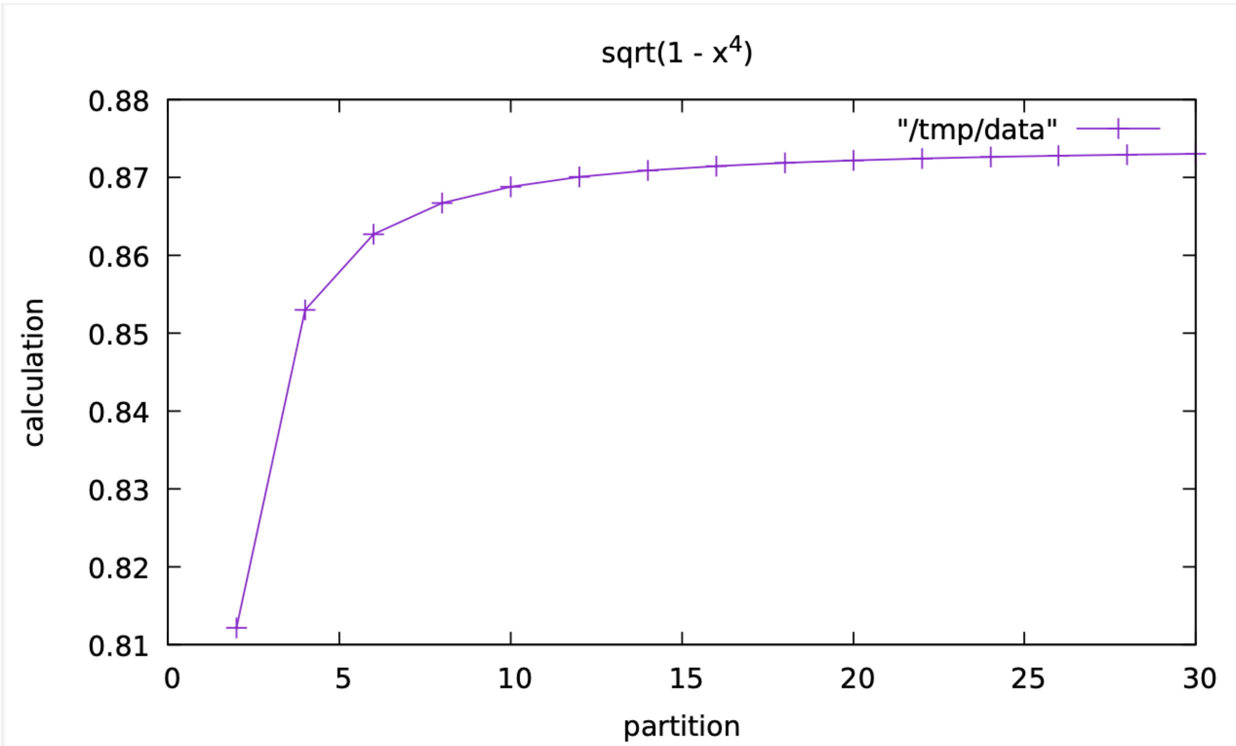
```
[krishasharma@atlas:~/cse13s-w22/krvsharm/asgn2$ ./integrate -j -p 0.0 -q 3.14159 -n 100
sqrt(sin^2(x) + cos^2(x) ,0.000000,3.141590,100
2,3.14159000000000
4,3.14159000000000
6,3.14159000000000
8,3.14159000000000
10,3.14159000000000
12,3.14159000000000
14,3.14159000000000
16,3.14159000000000
18,3.14159000000000
20,3.14159000000000
22,3.14159000000000
24,3.14159000000000
26,3.14159000000000
28,3.14159000000000
30,3.14159000000000
32,3.14159000000000
34,3.14159000000000
36,3.14159000000000
38,3.14159000000000
40,3.14159000000000
42,3.14159000000000
44,3.14159000000000
46,3.14159000000000
48,3.14159000000000
50,3.14159000000000
52,3.14159000000000
54,3.14159000000000
56,3.14159000000000
58,3.14159000000000
60,3.14159000000000
62,3.14159000000000
64,3.14159000000000
66,3.14159000000000
68,3.14159000000000
70,3.14159000000000
72,3.14159000000000
74,3.14159000000000
76,3.14159000000000
78,3.14159000000000
80,3.14159000000000
82,3.14159000000000
84,3.14159000000000
86,3.14159000000000
88,3.14159000000000
90,3.14159000000000
92,3.14159000000000
94,3.14159000000000
96,3.14159000000000
98,3.14159000000000
100,3.14159000000000
```

As you can see, each of the outputs generated from my own math library shows very little difference when compared to the outputs generated by the <math.h> function library.

Furthermore, if you look at the values generated at partition 100 for each of the cases, you can see that the values come extremely close to the values given in the table of approximate values, in fact, most of the values generated come within 5 decimal places of the table values. There are instances; however, where the values generated, even at an interval of 100, do not match the value given in the table of approximate value. In case -a, for example, the value computed, at an interval of 100, is 0.8738566 whereas the value given in the table is 0.8740191 instead. On the other hand when case b is called, -b, the value computed at an interval of 100 is 1.1184248 which matches the first 5 digits of the corresponding value in the table given. As you go through the values computed for each cause you can see that some match but others are slightly off. The values generated for -c matches perfectly at 100 intervals; however, -d is slightly off, with the generated value being 1.545348 rather than 1.545303. The same can be said for cases -e (the value computed is 1.131379 instead of 1.131387) and -h (the value computed is 0.219381 instead of 0.219379). The cases -a, -d, -e, -h, have values that compute slightly off when compared to the given table values. The cases -b, -c, -f, -g, -i, -j all generate values that match the corresponding values given in the table

Partitions:

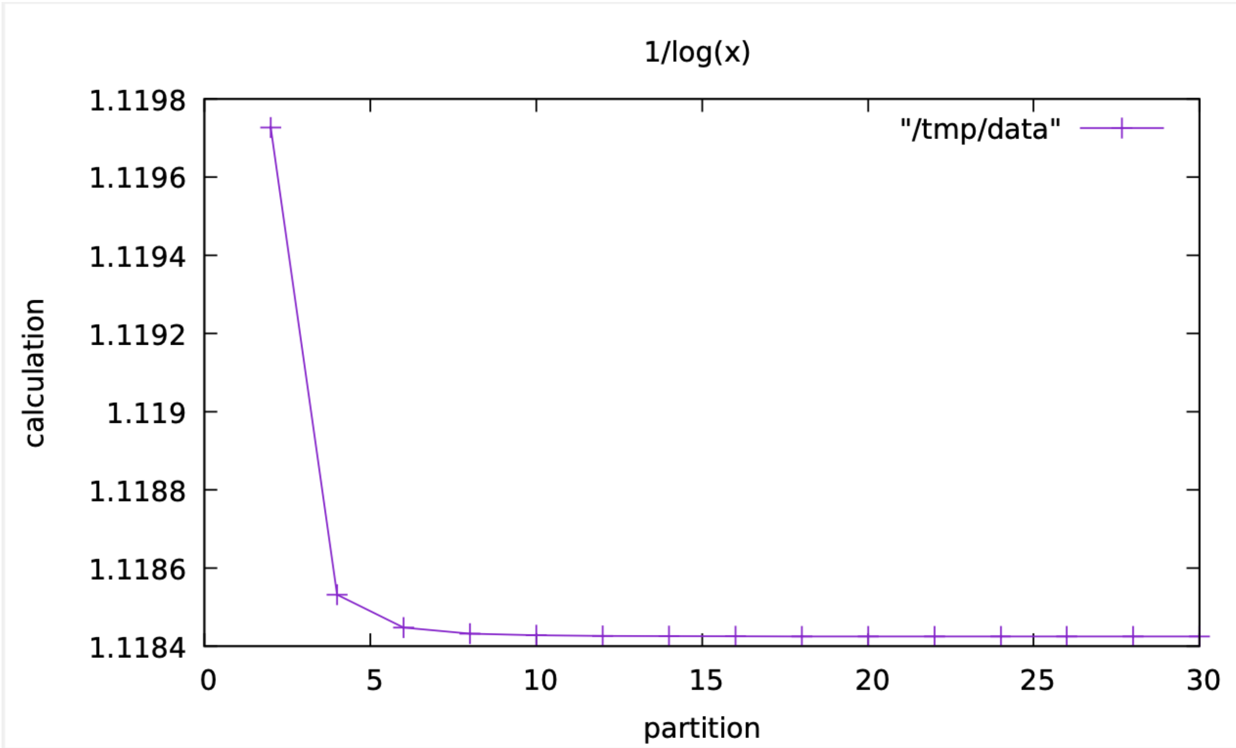
- This plot is the graphical representation of the function sqrt $(1 - x^4)$.



sqrt $(1 - x^4)$ *figure* 1

- When looking at this graph, you can clearly see that as the amount of partitions increases, the accuracy of the final calculation increases as well, leading to the conclusion that the number of partitions directly correlates with how accurate the final calculation is. When the number of partitions is low, the final calculation has a more rough answer; however, when the number of partitions is high, say around 30, then the accuracy of the calculation is also higher. When you look at the graph you can see that when the partition number nears 20, the final calculation number steadies out and remains mostly level.

- This plot is the graphical representation of the function $1/log(x)$.



$1/log(x)$ figure 2

- Similar to the previous graph, sqrt $(1 - x^4)$, this graph represents a log function, specifically $1/log(x)$. Looking at this graph, you would also reach the same conclusion that the previous graph came to as well, that the number of partitions is directly related to the result of the final calculation. As you increase the number of partitions on a function, the accuracy of the final calculation also increases. In this graph, the final calculation should be close to 1.118424814549702, and you can see that as the partition number nears 10-15, the final calculation number nears 1.1184 as well.
- Both the square root function graph and the log function graph reach the same conclusion that as the number of partitions increases, the final calculation number becomes more accurate as well.

Error Handling:

- I ran into an error while programming my math function for cos. Rather than have the term initially set to 1.0, I had my term set to x. When I was testing my code this was causing my cos function values to have a large difference when compared to the cos values I got from the actual math library (math.h). I realized that I needed to change where I initialized "term" as x to 1.0 because when looking at the equation for cos, the equation begins at 1.0 rather than at x. When I did this the difference between the values outputted from my cos function and the actual math library cos function became very small.
- As I was programming my square root and log functions, I ran into an error where my log function was stuck in an infinite loop because I had not yet scaled my log function code. After implementing scaling the error went away and the difference became small.
- While testing my integrate function and getopt loop, I ran into an error. When I called ./integrate -a -p 0.0 -q 1.0 -n 10, everything would work as expected; however, when I called ./integrate -b -p 2.0 -q 3.0 -n 10, the program would output incorrect values. Because my previous test worked as expected, I knew that my issue was somewhere in my log(x) function. When I went back to look over it in mathlib.c I realized that rather than adding the offset back into my sum I was multiplying it in. Once I fixed the error my values were generated correctly.

Citations:

- The code for my math library is based on the python pseudocode provided by the professor in the assignment 2 document. I converted the code from python to C in order to make my own math library. The code for my composite simpson's ⅓ rule is also based on the example python code that the professor provided in the assignment document.

- I collaborated with my sister Twisha Sharma on this assignment, specifically, there was high-level pseudocode collaboration on the math library functions.
- I attended Eugene's section on the 14th of January, 2022, and during it, he showed me how to create my Makefile as well as how to make a getopt loop when needed.
- I attended Brian's tutoring section on the 17th of January, 2022, at 12 pm. The tutoring section was one on one and during it, he showed me how to get started on part two of assignment 2, specifically the integrate function and the way it incorporates composite simpson's ⅓ rule. He also went over how to link files in my Makefile and the way that files such as functions.c, integrate.c, and mathlib.c will need to first be converted into object files and then linked together afterward. Brian also helped me understand how to scale my log and square root functions and how an offset is added or multiplied to the end sum to account for larger numbers.
- I attended Mile's section on the 18th of January, 2022, at 9 am, and during it, he helped me figure out how to get started on my getopt loop.
- I attended Ben's section on the 18th of January, 2022, and during it, he helped me understand how to structure my cases and the way the getopt loop functions to link together multiple arguments.
- Most of the code for my getopt loop was taught to me by Eugene. In his section video on the 14th of January, he showed me how to create the getopt loop as well as other hints, tricks, and strategies I would need to combine to successfully complete part 2 of assignment 2.

- I attended Brian’s section on the 19th of January, 2022, and during it, he helped me solve an error I encountered where my actual integrate(a, low, high, partition) command was not running due to the fact that I declared low and high as integers rather than doubles, and used the atoi command instead of strtod. Brain pointed out that I needed to specify what type of elements the low, high, and partition were when I was trying to print out the first line of my output statement. He told me that I needed to use %f to show that low and high were doubles and %d to show that partition was an integer when I was writing out the print statement that specified what type of operation was being performed. After Brian pointed this out I was able to fix my error and my print statement now looks like printf(“sqrt(1-x^4) , %f,%f,%d\n”, low, high, partition).
- I attended Audrey’s section on the 19th of January, 2022 at 4 pm, and during it, she answered my question on to test π and $-\pi$ on the command line. She told me that I should test that part of my integrated function by approximating π as 3.14 or -3.14.

Summary:

This write-up looked at the math library functions I made and showed the difference between the outputs generated by the math library when subtracted from the outputs of the <math.h> library. It also looked at the way I tested my integrate.c during this assignment. More specifically it examined the way I tested my getopt loop. The write-up showed you photos of my generated values and compare them to the final value taken from the table of approximate values in the assignment 2 instruction pdf. The write-up also looked at the role partitions (or intervals) play in computation accuracy using graphs and plots. It reached the conclusion that the more partitions or intervals you allow the program to have, the more accurate the value computed will be.