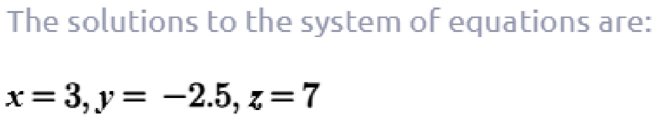
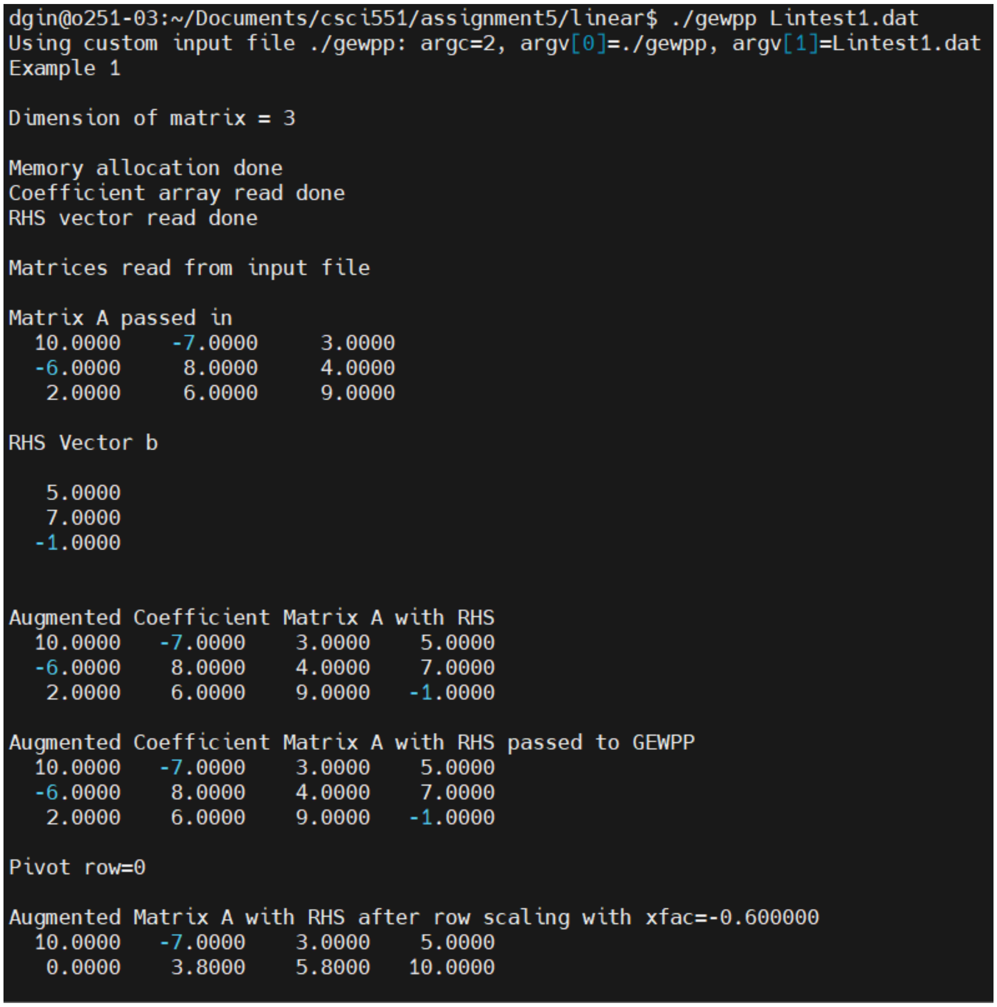
1) Code in linear folder



The solutions to the system of equations are:

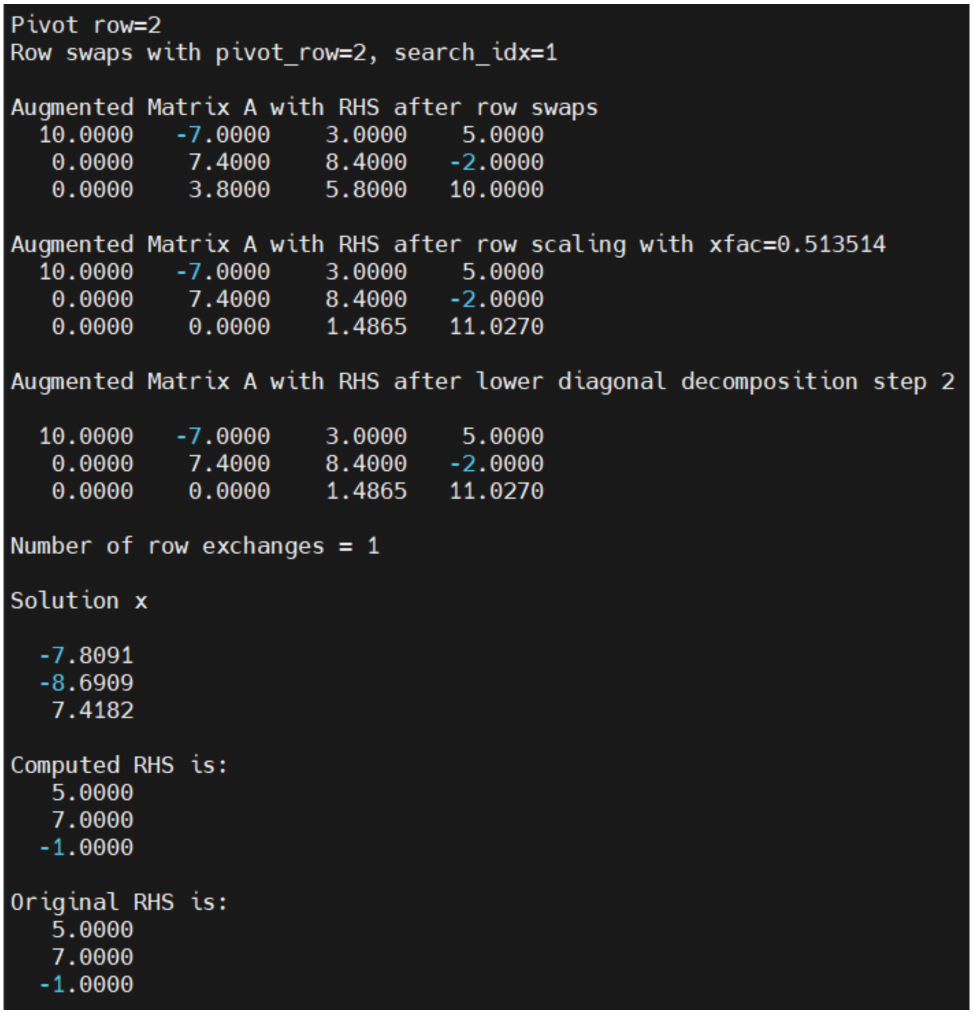
$$x=3, y=-2.5, z=7$$

Symbolab substitution answer^



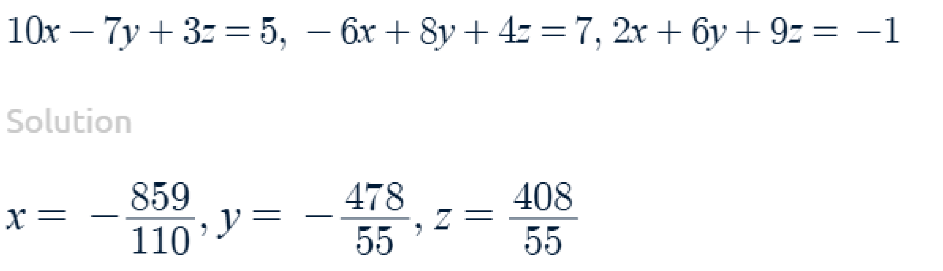
```
dgin@o251-03:~/Documents/csci551/assignment5/linear$ ./gewpp Lintest1.dat
Using custom input file ./gewpp: argc=2, argv[0]=./gewpp, argv[1]=Lintest1.dat
Example 1

Dimension of matrix = 3

Memory allocation done
Coefficient array read done
RHS vector read done

Matrices read from input file

Matrix A passed in
  10.0000    -7.0000     3.0000
  -6.0000     8.0000     4.0000
   2.0000     6.0000     9.0000

RHS Vector b

   5.0000
   7.0000
  -1.0000


Augmented Coefficient Matrix A with RHS
  10.0000   -7.0000    3.0000    5.0000
  -6.0000    8.0000    4.0000    7.0000
   2.0000    6.0000    9.0000   -1.0000

Augmented Coefficient Matrix A with RHS passed to GEWPP
  10.0000   -7.0000    3.0000    5.0000
  -6.0000    8.0000    4.0000    7.0000
   2.0000    6.0000    9.0000   -1.0000

Pivot row=0

Augmented Matrix A with RHS after row scaling with xfac=-0.600000
  10.0000   -7.0000    3.0000    5.0000
   0.0000    3.8000    5.8000   10.0000
```

gewpp.c on Lintest1.dat^



```
Pivot row=2
Row swaps with pivot_row=2, search_idx=1

Augmented Matrix A with RHS after row swaps
  10.0000   -7.0000    3.0000    5.0000
   0.0000    7.4000    8.4000   -2.0000
   0.0000    3.8000    5.8000   10.0000

Augmented Matrix A with RHS after row scaling with xfac=0.513514
  10.0000   -7.0000    3.0000    5.0000
   0.0000    7.4000    8.4000   -2.0000
   0.0000    0.0000    1.4865   11.0270

Augmented Matrix A with RHS after lower diagonal decomposition step 2

  10.0000   -7.0000    3.0000    5.0000
   0.0000    7.4000    8.4000   -2.0000
   0.0000    0.0000    1.4865   11.0270

Number of row exchanges = 1

Solution x

  -7.8091
  -8.6909
   7.4182

Computed RHS is:
   5.0000
   7.0000
  -1.0000

Original RHS is:
   5.0000
   7.0000
  -1.0000
```

gewpp.c on Lintest1.dat continued & answer^

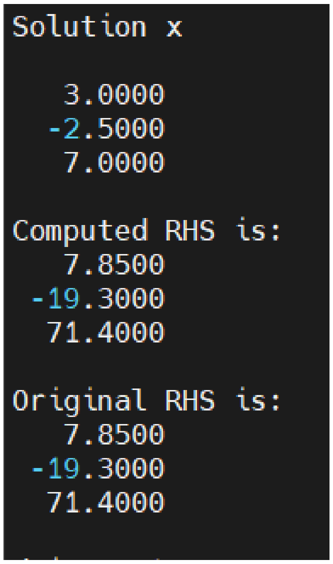


$$10x-7y+3z=5,\ -6x+8y+4z=7,\ 2x+6y+9z=\ -1$$

Solution

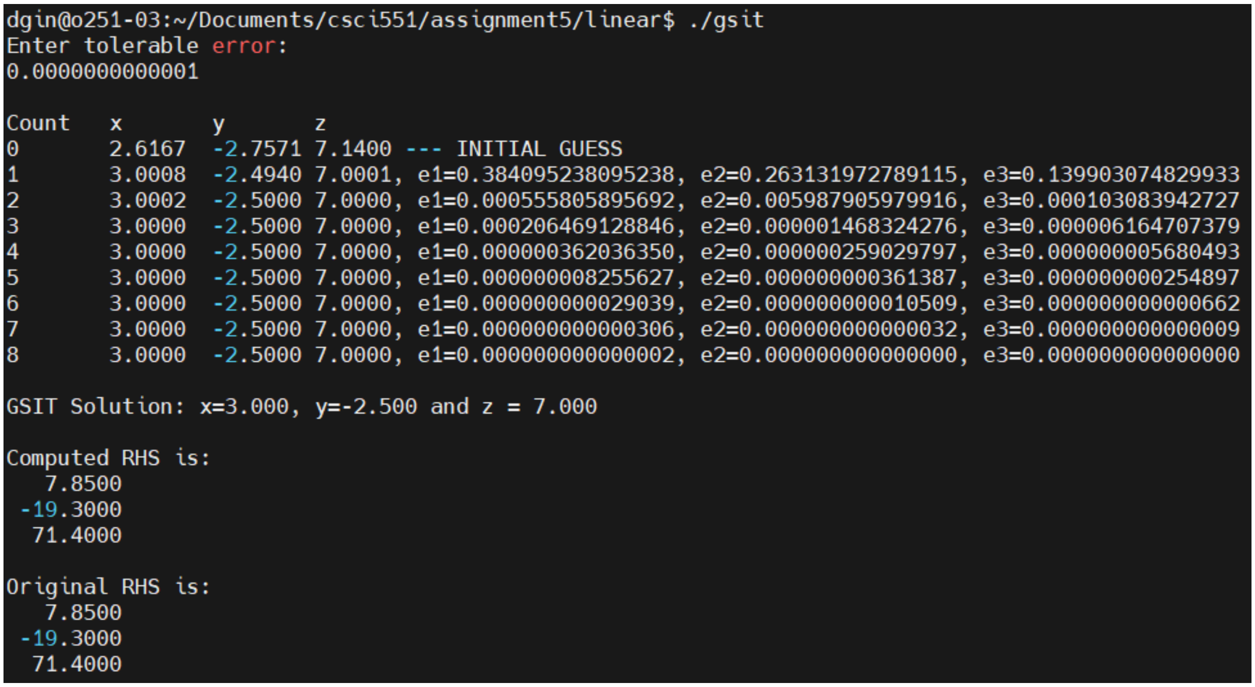
$$x=\ -\frac{859}{110},y=\ -\frac{478}{55},z=\frac{408}{55}$$

Symbolab confirmation on Lintest1.dat numbers^



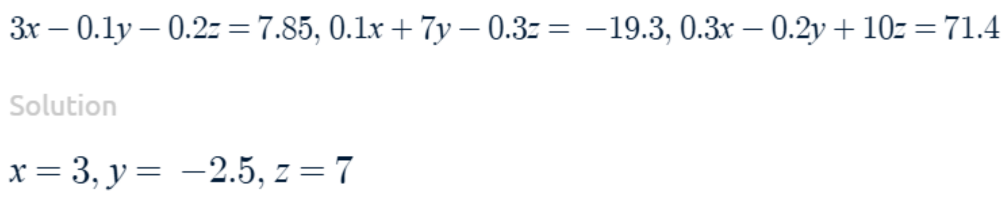
```
Solution x

   3.0000
  -2.5000
   7.0000

Computed RHS is:
   7.8500
 -19.3000
  71.4000

Original RHS is:
   7.8500
 -19.3000
  71.4000
```

gewpp.c results on Lintest4.dat^



```
dgin@o251-03:~/Documents/csci551/assignment5/linear$ ./gsit
Enter tolerable error:
0.0000000000001

Count   x       y       z
0       2.6167  -2.7571 7.1400 --- INITIAL GUESS
1       3.0008  -2.4940 7.0001, e1=0.384095238095238, e2=0.263131972789115, e3=0.139903074829933
2       3.0002  -2.5000 7.0000, e1=0.000555805895692, e2=0.005987905979916, e3=0.000103083942727
3       3.0000  -2.5000 7.0000, e1=0.000206469128846, e2=0.000001468324276, e3=0.000006164707379
4       3.0000  -2.5000 7.0000, e1=0.000000362036350, e2=0.000000259029797, e3=0.000000005680493
5       3.0000  -2.5000 7.0000, e1=0.000000008255627, e2=0.000000000361387, e3=0.000000000254897
6       3.0000  -2.5000 7.0000, e1=0.000000000029039, e2=0.000000000010509, e3=0.000000000000662
7       3.0000  -2.5000 7.0000, e1=0.000000000000306, e2=0.000000000000032, e3=0.000000000000009
8       3.0000  -2.5000 7.0000, e1=0.000000000000002, e2=0.000000000000000, e3=0.000000000000000

GSIT Solution: x=3.000, y=-2.500 and z = 7.000

Computed RHS is:
   7.8500
 -19.3000
  71.4000

Original RHS is:
   7.8500
 -19.3000
  71.4000
```

gsit.c output on Lintest4.dat^



$3x-0.1y-0.2z=7.85,\ 0.1x+7y-0.3z=-19.3,\ 0.3x-0.2y+10z=71.4$

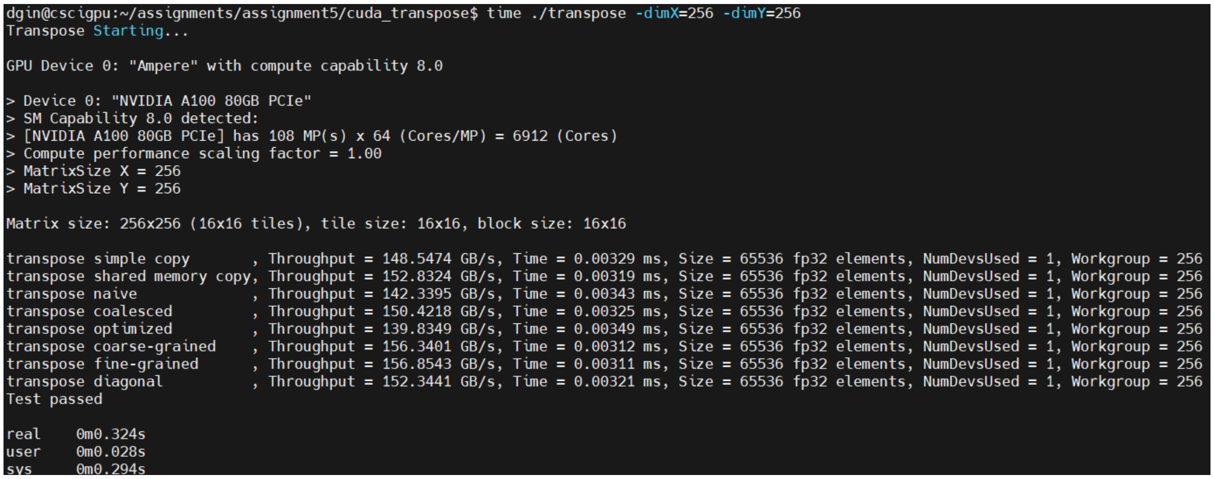
Solution

$x=3,\ y=-2.5,\ z=7$

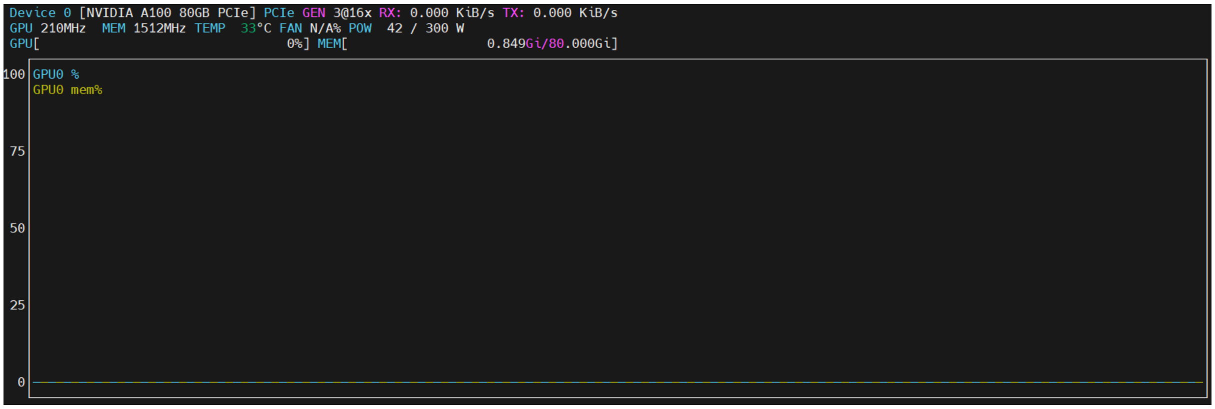
Symbolab confirmation on Lintest4.dat^

I prefer gewpp.c simply because it runs faster, but I do like the iterative guess process of gsit.c.

2) **Code in cude_transpose and hello_cluster folders**

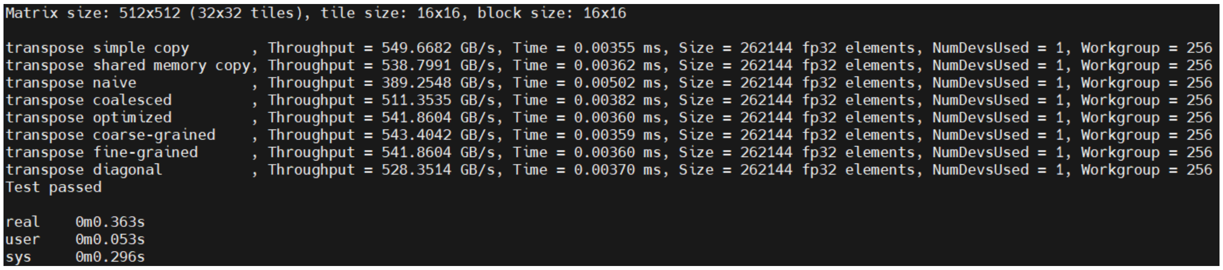


```
dgin@cscigpu:~/assignments/assignment5/cuda_transpose$ time ./transpose -dimX=256 -dimY=256
Transpose Starting...

GPU Device 0: "Ampere" with compute capability 8.0

> Device 0: "NVIDIA A100 80GB PCIe"
> SM Capability 8.0 detected:
> [NVIDIA A100 80GB PCIe] has 108 MP(s) x 64 (Cores/MP) = 6912 (Cores)
> Compute performance scaling factor = 1.00
> MatrixSize X = 256
> MatrixSize Y = 256

Matrix size: 256x256 (16x16 tiles), tile size: 16x16, block size: 16x16

transpose simple copy       , Throughput = 148.5474 GB/s, Time = 0.00329 ms, Size = 65536 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose shared memory copy, Throughput = 152.8324 GB/s, Time = 0.00319 ms, Size = 65536 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose naive             , Throughput = 142.3395 GB/s, Time = 0.00343 ms, Size = 65536 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose coalesced         , Throughput = 150.4218 GB/s, Time = 0.00325 ms, Size = 65536 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose optimized         , Throughput = 139.8349 GB/s, Time = 0.00349 ms, Size = 65536 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose coarse-grained    , Throughput = 156.3401 GB/s, Time = 0.00312 ms, Size = 65536 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose fine-grained      , Throughput = 156.8543 GB/s, Time = 0.00311 ms, Size = 65536 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose diagonal          , Throughput = 152.3441 GB/s, Time = 0.00321 ms, Size = 65536 fp32 elements, NumDevsUsed = 1, Workgroup = 256
Test passed

real    0m0.324s
user    0m0.028s
sys     0m0.294s
```

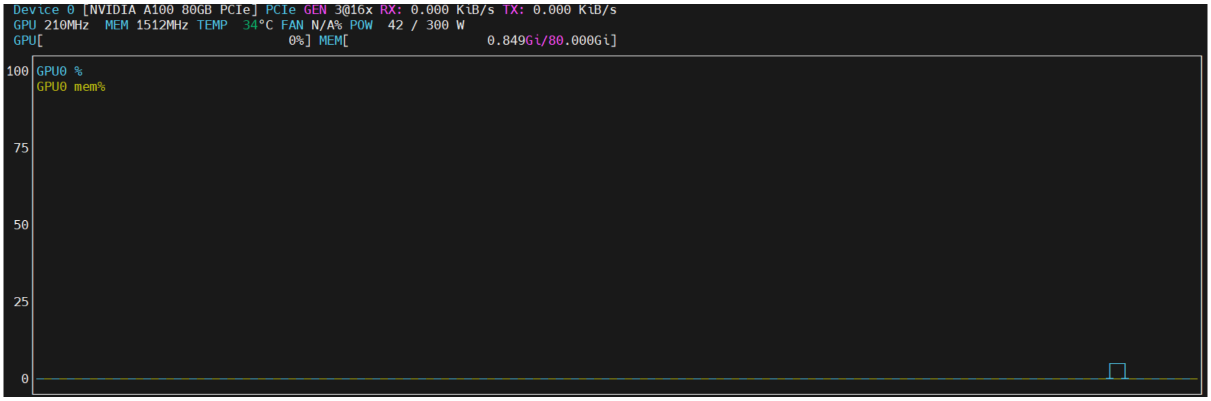


```
Device 0 [NVIDIA A100 80GB PCIe] PCIe GEN 3@16x RX: 0.000 KiB/s TX: 0.000 KiB/s
GPU 210MHz  MEM 1512MHz TEMP  33°C FAN N/A% POW  42 / 300 W
GPU[                                0%] MEM[                   0.849Gi/80.000Gi]
```

Run transpose.c on 256x256 matrix^

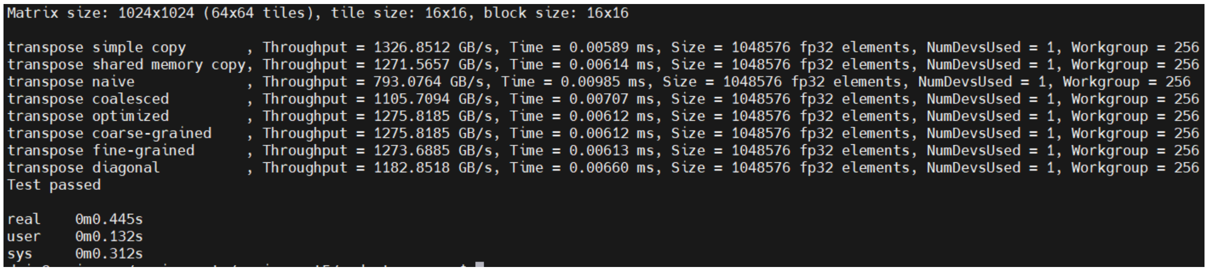


```
Matrix size: 512x512 (32x32 tiles), tile size: 16x16, block size: 16x16

transpose simple copy       , Throughput = 549.6682 GB/s, Time = 0.00355 ms, Size = 262144 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose shared memory copy, Throughput = 538.7991 GB/s, Time = 0.00362 ms, Size = 262144 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose naive             , Throughput = 389.2548 GB/s, Time = 0.00502 ms, Size = 262144 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose coalesced         , Throughput = 511.3535 GB/s, Time = 0.00382 ms, Size = 262144 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose optimized         , Throughput = 541.8604 GB/s, Time = 0.00360 ms, Size = 262144 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose coarse-grained    , Throughput = 543.4042 GB/s, Time = 0.00359 ms, Size = 262144 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose fine-grained      , Throughput = 541.8604 GB/s, Time = 0.00360 ms, Size = 262144 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose diagonal          , Throughput = 528.3514 GB/s, Time = 0.00370 ms, Size = 262144 fp32 elements, NumDevsUsed = 1, Workgroup = 256
Test passed

real	0m0.363s
user	0m0.053s
sys	0m0.296s
```

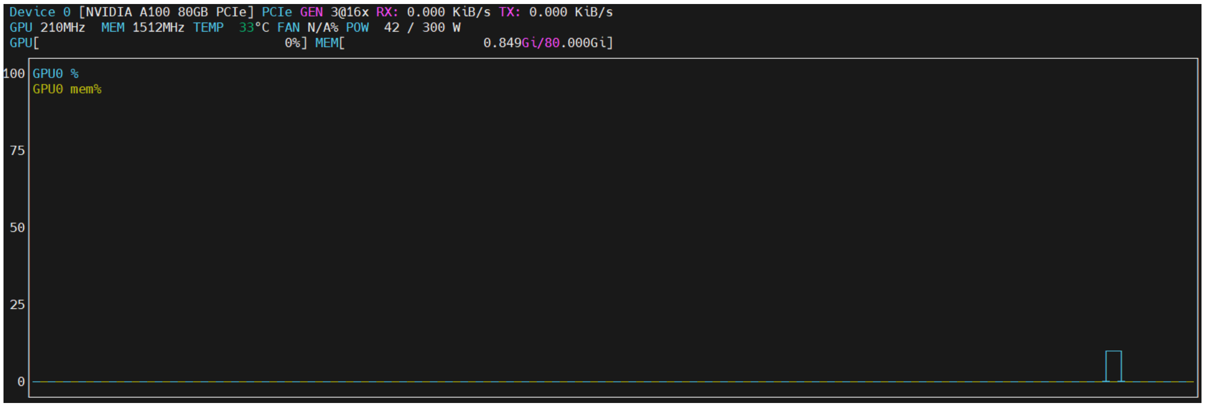


```
 Device 0 [NVIDIA A100 80GB PCIe] PCIe GEN 3@16x RX: 0.000 KiB/s TX: 0.000 KiB/s
 GPU 210MHz  MEM 1512MHz TEMP  34°C FAN N/A% POW  42 / 300 W
 GPU[                                0%] MEM[                 0.849Gi/80.000Gi]
```

Output of 512x512 matrix^

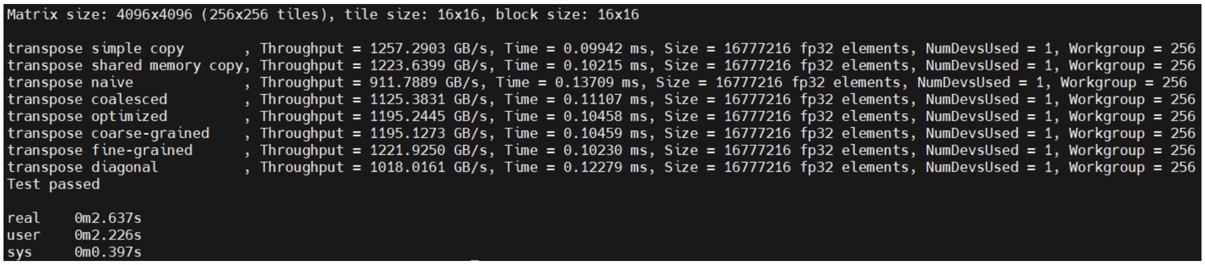


```
Matrix size: 1024x1024 (64x64 tiles), tile size: 16x16, block size: 16x16

transpose simple copy       , Throughput = 1326.8512 GB/s, Time = 0.00589 ms, Size = 1048576 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose shared memory copy, Throughput = 1271.5657 GB/s, Time = 0.00614 ms, Size = 1048576 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose naive             , Throughput = 793.0764 GB/s, Time = 0.00985 ms, Size = 1048576 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose coalesced         , Throughput = 1105.7094 GB/s, Time = 0.00707 ms, Size = 1048576 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose optimized         , Throughput = 1275.8185 GB/s, Time = 0.00612 ms, Size = 1048576 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose coarse-grained    , Throughput = 1275.8185 GB/s, Time = 0.00612 ms, Size = 1048576 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose fine-grained      , Throughput = 1273.6885 GB/s, Time = 0.00613 ms, Size = 1048576 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose diagonal          , Throughput = 1182.8518 GB/s, Time = 0.00660 ms, Size = 1048576 fp32 elements, NumDevsUsed = 1, Workgroup = 256
Test passed

real	0m0.445s
user	0m0.132s
sys	0m0.312s
```

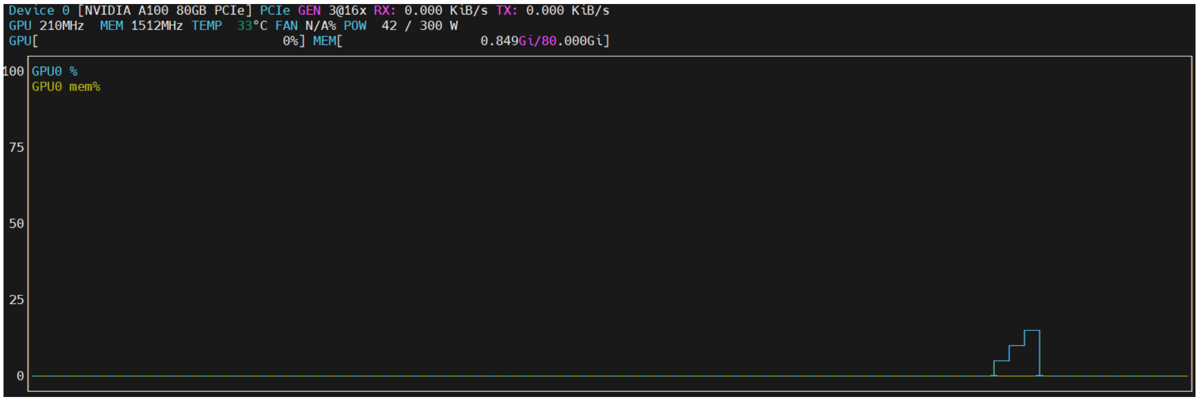


```
 Device 0 [NVIDIA A100 80GB PCIe] PCIe GEN 3@16x RX: 0.000 KiB/s TX: 0.000 KiB/s
 GPU 210MHz  MEM 1512MHz TEMP  33°C FAN N/A% POW  42 / 300 W
 GPU[                                0%] MEM[                 0.849Gi/80.000Gi]
```

Output of 1024x1024 matrix^

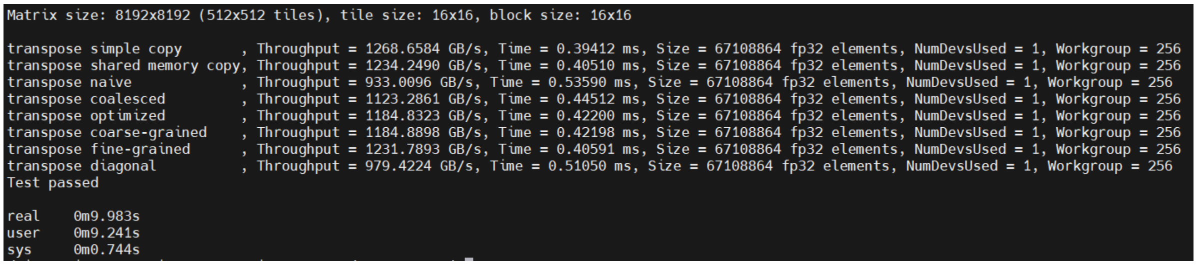


```
Matrix size: 4096x4096 (256x256 tiles), tile size: 16x16, block size: 16x16

transpose simple copy       , Throughput = 1257.2903 GB/s, Time = 0.09942 ms, Size = 16777216 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose shared memory copy, Throughput = 1223.6399 GB/s, Time = 0.10215 ms, Size = 16777216 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose naive             , Throughput = 911.7889 GB/s, Time = 0.13709 ms, Size = 16777216 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose coalesced         , Throughput = 1125.3831 GB/s, Time = 0.11107 ms, Size = 16777216 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose optimized         , Throughput = 1195.2445 GB/s, Time = 0.10458 ms, Size = 16777216 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose coarse-grained    , Throughput = 1195.1273 GB/s, Time = 0.10459 ms, Size = 16777216 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose fine-grained      , Throughput = 1221.9250 GB/s, Time = 0.10230 ms, Size = 16777216 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose diagonal          , Throughput = 1018.0161 GB/s, Time = 0.12279 ms, Size = 16777216 fp32 elements, NumDevsUsed = 1, Workgroup = 256
Test passed

real    0m2.637s
user    0m2.226s
sys     0m0.397s
```

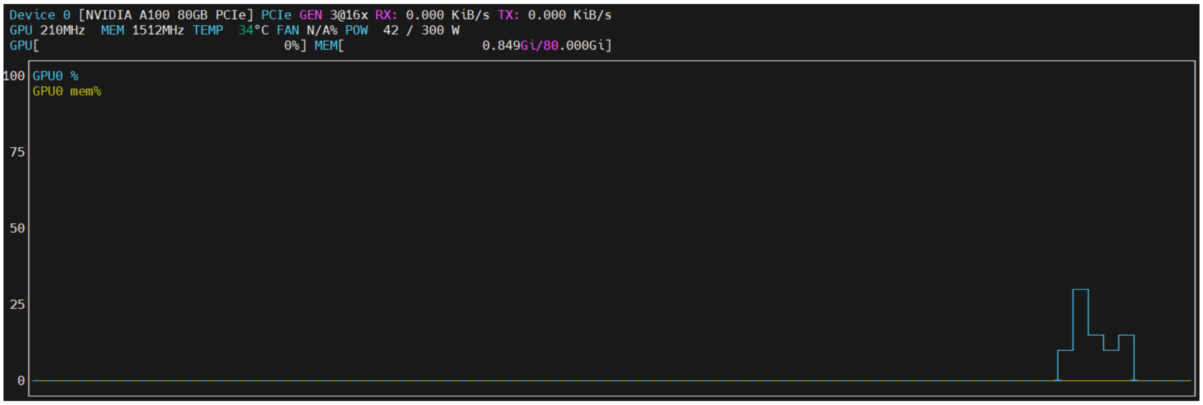


```
Device 0 [NVIDIA A100 80GB PCIe] PCIe GEN 3@16x RX: 0.000 KiB/s TX: 0.000 KiB/s
GPU 210MHz  MEM 1512MHz TEMP  33°C FAN N/A% POW  42 / 300 W
GPU[                              0%] MEM[                  0.849Gi/80.000Gi]
```

Output of 4096x4096 matrix^



```
Matrix size: 8192x8192 (512x512 tiles), tile size: 16x16, block size: 16x16

transpose simple copy       , Throughput = 1268.6584 GB/s, Time = 0.39412 ms, Size = 67108864 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose shared memory copy, Throughput = 1234.2490 GB/s, Time = 0.40510 ms, Size = 67108864 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose naive             , Throughput = 933.0096 GB/s, Time = 0.53590 ms, Size = 67108864 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose coalesced         , Throughput = 1123.2861 GB/s, Time = 0.44512 ms, Size = 67108864 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose optimized         , Throughput = 1184.8323 GB/s, Time = 0.42200 ms, Size = 67108864 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose coarse-grained    , Throughput = 1184.8898 GB/s, Time = 0.42198 ms, Size = 67108864 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose fine-grained      , Throughput = 1231.7893 GB/s, Time = 0.40591 ms, Size = 67108864 fp32 elements, NumDevsUsed = 1, Workgroup = 256
transpose diagonal          , Throughput = 979.4224 GB/s, Time = 0.51050 ms, Size = 67108864 fp32 elements, NumDevsUsed = 1, Workgroup = 256
Test passed

real    0m9.983s
user    0m9.241s
sys     0m0.744s
```



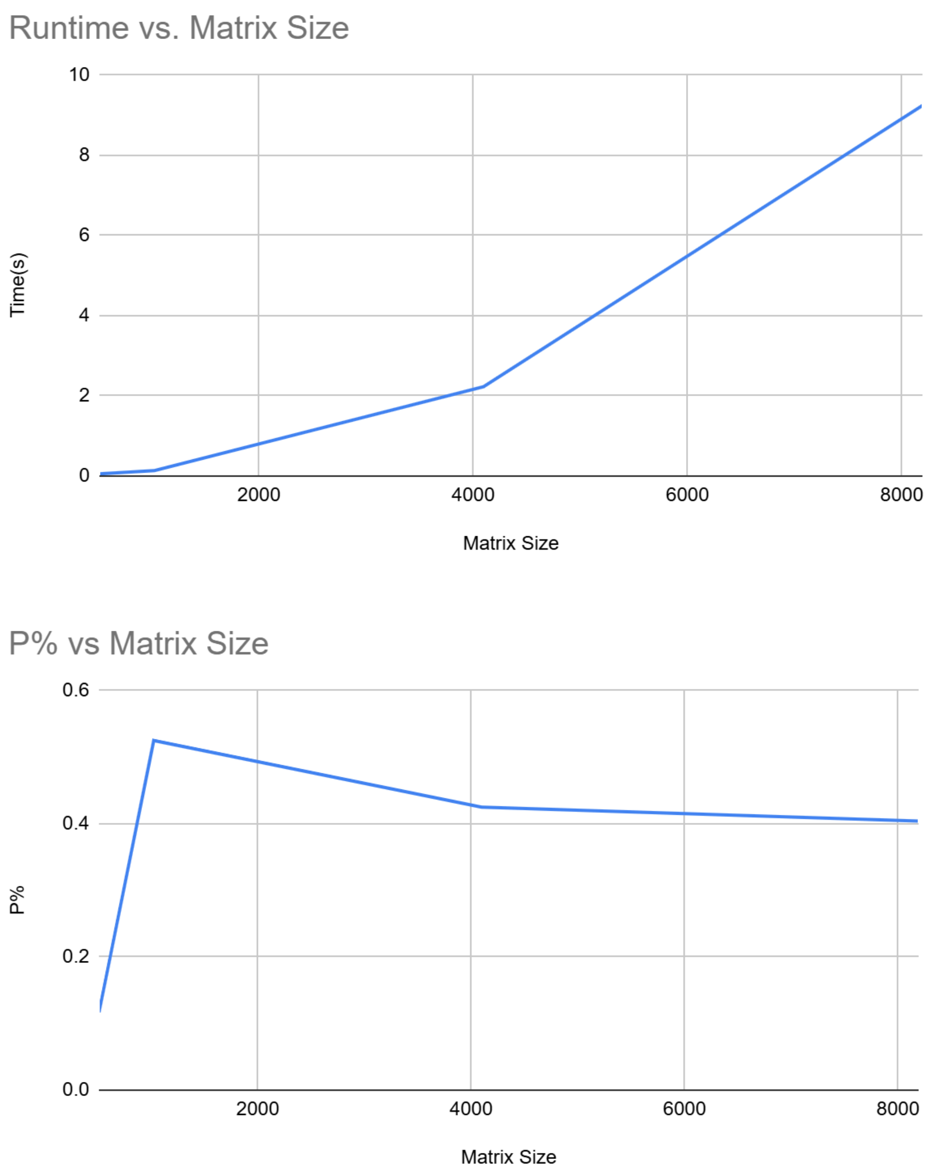
```
Device 0 [NVIDIA A100 80GB PCIe] PCIe GEN 3@16x RX: 0.000 KiB/s TX: 0.000 KiB/s
GPU 210MHz  MEM 1512MHz TEMP  34°C FAN N/A% POW  42 / 300 W
GPU[                              0%] MEM[                  0.849Gi/80.000Gi]
```

Output of 8192x8192 matrix^

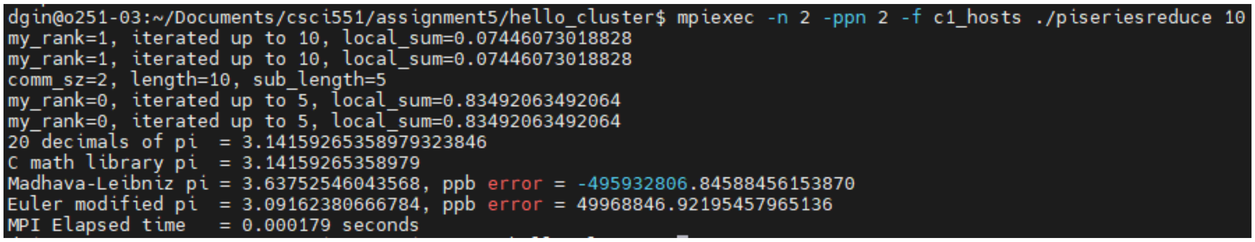
Sequential Times:

Matrix Dimensions	Sequential	CUDA
256	0.014s	0.028s
512	0.060s	0.053s
1024	0.278s	0.132s
4096	3.871s	2.226s
8192	15.498s	9.241s

Parallel portion => P% = C(1-1/SU)/C-1:
256: 6912(1-1/0.5)/6911 = -1.0001 => Sequential was faster so this breaks the equation
512: 6912(1-1/1.132)/6911 = 0.117 => **11.7% P**
1024: 6912(1-1/2.106)/6911 = 0.525 => **52.5% P**
4096: 6912(1-1/1.739)/6911 = 0.425 => **42.5% P**
8192: 6912(1-1/1.677)/6911 = 0.404 => **40.4% P**



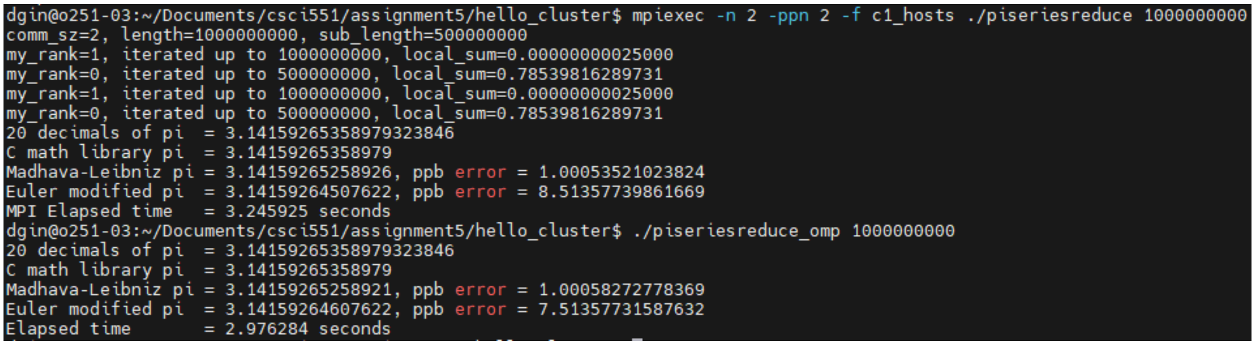
Runtimes and P% visualized^



```
dgin@o251-03:~/Documents/csci551/assignment5/hello_cluster$ mpiexec -n 2 -ppn 2 -f c1_hosts ./piseriesreduce 10
my_rank=1, iterated up to 10, local_sum=0.07446073018828
my_rank=1, iterated up to 10, local_sum=0.07446073018828
comm_sz=2, length=10, sub_length=5
my_rank=0, iterated up to 5, local_sum=0.83492063492064
my_rank=0, iterated up to 5, local_sum=0.83492063492064
20 decimals of pi  = 3.14159265358979323846
C math library pi  = 3.14159265358979
Madhava-Leibniz pi = 3.63752546043568, ppb error = -495932806.84588456153870
Euler modified pi  = 3.09162380666784, ppb error = 49968846.92195457965136
MPI Elapsed time   = 0.000179 seconds
```

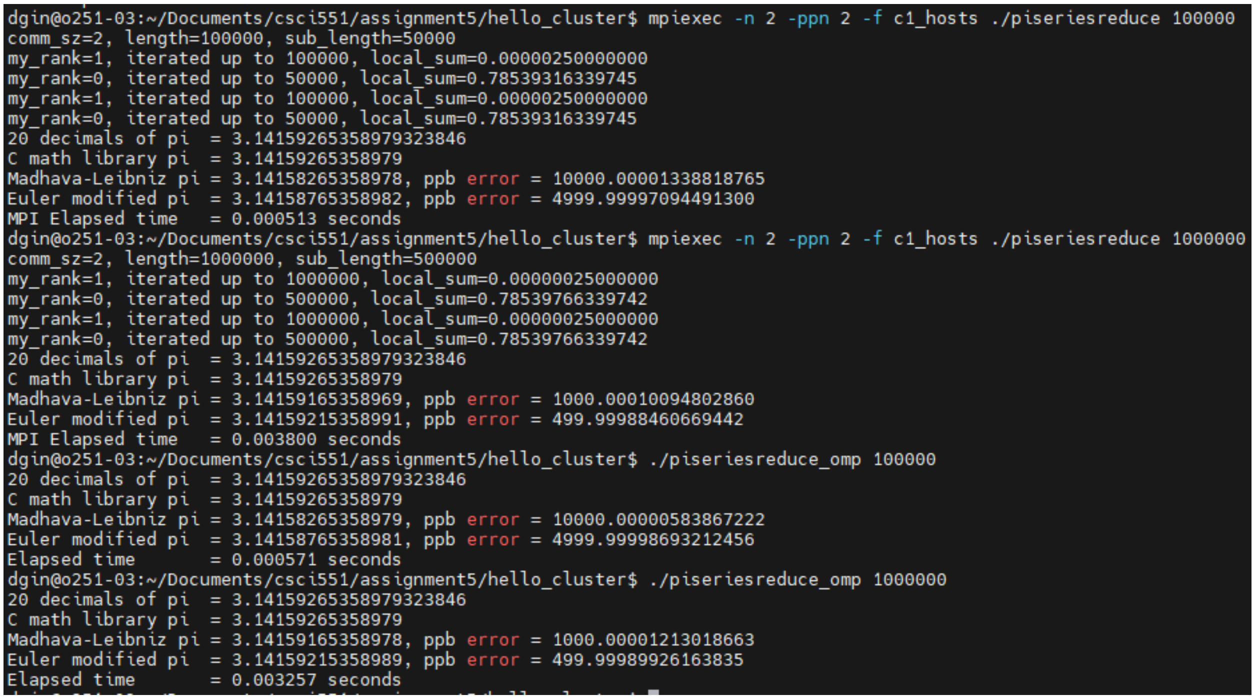
MPI execution on 10 iterations^

With only 10 iterations, the pi estimation is expectedly quite off. Quite satisfyingly, every multiple of 10 that the iterations are increased also increases the estimated pi value accuracy by a multiple of 10. With this trend, I found that beyond 1 billion iterations, the estimation was much better within a hundred-millionth of the expected value.



```
dgin@o251-03:~/Documents/csci551/assignment5/hello_cluster$ mpiexec -n 2 -ppn 2 -f c1_hosts ./piseriesreduce 1000000000
comm_sz=2, length=1000000000, sub_length=500000000
my_rank=1, iterated up to 1000000000, local_sum=0.00000000025000
my_rank=0, iterated up to 500000000, local_sum=0.78539816289731
my_rank=1, iterated up to 1000000000, local_sum=0.00000000025000
my_rank=0, iterated up to 500000000, local_sum=0.78539816289731
20 decimals of pi  = 3.14159265358979323846
C math library pi  = 3.14159265358979
Madhava-Leibniz pi = 3.14159265258926, ppb error = 1.00053521023824
Euler modified pi  = 3.14159264507622, ppb error = 8.51357739861669
MPI Elapsed time   = 3.245925 seconds
dgin@o251-03:~/Documents/csci551/assignment5/hello_cluster$ ./piseriesreduce_omp 1000000000
20 decimals of pi  = 3.14159265358979323846
C math library pi  = 3.14159265358979
Madhava-Leibniz pi = 3.14159265258921, ppb error = 1.00058272778369
Euler modified pi  = 3.14159264607622, ppb error = 7.51357731587632
Elapsed time       = 2.976284 seconds
```

OpenMP and MPI execution of 1 billion iterations^

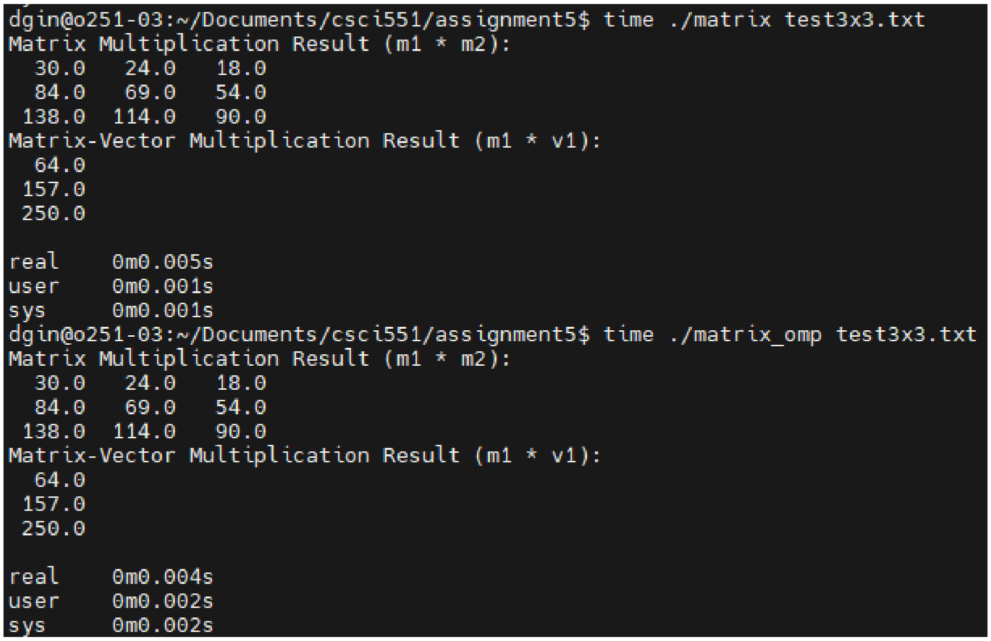


```
dgin@o251-03:~/Documents/csci551/assignment5/hello_cluster$ mpiexec -n 2 -ppn 2 -f c1_hosts ./piseriesreduce 100000
comm_sz=2, length=100000, sub_length=50000
my_rank=1, iterated up to 100000, local_sum=0.00000250000000
my_rank=0, iterated up to 50000, local_sum=0.78539316339745
my_rank=1, iterated up to 100000, local_sum=0.00000250000000
my_rank=0, iterated up to 50000, local_sum=0.78539316339745
20 decimals of pi  = 3.14159265358979323846
C math library pi  = 3.14159265358979
Madhava-Leibniz pi = 3.14158265358978, ppb error = 10000.00001338818765
Euler modified pi  = 3.14158765358982, ppb error = 4999.99997094491300
MPI Elapsed time   = 0.000513 seconds
dgin@o251-03:~/Documents/csci551/assignment5/hello_cluster$ mpiexec -n 2 -ppn 2 -f c1_hosts ./piseriesreduce 1000000
comm_sz=2, length=1000000, sub_length=500000
my_rank=1, iterated up to 1000000, local_sum=0.00000025000000
my_rank=0, iterated up to 500000, local_sum=0.78539766339742
my_rank=1, iterated up to 1000000, local_sum=0.00000025000000
my_rank=0, iterated up to 500000, local_sum=0.78539766339742
20 decimals of pi  = 3.14159265358979323846
C math library pi  = 3.14159265358979
Madhava-Leibniz pi = 3.14159165358969, ppb error = 1000.00010094802860
Euler modified pi  = 3.14159215358991, ppb error = 499.99988460669442
MPI Elapsed time   = 0.003800 seconds
dgin@o251-03:~/Documents/csci551/assignment5/hello_cluster$ ./piseriesreduce_omp 100000
20 decimals of pi  = 3.14159265358979323846
C math library pi  = 3.14159265358979
Madhava-Leibniz pi = 3.14158265358979, ppb error = 10000.00000583867222
Euler modified pi  = 3.14158765358981, ppb error = 4999.99998693212456
Elapsed time       = 0.000571 seconds
dgin@o251-03:~/Documents/csci551/assignment5/hello_cluster$ ./piseriesreduce_omp 1000000
20 decimals of pi  = 3.14159265358979323846
C math library pi  = 3.14159265358979
Madhava-Leibniz pi = 3.14159165358978, ppb error = 1000.00001213018663
Euler modified pi  = 3.14159215358989, ppb error = 499.99989926163835
Elapsed time       = 0.003257 seconds
```

OpenMP and MPI execution on range between 100,000 - 1,000,000

There was very little difference between OpenMP and MPI outputs. The most notable difference was that the Euler estimation was more accurate than the Madhava-Leibniz at 1 million iterations for both implementations.

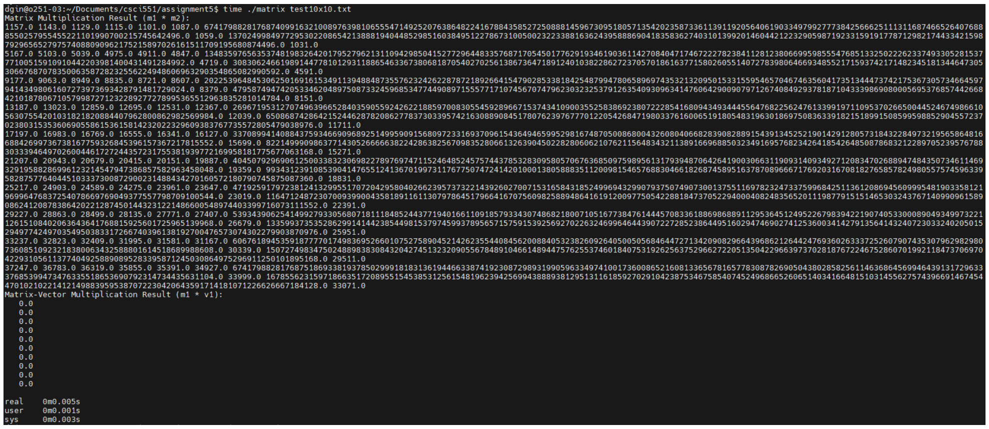
3)



```
dgin@o251-03:~/Documents/csci551/assignment5$ time ./matrix test3x3.txt
Matrix Multiplication Result (m1 * m2):
  30.0   24.0   18.0
  84.0   69.0   54.0
 138.0  114.0   90.0
Matrix-Vector Multiplication Result (m1 * v1):
  64.0
 157.0
 250.0

real    0m0.005s
user    0m0.001s
sys     0m0.001s
dgin@o251-03:~/Documents/csci551/assignment5$ time ./matrix_omp test3x3.txt
Matrix Multiplication Result (m1 * m2):
  30.0   24.0   18.0
  84.0   69.0   54.0
 138.0  114.0   90.0
Matrix-Vector Multiplication Result (m1 * v1):
  64.0
 157.0
 250.0

real    0m0.004s
user    0m0.002s
sys     0m0.002s
```

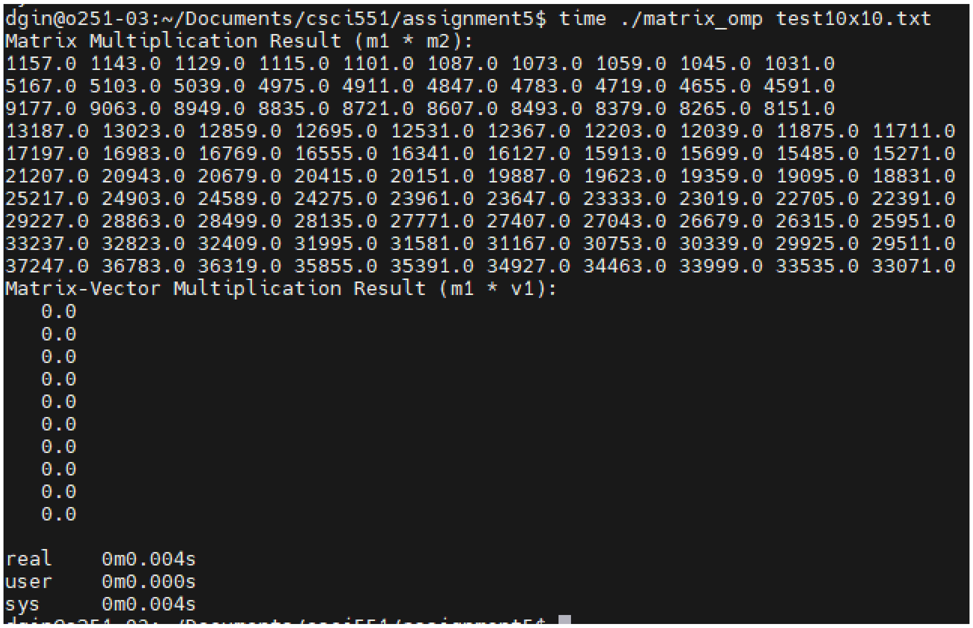
Run on given 3x3 matrix^

Unfortunately, rather than reaching speedup analysis, I instead ran into precision issues, so I'm pretty certain that any comparison would be incorrect. In the very least, the OpenMP implementation didn't run into such extreme rounding errors.



```
dgin@o251-03:~/Documents/csci551/assignment5$ time ./matrix test10x10.txt
Matrix Multiplication Result (m1 * m2):
1157.0 1143.0 1129.0 1115.0 1101.0 1087.0 67417988281768740991632100897639810655547149252076386482241678843585272508881459673095180571354202358733611391192056406190334979927773842566625111311687466526407688
8550257955455221101990700215745642496.0 1059.0 137024998497729530220865421388819404485298516038495122786731005002322338816362439588869041835836274031013992014604421223290598719233159191778712982174433421598
792965652797574088090962175215897026161511709195680874496.0 1031.0
5167.0 5103.0 5039.0 4975.0 4911.0 4847.0 13483597656353748198326420179527962131109429850415277296448335768717054501776291934619036114270840471746722278238411281238066995985554768513325022262337493305281537
77100515910910442203981400431491284992.0 4719.0 30830624661989144778101293118865463367380681870540270256138673647189124010382286272370570186163771580260551407278398064669348552171593742171482345181344647305
3066768707835006358728232556224948606963290354865082990592.0 4591.0
9177.0 9063.0 8949.0 8835.0 8721.0 8607.0 20225396484530625016916153491139488487355762324262287872189266415479028533818425487994780658969743532132095015331559546570467463560417351344473742175367305734664597
94143498061607273973693428791481729024.0 8379.0 47958749474205334620489750873324596853477449089715557717107456707479623032325379126354093096341476064290090797126740849293781871043339869080005695376857442668
4210187806710579987271232289277278995365512963835281014784.0 8151.0
13187.0 13023.0 12859.0 12695.0 12531.0 12367.0 26967195312707496396652840359055924262218859700830554592896671537434109003552583869238072228541680943493444556476822562476133991971109537026650044524674986610
56307554201031821820884407962800862982569984.0 12039.0 6508687428642152446287820862778373033957421630889084517807623976777012205426847198033761600651918054831963018697508363391821518991508599598852904557237
02380315353606905586153615814232022329609383767735572805479038976.0 11711.0
17197.0 16983.0 16769.0 16555.0 16341.0 16127.0 33708994140884375934669096892514995909156809723316937096154364946599529816748705008680043260804066828390828891543913452521901429128057318432284973219565864816
68842699736738167759326845396157367217815552.0 15699.0 8221499909863771430526666638224286382567098352806613263904502282806062107621156483432113891669688503234916957682342641854264850878683212289705239576788
30333946497026004461727244357231755381939772169958181775677063168.0 15271.0
21207.0 20943.0 20679.0 20415.0 20151.0 19887.0 40450792969061250033832306982278976974711524648524575744378532830958057067636850975989561317939487064264190030663119093140934927120834702688947484350734611469
32919588286996123214547947386857582963458048.0 19359.0 9934312391085390414765512413670199731176775074724142010001380588835112009815465768830466182687458951637870896667176920316708182765857824980557574596339
58287577640445103337300872900231488434270160572180790745875087360.0 18831.0
25217.0 24903.0 24589.0 24275.0 23961.0 23647.0 47192591797238124132995517072042958040266239573732214392602700715316584318524996943299079375074907300137551169782324733759968425113612086945609995481903358121
96996476837254078669769049377557798709100544.0 23019.0 1164712487230700939900435818911611307978645179664167075609825889486416191200977505422881847370522940004082483565201119877915151465303243767140990961589
086241208783864202212874501443231221486600548974403399716073111552.0 22391.0
29227.0 28863.0 28499.0 28135.0 27771.0 27407.0 539343906254149927933056807181111848524437719401661109185793343074868218007105167738476144457083361886986889112953645124952267983942219074053300089049349973221
12615108402063643641768815925601725965139968.0 26679.0 1335993735352862991414423854498153797459937895657157591539256927022632469964644390722728523864495160294746902741253600341427913564143240723033240205015
294977424970354950383317266740396138192700476573074302279903870976.0 25951.0
33237.0 32823.0 32409.0 31995.0 31581.0 31167.0 60676189453591877770174983695266010752758904521426235544084562008840532382609264050050568464472713420908296643968621264424769360263337252607907435307962982980
73608510923218380063432588801614518689988608.0 30339.0 1507274983475024889838308432042745132320905567848910466148944757625537460184075319262563752966272205135042296639737028187672246752860701992118473706970
422931056113774049258890895283395871245030864975296911250101895168.0 29511.0
37247.0 36783.0 36319.0 35855.0 35391.0 34927.0 67417988281768751869338193785029991818313619446633874192308729893199059633497410017360086521608133656781657783087826905043802858256114636864569946439131729633
37685399473476335518653690792314734435631104.0 33999.0 1678556231597186635172089551545385312561548196239425699438889381295131161859270291042387534675854074524968665260651403416648151031455627574396691467454
470102102214121498839595387072230420643591714181071226626667184128.0 33071.0
Matrix-Vector Multiplication Result (m1 * v1):
   0.0
   0.0
   0.0
   0.0
   0.0
   0.0
   0.0
   0.0
   0.0
   0.0

real    0m0.005s
user    0m0.001s
sys     0m0.003s
```

Overflowing output of sequential implementation on 10x10^

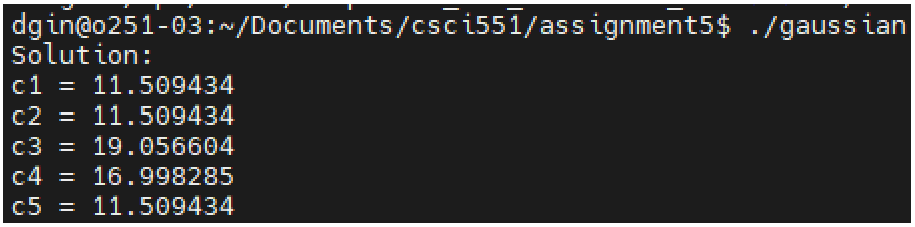


```
dgin@o251-03:~/Documents/csci551/assignment5$ time ./matrix_omp test10x10.txt
Matrix Multiplication Result (m1 * m2):
1157.0 1143.0 1129.0 1115.0 1101.0 1087.0 1073.0 1059.0 1045.0 1031.0
5167.0 5103.0 5039.0 4975.0 4911.0 4847.0 4783.0 4719.0 4655.0 4591.0
9177.0 9063.0 8949.0 8835.0 8721.0 8607.0 8493.0 8379.0 8265.0 8151.0
13187.0 13023.0 12859.0 12695.0 12531.0 12367.0 12203.0 12039.0 11875.0 11711.0
17197.0 16983.0 16769.0 16555.0 16341.0 16127.0 15913.0 15699.0 15485.0 15271.0
21207.0 20943.0 20679.0 20415.0 20151.0 19887.0 19623.0 19359.0 19095.0 18831.0
25217.0 24903.0 24589.0 24275.0 23961.0 23647.0 23333.0 23019.0 22705.0 22391.0
29227.0 28863.0 28499.0 28135.0 27771.0 27407.0 27043.0 26679.0 26315.0 25951.0
33237.0 32823.0 32409.0 31995.0 31581.0 31167.0 30753.0 30339.0 29925.0 29511.0
37247.0 36783.0 36319.0 35855.0 35391.0 34927.0 34463.0 33999.0 33535.0 33071.0
Matrix-Vector Multiplication Result (m1 * v1):
   0.0
   0.0
   0.0
   0.0
   0.0
   0.0
   0.0
   0.0
   0.0
   0.0

real    0m0.004s
user    0m0.000s
sys     0m0.004s
```

OpenMP implementation on 10x10^

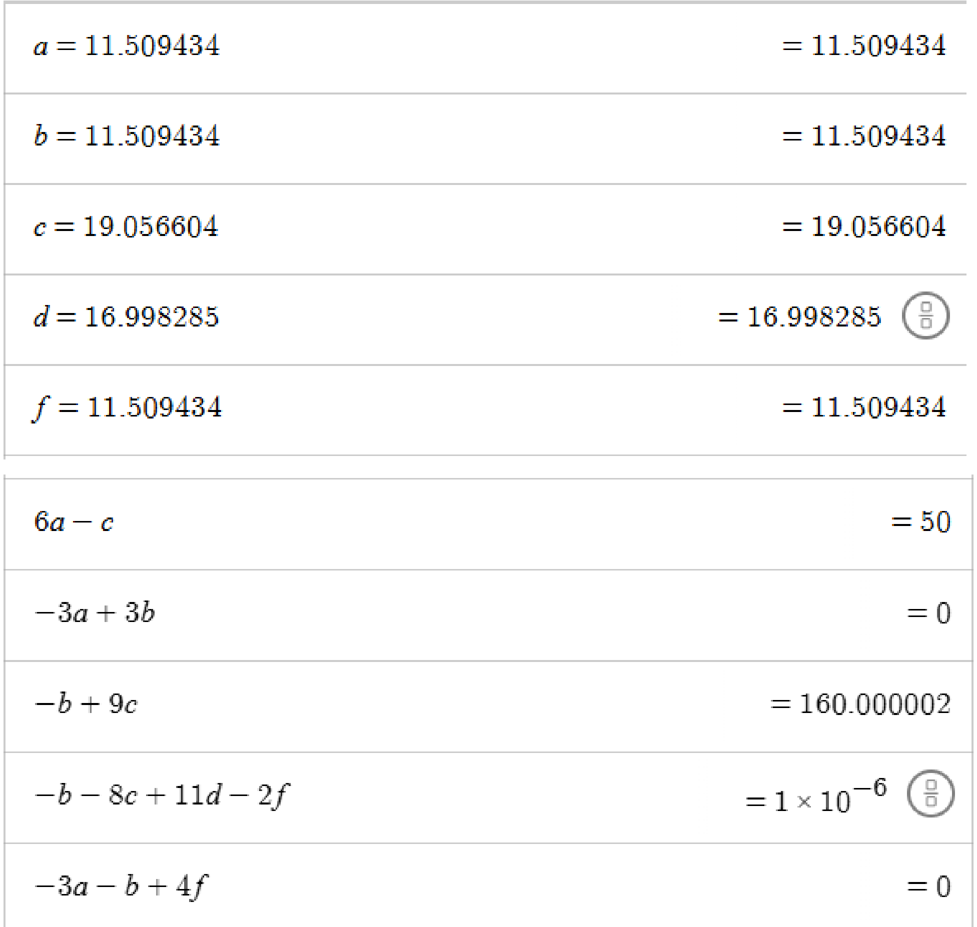
The answers seem quite off. Answers for the 3x3 were only off by ~5-10, but as the expected answers increased in size, so too did the margin of error.
I unfortunately can't work any further on this problem due to time.

4)



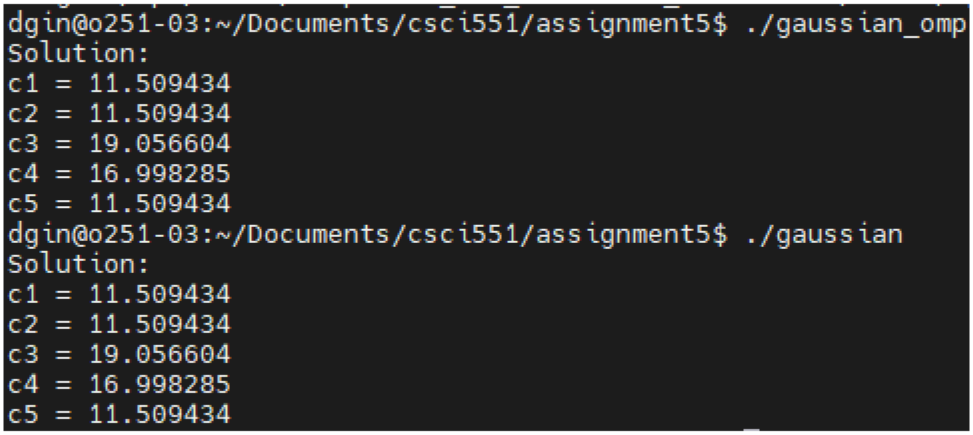
```
dgin@o251-03:~/Documents/csci551/assignment5$ ./gaussian
Solution:
c1 = 11.509434
c2 = 11.509434
c3 = 19.056604
c4 = 16.998285
c5 = 11.509434
```

Gaussian elimination run on hardcoded problem values^



$a = 11.509434$	$= 11.509434$
$b = 11.509434$	$= 11.509434$
$c = 19.056604$	$= 19.056604$
$d = 16.998285$	$= 16.998285$
$f = 11.509434$	$= 11.509434$
$6a - c$	$= 50$
$-3a + 3b$	$= 0$
$-b + 9c$	$= 160.000002$
$-b - 8c + 11d - 2f$	$= 1 \times 10^{-6}$
$-3a - b + 4f$	$= 0$

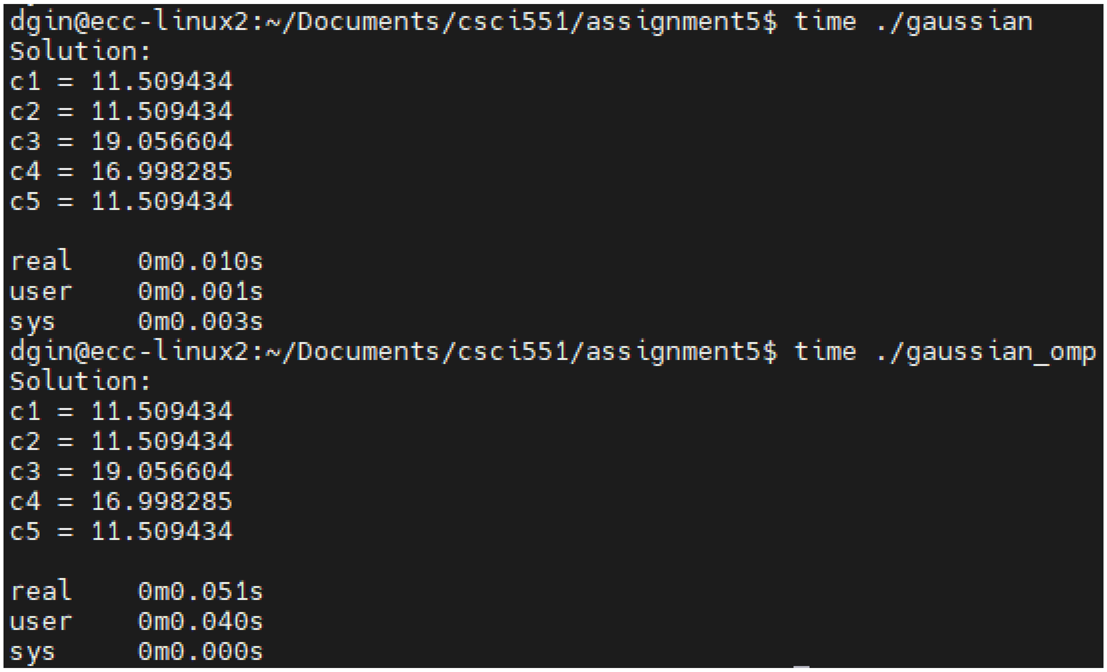
Values confirmed on Desmos scientific calculator^



```
dgin@o251-03:~/Documents/csci551/assignment5$ ./gaussian_omp
Solution:
c1 = 11.509434
c2 = 11.509434
c3 = 19.056604
c4 = 16.998285
c5 = 11.509434
dgin@o251-03:~/Documents/csci551/assignment5$ ./gaussian
Solution:
c1 = 11.509434
c2 = 11.509434
c3 = 19.056604
c4 = 16.998285
c5 = 11.509434
```

Confirmation of matching outputs^

There was no speedup due to the minimal parallelism in my code. As is standard with too small of application, the parallel version ended up being slower due to resynchronizing operations.



```
dgin@ecc-linux2:~/Documents/csci551/assignment5$ time ./gaussian
Solution:
c1 = 11.509434
c2 = 11.509434
c3 = 19.056604
c4 = 16.998285
c5 = 11.509434

real    0m0.010s
user    0m0.001s
sys     0m0.003s
dgin@ecc-linux2:~/Documents/csci551/assignment5$ time ./gaussian_omp
Solution:
c1 = 11.509434
c2 = 11.509434
c3 = 19.056604
c4 = 16.998285
c5 = 11.509434

real    0m0.051s
user    0m0.040s
sys     0m0.000s
```

Sequential vs. parallel runtimes^