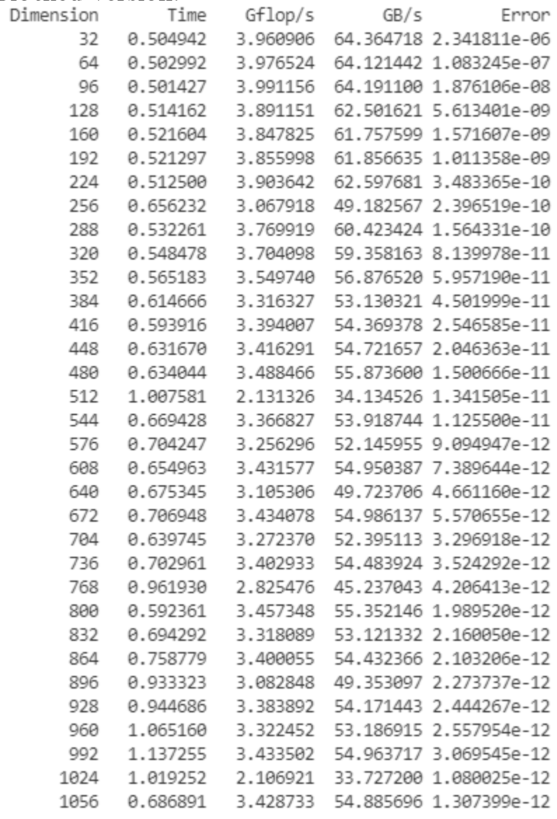
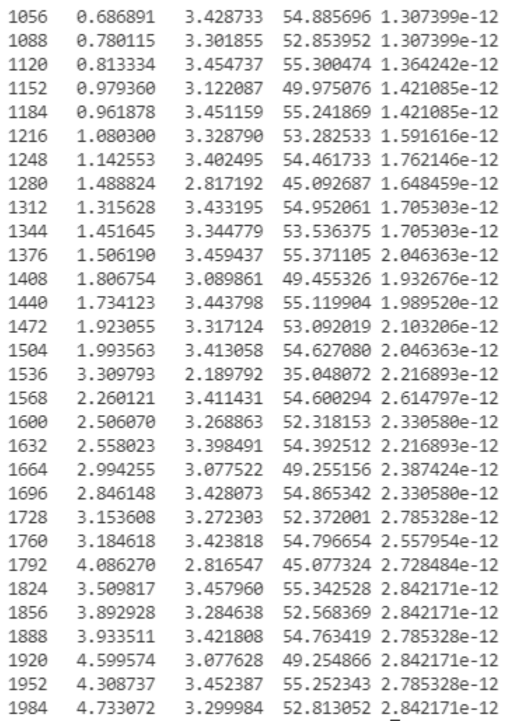
Shengyi Liang
HPC
Prof. Peherstorfer
Homework 2
Here is my github address for homework: https://github.com/TonyLiang0518/Shengyi_Liang_HPC.git

1.
test01:
First error is indexing out of range so changing i <= n to i < n can fix the problem
Second error is mismatch between malloc and delete[], changing delete[] to free(x) works

test02:
The error is indexing uninitialized values at indices 2, 5-9 of x, setting loop at line 81 to initialize all values of x fixes the problem

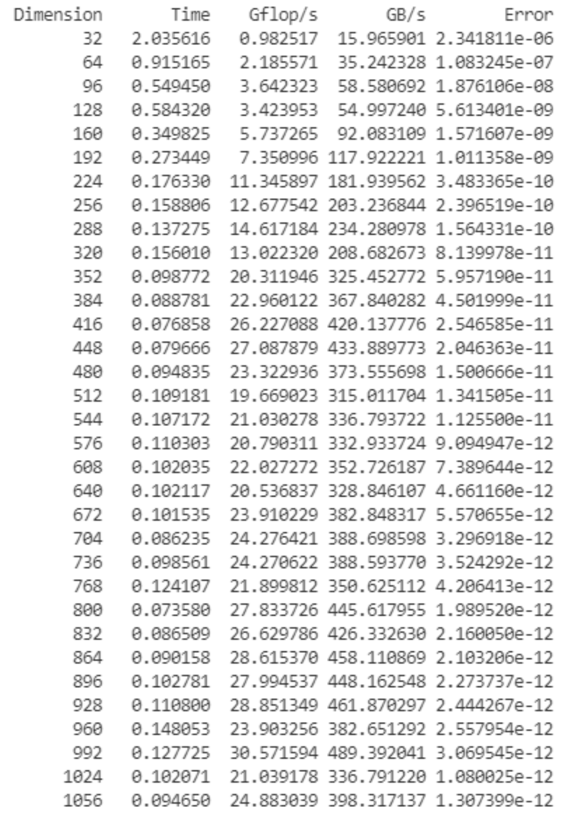
2.
I use Intel i9-9900K 3.6GHz 16 CPUs with 32GBs memory.
Blocked version:


```
 Dimension       Time    Gflop/s       GB/s      Error
        32   0.504942   3.960906  64.364718 2.341811e-06
        64   0.502992   3.976524  64.121442 1.083245e-07
        96   0.501427   3.991156  64.191100 1.876106e-08
       128   0.514162   3.891151  62.501621 5.613401e-09
       160   0.521604   3.847825  61.757599 1.571607e-09
       192   0.521297   3.855998  61.856635 1.011358e-09
       224   0.512500   3.903642  62.597681 3.483365e-10
       256   0.656232   3.067918  49.182567 2.396519e-10
       288   0.532261   3.769919  60.423424 1.564331e-10
       320   0.548478   3.704098  59.358163 8.139978e-11
       352   0.565183   3.549740  56.876520 5.957190e-11
       384   0.614666   3.316327  53.130321 4.501999e-11
       416   0.593916   3.394007  54.369378 2.546585e-11
       448   0.631670   3.416291  54.721657 2.046363e-11
       480   0.634044   3.488466  55.873600 1.500666e-11
       512   1.007581   2.131326  34.134526 1.341505e-11
       544   0.669428   3.366827  53.918744 1.125500e-11
       576   0.704247   3.256296  52.145955 9.094947e-12
       608   0.654963   3.431577  54.950387 7.389644e-12
       640   0.675345   3.105306  49.723706 4.661160e-12
       672   0.706948   3.434078  54.986137 5.570655e-12
       704   0.639745   3.272370  52.395113 3.296918e-12
       736   0.702961   3.402933  54.483924 3.524292e-12
       768   0.961930   2.825476  45.237043 4.206413e-12
       800   0.592361   3.457348  55.352146 1.989520e-12
       832   0.694292   3.318089  53.121332 2.160050e-12
       864   0.758779   3.400055  54.432366 2.103206e-12
       896   0.933323   3.082848  49.353097 2.273737e-12
       928   0.944686   3.383892  54.171443 2.444267e-12
       960   1.065160   3.322452  53.186915 2.557954e-12
       992   1.137255   3.433502  54.963717 3.069545e-12
      1024   1.019252   2.106921  33.727200 1.080025e-12
      1056   0.686891   3.428733  54.885696 1.307399e-12
```

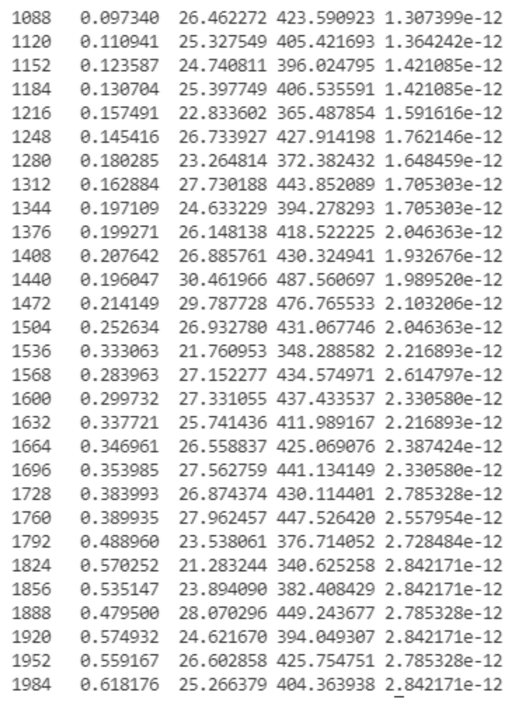


```
1056   0.686891   3.428733   54.885696 1.307399e-12
1088   0.780115   3.301855   52.853952 1.307399e-12
1120   0.813334   3.454737   55.300474 1.364242e-12
1152   0.979360   3.122087   49.975076 1.421085e-12
1184   0.961878   3.451159   55.241869 1.421085e-12
1216   1.080300   3.328790   53.282533 1.591616e-12
1248   1.142553   3.402495   54.461733 1.762146e-12
1280   1.488824   2.817192   45.092687 1.648459e-12
1312   1.315628   3.433195   54.952061 1.705303e-12
1344   1.451645   3.344779   53.536375 1.705303e-12
1376   1.506190   3.459437   55.371105 2.046363e-12
1408   1.806754   3.089861   49.455326 1.932676e-12
1440   1.734123   3.443798   55.119904 1.989520e-12
1472   1.923055   3.317124   53.092019 2.103206e-12
1504   1.993563   3.413058   54.627080 2.046363e-12
1536   3.309793   2.189792   35.048072 2.216893e-12
1568   2.260121   3.411431   54.600294 2.614797e-12
1600   2.506070   3.268863   52.318153 2.330580e-12
1632   2.558023   3.398491   54.392512 2.216893e-12
1664   2.994255   3.077522   49.255156 2.387424e-12
1696   2.846148   3.428073   54.865342 2.330580e-12
1728   3.153608   3.272303   52.372001 2.785328e-12
1760   3.184618   3.423818   54.796654 2.557954e-12
1792   4.086270   2.816547   45.077324 2.728484e-12
1824   3.509817   3.457960   55.342528 2.842171e-12
1856   3.892928   3.284638   52.568369 2.842171e-12
1888   3.933511   3.421808   54.763419 2.785328e-12
1920   4.599574   3.077628   49.254866 2.842171e-12
1952   4.308737   3.452387   55.252343 2.785328e-12
1984   4.733072   3.299984   52.813052 2.842171e-12
```

OpenMP optimized version:



```
 Dimension       Time    Gflop/s       GB/s        Error
        32   2.035616   0.982517  15.965901 2.341811e-06
        64   0.915165   2.185571  35.242328 1.083245e-07
        96   0.549450   3.642323  58.580692 1.876106e-08
       128   0.584320   3.423953  54.997240 5.613401e-09
       160   0.349825   5.737265  92.083109 1.571607e-09
       192   0.273449   7.350996 117.922221 1.011358e-09
       224   0.176330  11.345897 181.939562 3.483365e-10
       256   0.158806  12.677542 203.236844 2.396519e-10
       288   0.137275  14.617184 234.280978 1.564331e-10
       320   0.156010  13.022320 208.682673 8.139978e-11
       352   0.098772  20.311946 325.452772 5.957190e-11
       384   0.088781  22.960122 367.840282 4.501999e-11
       416   0.076858  26.227088 420.137776 2.546585e-11
       448   0.079666  27.087879 433.889773 2.046363e-11
       480   0.094835  23.322936 373.555698 1.500666e-11
       512   0.109181  19.669023 315.011704 1.341505e-11
       544   0.107172  21.030278 336.793722 1.125500e-11
       576   0.110303  20.790311 332.933724 9.094947e-12
       608   0.102035  22.027272 352.726187 7.389644e-12
       640   0.102117  20.536837 328.846107 4.661160e-12
       672   0.101535  23.910229 382.848317 5.570655e-12
       704   0.086235  24.276421 388.698598 3.296918e-12
       736   0.098561  24.270622 388.593770 3.524292e-12
       768   0.124107  21.899812 350.625112 4.206413e-12
       800   0.073580  27.833726 445.617955 1.989520e-12
       832   0.086509  26.629786 426.332630 2.160050e-12
       864   0.090158  28.615370 458.110869 2.103206e-12
       896   0.102781  27.994537 448.162548 2.273737e-12
       928   0.110800  28.851349 461.870297 2.444267e-12
       960   0.148053  23.903256 382.651292 2.557954e-12
       992   0.127725  30.571594 489.392041 3.069545e-12
      1024   0.102071  21.039178 336.791220 1.080025e-12
      1056   0.094650  24.883039 398.317137 1.307399e-12
```



```
1088   0.097340  26.462272 423.590923 1.307399e-12
1120   0.110941  25.327549 405.421693 1.364242e-12
1152   0.123587  24.740811 396.024795 1.421085e-12
1184   0.130704  25.397749 406.535591 1.421085e-12
1216   0.157491  22.833602 365.487854 1.591616e-12
1248   0.145416  26.733927 427.914198 1.762146e-12
1280   0.180285  23.264814 372.382432 1.648459e-12
1312   0.162884  27.730188 443.852089 1.705303e-12
1344   0.197109  24.633229 394.278293 1.705303e-12
1376   0.199271  26.148138 418.522225 2.046363e-12
1408   0.207642  26.885761 430.324941 1.932676e-12
1440   0.196047  30.461966 487.560697 1.989520e-12
1472   0.214149  29.787728 476.765533 2.103206e-12
1504   0.252634  26.932780 431.067746 2.046363e-12
1536   0.333063  21.760953 348.288582 2.216893e-12
1568   0.283963  27.152277 434.574971 2.614797e-12
1600   0.299732  27.331055 437.433537 2.330580e-12
1632   0.337721  25.741436 411.989167 2.216893e-12
1664   0.346961  26.558837 425.069076 2.387424e-12
1696   0.353985  27.562759 441.134149 2.330580e-12
1728   0.383993  26.874374 430.114401 2.785328e-12
1760   0.389935  27.962457 447.526420 2.557954e-12
1792   0.488960  23.538061 376.714052 2.728484e-12
1824   0.570252  21.283244 340.625258 2.842171e-12
1856   0.535147  23.894090 382.408429 2.842171e-12
1888   0.479500  28.070296 449.243677 2.785328e-12
1920   0.574932  24.621670 394.049307 2.842171e-12
1952   0.559167  26.602858 425.754751 2.785328e-12
1984   0.618176  25.266379 404.363938 2.842171e-12
```

3.

omp_bug2:

The error is the shared variables: $tid, i, total$.

By setting $private(tid)$ at line 18 and creating new parallel construct at line 33 with $private(total, i)$, the issue is resolved.

Another minor error is that the output does not always have "Number of threads = 16" at the top, problem fixed by adding barrier at line 27.

omp_bug3:

The error is at line 86, there is a barrier to wait for all threads to execute and proceed but only two threads will eventually be able to reach it. Problem fixed by commenting out the barrier

omp_bug4:

The 2d array a of size $1048 * 1048 * sizeof(double)$ is too large and there is no calculation involving double precision so problem is fixed by initializing array $int\ a[N][N]$

omp_bug5:

There is a deadlock appears when two sections runs simultaneously. After $locka$ is set in section 1, $lockb$ in section 2 may be set as well; in this case, section 1 is unable to perform the operation "adding a[] to b[]" while section 2 is unable to perform the operation "adding b[] to a[]" and a deadlock appears. Moreover, it is also possible that when there is only 1 thread, section 1 would lead to computation of uninitialized values in b.

To fix this, I first set lock on both a and b at beginning of section 1. After initialization ends in either section 1 or 2, unset the lock on a or b so initialization will end for sure for both a and b.

omp_bug6:

1. $dotprod$ should be void, fixed by simply replacing $float$ by $void$

2. sum is initialized both in $main$ and $dotprod$ and not shared properly, to fix this, I initialize it as global at line 15

4.
I use Intel i9-9900K 3.6GHz 16 CPUs with 32GBs memory.

Jacobi runtime:

	N=100	N=200	N=400
2 Threads	0.210s	3.283s	51.034s
4 Threads	0.136s	1.921s	30.785s
8 Threads	0.122s	1.310s	19.708s

Gauss-Seidel runtime:

	N=100	N=200	N=400
2 Threads	0.197s	2.590s	41.064s
4 Threads	0.140s	1.770s	28.408s
8 Threads	0.148s	1.470s	20.884s

As N increases, the runtime increases as expected and each time N doubles, total runtime for same number of threads quadruples since we compute in two dimensions. On the other hand, the iterations needed quadruples as well which means the compute time for each iteration is about the same for different N due to OpenMP.