

# HW2 wf2099

---

- HW2 wf2099
  - 1. Finding Memory bugs using valgrind
    - Val\_test01
    - Val\_test02
  - 2. Optimizing matrix-matrix multiplication.
    - Comparing different loop arrangements for MMult0.
    - one level blocking scheme
    - When use OpenMP
  - 3. Fast Sin
  - 4. Pipelining and optimization
    - compute
    - Compute vec
    - Compute vec pipeline
    - Compare different M
- Appendix
  - Optimizing matrix-matrix multiplication output
    - MMULT0 (  $n * k * m$  )
    - $m * n * k$
    - $k * m * n$
  - one level blocking scheme
    - BS = 16
    - BS = 4
    - BS = 32
  - OpenMP

## 1. Finding Memory bugs using valgrind

1. Run in CIMS Linux Server. 4 AMD EPYC Processor. 7.6G Memory.
2. GCC 8.2, OpenMP4.0

Val\_test01

First Run:

```
valgrind ./val_test01_solved
==9586== Memcheck, a memory error detector
==9586== Invalid write of size 4
==9586== Invalid read of size 4
==9586== Mismatched free() / delete / delete []
==9586== ERROR SUMMARY: 3 errors from 3 contexts (suppressed: 0 from 0)
```

Two Bug:

```
    x = (int *)malloc((n + 1) * sizeof(int)); // It should allocate n+1
    space not n. Otherwise it would out of index.

    // delete[] x;
    // Here is the bug, it should be free, but not delete.
    free(x);
```

After fix Bug:

```
==10502== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Val\_test02

First Run

```
==12136== Memcheck, a memory error detector
==12136== Conditional jump or move depends on uninitialised value(s)
==12136== Use of uninitialised value of size 8
==12136== ERROR SUMMARY: 24 errors from 4 contexts (suppressed: 0 from 0)
```

One Bug, Initialize the array:

```
// Here is the bug, it didn't initialize, but default is 0?
for (int i = 0; i < 10; i++)
{
    x[i] = 0;
}
```

After fix Bug:

```
==12870== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
[wf2099@access1 hw2]$
```

2. Optimizing matrix-matrix multiplication.

Comparing different loop arrangements for MMult0.

Performance (Dimension 1984)	n * k * m	m * n * k	k * m * n
time (s)	4.5	40.7	79.7
flops (Gflop/s)	3.4	0.38	0.13

Performance (Dimension 1984)	$n * k * m$	$m * n * k$	$k * m * n$
bandwidth (GB/s)	13.79	1.53	0.5
ERR	5.441071e+02	5.441071e+02	5.441071e+02

- Details are in the appendix.

I think the reason why the arrange has such a great impact on performance is the locality. Both Space and time locality. Cause when change  $i$ , you can find its nearby in cache.

```
double A_ip = a[i + p * m];
double B_pj = b[p + j * k];
double C_ij = c[i + j * m];
```

one level blocking scheme

Performance (Dimension 1984)	BS = 16	BS = 4	BS = 32
time (s)	1.55	2.77	1.76
flops (Gflop/s)	10	5.52	8.87
bandwidth (GB/s)	40	22.12	35.54
ERR	5.441071e+02	5.441071e+02	5.441071e+02

- Details are in the appendix. code:

```
void MMult1(long m, long n, long k, double *a, double *b, double *c)
{
    for (long j = 0; j < n; j += BLOCK_SIZE)
    {
        for (long p = 0; p < k; p++)
        {
            for (long i = 0; i < m; i += BLOCK_SIZE)
            {
                for (long jj = j; jj < j + BLOCK_SIZE; ++jj)
                {
                    for (long ii = i; ii < i + BLOCK_SIZE; ++ii)
                    {
                        double A_ip = a[ii + p * m];
                        double B_pj = b[p + jj * k];
                        double C_ij = c[ii + jj * m];
                        C_ij = C_ij + A_ip * B_pj;
                        c[ii + jj * m] = C_ij;
                    }
                }
            }
        }
    }
}
```

```
}
}
```

## When use OpenMP

Dimension	Time	Gflop/s	GB/s	Error
1984	13.307310	1.173721	4.701981	2.176428e+03

## 3. Fast Sin

I do this for sin4\_vec(). result

Reference time:	0.3226	
Taylor time:	1.1904	Error: 6.928125e-12
Intrin time:	0.0021	Error: 2.454130e-03
Vector time:	0.0022	Error: 6.928125e-12

code:

```
void sin4_vector(double *sinx, const double *x)
{
    // The Vec class is defined in the file intrin-wrapper.h
    typedef Vec<double, 4> Vec4;
    Vec4 x1, x2, x3, x5, x7, x9, x11;
    x1 = Vec4::LoadAligned(x);
    x2 = x1 * x1;
    x3 = x1 * x2;
    x5 = x3 * x2;
    x7 = x5 * x2;
    x9 = x7 * x2;
    x11 = x9 * x2;

    Vec4 s = x1;
    s += x3 * c3;
    s += x5 * c5;
    s += x7 * c7;
    s += x9 * c9;
    s += x11 * c11;
    s.StoreAligned(sinx);
}
```

## 4. Pipelining and optimization

x86\_64, AMD EPYC Processor (with IBPB), CPU MHz: 2894.562

## compute

optimize method	seconds	cycles/eval	Glop/s
O3	1.646260	4.775115	1.214629
O0	4.551409	13.199648	0.439405
O2	1.563473	4.534315	1.279134
O1	4.923109	14.277421	0.406235

## Compute vec

## report

```
[wf2099@access1 lecture4]$ g++ -fopenmp -std=c++11 -O3 -march=native
compute-vec.cpp && ./a.out -n 1000000000
time = 1.631802
flop-rate = 4.902162 Gflop/s

time = 1.507671
flop-rate = 5.306070 Gflop/s

time = 1.510943
flop-rate = 5.294581 Gflop/s
```

The different is:

1. compute\_fn0 is standard for loop.
2. compute\_fn1 is vectorized using AVX intrinsics.
3. compute\_fn2 is vectorized using Vec Class.

## Compute vec pipeline

## report

## Compare different M

## Observation

1. As M increase from 1 to 8, both AVS intrinsics and Vec class method have significant improve. I think is the multi process/core result.
2. When M increase from 4 to 9, the standard method also improve significantly. And when M = 16, it is better than vectorized method.

M = 1

```
[wf2099@access1 lecture4]$ g++ -fopenmp -std=c++11 -O3 -march=native
compute-vec-pipe.cpp && ./a.out -n 1000000000
```

```
time = 1.514795
flop-rate = 5.280798 Gflop/s

time = 1.509405
flop-rate = 5.299920 Gflop/s

time = 1.506867
flop-rate = 5.308907 Gflop/s
```

M = 4

```
[wf2099@access1 lecture4]$ g++ -fopenmp -std=c++11 -O3 -march=native
compute-vec-pipe.cpp && ./a.out -n 1000000000
time = 5.312123
flop-rate = 6.023828 Gflop/s

time = 1.506742
flop-rate = 21.237194 Gflop/s

time = 1.508429
flop-rate = 21.213565 Gflop/s
```

M = 8

```
[wf2099@access1 lecture4]$ g++ -fopenmp -std=c++11 -O3 -march=native
compute-vec-pipe.cpp && ./a.out -n 1000000000
time = 1.740536
flop-rate = 36.768448 Gflop/s

time = 1.747677
flop-rate = 36.609474 Gflop/s

time = 1.588393
flop-rate = 40.289816 Gflop/s
```

M = 16

```
[wf2099@access1 lecture4]$ g++ -fopenmp -std=c++11 -O3 -march=native
compute-vec-pipe.cpp && ./a.out -n 1000000000
time = 4.097863
flop-rate = 31.234891 Gflop/s

time = 5.596113
flop-rate = 22.872821 Gflop/s

time = 5.658605
flop-rate = 22.620046 Gflop/s
```

# Appendix

## Optimizing matrix-matrix multiplication output

MMULT0 (  $n * k * m$  )

Dimension	Time	Gflop/s	GB/s	Error
16	0.389202	5.138730	24.408965	1.385951e+06
64	0.210830	9.487061	39.727066	9.440258e+04
112	0.219419	9.117795	37.448087	2.693089e+04
160	0.283867	7.070349	28.811673	1.233188e+04
208	0.247918	8.130742	32.992048	7.315887e+03
256	0.301170	6.684813	27.052601	4.826596e+03
304	0.225179	8.983074	36.286891	3.267805e+03
352	0.228382	8.784631	35.438002	2.471286e+03
400	0.220176	9.301666	37.485714	1.891448e+03
448	0.236387	9.128978	36.760440	1.574612e+03
496	0.233813	9.393980	37.803192	1.312905e+03
544	0.238176	9.462967	38.060609	1.104834e+03
592	0.224794	9.229538	37.105238	8.560616e+02
640	0.235487	8.905594	35.789356	7.445709e+02
688	0.277906	9.374715	37.662372	7.953869e+02
736	0.255033	9.379679	37.671647	6.299729e+02
784	0.318209	9.086290	36.484238	6.724575e+02
832	0.326867	7.047882	28.293181	4.889743e+02
880	0.289321	9.421684	37.815213	4.996540e+02
928	0.343411	9.308723	37.355262	5.309204e+02
976	0.413475	8.994143	36.087156	5.570778e+02
1024	0.548490	3.915262	15.706928	2.879263e+02
1072	0.458317	5.375868	21.563650	3.032963e+02
1120	0.328639	8.549974	34.291501	3.193277e+02
1168	0.385964	8.256793	33.112002	3.318287e+02
1216	0.473931	7.587798	30.426070	3.425009e+02
1264	0.579649	6.967972	27.938041	3.541980e+02
1312	0.726666	6.215792	24.920018	3.661419e+02
1360	0.866787	5.804095	23.267593	3.795629e+02
1408	1.386899	4.025251	16.135312	3.887651e+02
1456	1.567307	3.938764	15.787518	4.048556e+02
1504	1.635187	4.161079	16.677516	4.184804e+02
1552	1.904722	3.925307	15.731577	4.322260e+02
1600	2.415287	3.391729	13.592353	4.431469e+02
1648	2.308219	3.878150	15.540838	4.596910e+02
1696	2.548807	3.827988	15.339038	4.692641e+02
1744	2.950545	3.595566	14.407002	4.866494e+02
1792	3.732496	3.083505	12.354668	4.954003e+02
1840	3.859976	3.227742	12.932020	5.048999e+02
1888	4.575723	2.941550	11.784896	5.153721e+02
1936	4.023079	3.607343	14.451732	5.383513e+02
1984	4.534441	3.444540	13.798995	5.441071e+02

$m * n * k$

Dimension	Time	Gflop/s	GB/s	Error
16	0.482501	4.145075	19.689105	1.385951e+06
64	0.604529	3.308621	13.854850	9.440258e+04
112	0.803658	2.489388	10.224271	2.693089e+04
160	0.742537	2.702950	11.014520	1.233188e+04
208	0.828276	2.433678	9.875117	7.315887e+03
256	3.220182	0.625202	2.530116	4.826596e+03
304	0.966262	2.093429	8.456351	3.267805e+03
352	0.991519	2.023413	8.162633	2.471286e+03
400	1.010062	2.027598	8.171222	1.891448e+03
448	1.399642	1.541801	6.208501	1.574612e+03
496	1.119387	1.962173	7.896165	1.312905e+03
544	1.068389	2.109576	8.484840	1.104834e+03
592	1.063374	1.951098	7.843942	8.560616e+02
640	1.673656	1.253036	5.035640	7.445709e+02
688	1.387061	1.878278	7.545873	7.953869e+02
736	1.114985	2.145436	8.616723	6.299729e+02
784	1.329093	2.175425	8.734997	6.724575e+02
832	1.697939	1.356775	5.446670	4.889743e+02
880	1.352970	2.014744	8.086451	4.996540e+02
928	1.590583	2.009775	8.065090	5.309204e+02
976	1.778209	2.091349	8.391110	5.570778e+02
1024	8.628961	0.248869	0.998394	2.879263e+02
1072	1.295606	1.901697	7.628077	3.032963e+02
1120	1.512499	1.857757	7.450935	3.193277e+02
1168	2.501759	1.273835	5.108426	3.318287e+02
1216	3.348520	1.073934	4.306336	3.425009e+02
1264	1.999995	2.019493	8.097144	3.541980e+02
1312	2.727618	1.655953	6.638957	3.661419e+02
1360	3.207850	1.568313	6.287088	3.795629e+02
1408	6.826230	0.817819	3.278245	3.887651e+02
1456	17.276323	0.357325	1.432243	4.048556e+02
1504	4.312942	1.577611	6.323030	4.184804e+02
1552	5.241086	1.426540	5.717189	4.322260e+02
1600	9.288686	0.881933	3.534347	4.431469e+02
1648	6.823761	1.311831	5.256875	4.596910e+02
1696	10.991228	0.887690	3.557041	4.692641e+02
1744	13.449212	0.788810	3.160669	4.866494e+02
1792	25.899131	0.444384	1.780513	4.954003e+02
1840	25.461491	0.489328	1.960501	5.048999e+02
1888	28.001645	0.480676	1.925759	5.153721e+02
1936	33.351404	0.435143	1.743269	5.383513e+02
1984	40.794496	0.382872	1.533803	5.441071e+02

$k * m * n$



Dimension	Time	Gflop/s	GB/s	Error
16	0.629579	3.176731	15.089470	1.385951e+06
64	0.730679	2.737398	11.462854	9.440258e+04
112	0.530453	3.771523	15.490184	2.693089e+04
160	1.224356	1.639262	6.679993	1.233188e+04
208	1.011874	1.992102	8.083336	7.315887e+03
256	15.988613	0.125919	0.509577	4.826596e+03
304	1.257015	1.609210	6.500361	3.267805e+03
352	1.348202	1.488095	6.003112	2.471286e+03
400	1.394900	1.468205	5.916868	1.891448e+03
448	1.689133	1.277560	5.144462	1.574612e+03
496	1.492252	1.471890	5.923170	1.312905e+03
544	1.634222	1.379157	5.547049	1.104834e+03
592	1.336083	1.552857	6.242906	8.560616e+02
640	2.603997	0.805359	3.236536	7.445709e+02
688	1.860342	1.400434	5.626161	7.953869e+02
736	3.004341	0.796224	3.197879	6.299729e+02
784	2.860353	1.010834	4.058808	6.724575e+02
832	2.639635	0.872742	3.503557	4.889743e+02
880	4.678871	0.582595	2.338325	4.996540e+02
928	11.067304	0.288843	1.159108	5.309204e+02
976	17.310106	0.214837	0.861991	5.570778e+02
1024	20.849918	0.102997	0.413196	2.879263e+02
1072	14.816514	0.166291	0.667025	3.032963e+02
1120	17.202611	0.163339	0.655106	3.193277e+02
1168	19.579938	0.162760	0.652711	3.318287e+02
1216	24.952311	0.144119	0.577897	3.425009e+02
1264	27.541195	0.146652	0.588001	3.541980e+02
1312	33.361995	0.135388	0.542789	3.661419e+02
1360	38.308784	0.131325	0.526460	3.795629e+02
1408	43.088220	0.129563	0.519354	3.887651e+02
1456	48.525336	0.127217	0.509917	4.048556e+02
1504	55.993478	0.121517	0.487036	4.184804e+02
1552	63.284940	0.118142	0.473482	4.322260e+02
1600	66.339399	0.123486	0.494871	4.431469e+02
1648	66.399323	0.134815	0.540241	4.596910e+02
1696	75.085905	0.129942	0.520687	4.692641e+02
1744	86.490421	0.122660	0.491482	4.866494e+02
1792	88.284345	0.130365	0.522332	4.954003e+02
1840	92.097975	0.135280	0.542002	5.048999e+02
1888	100.077450	0.134493	0.538827	5.153721e+02
1936	107.628582	0.134840	0.540195	5.383513e+02
1984	118.973367	0.131282	0.525922	5.441071e+02

## one level blocking scheme

BS = 16

Dimension	Time	Gflop/s	GB/s	Error
16	0.151593	13.193213	62.667762	1.385951e+06

64	0.151174	13.230858	55.404216	9.440258e+04
112	0.153431	13.039161	53.553698	2.693089e+04
160	0.154294	13.007859	53.007025	1.233188e+04
208	0.156780	12.857269	52.170842	7.315887e+03
256	0.201870	9.973102	40.359896	4.826596e+03
304	0.171511	11.794025	47.641653	3.267805e+03
352	0.174994	11.464695	46.249623	2.471286e+03
400	0.179123	11.433492	46.076972	1.891448e+03
448	0.189355	11.396404	45.890878	1.574612e+03
496	0.198799	11.048507	44.461329	1.312905e+03
544	0.198435	11.358130	45.683069	1.104834e+03
592	0.183022	11.336021	45.573868	8.560616e+02
640	0.188480	11.126627	44.715132	7.445709e+02
688	0.231793	11.239695	45.154821	7.953869e+02
736	0.212373	11.263806	45.238873	6.299729e+02
784	0.317768	9.098902	36.534876	6.724575e+02
832	0.204866	11.245009	45.142223	4.889743e+02
880	0.242609	11.235724	45.096111	4.996540e+02
928	0.352915	9.058036	36.349274	5.309204e+02
976	0.330528	11.251258	45.143368	5.570778e+02
1024	0.392235	5.474986	21.964106	2.879263e+02
1072	0.218754	11.263122	45.178570	3.032963e+02
1120	0.247892	11.335022	45.461534	3.193277e+02
1168	0.282861	11.266398	45.181344	3.318287e+02
1216	0.370530	9.705260	38.916815	3.425009e+02
1264	0.362179	11.151862	44.713320	3.541980e+02
1312	0.402927	11.209977	44.942437	3.661419e+02
1360	0.454995	11.057071	44.325848	3.795629e+02
1408	0.607806	9.184872	36.817768	3.887651e+02
1456	0.570394	10.822787	43.380348	4.048556e+02
1504	0.694809	9.792833	39.249466	4.184804e+02
1552	0.708622	10.550927	42.285286	4.322260e+02
1600	0.746216	10.978061	43.994580	4.431469e+02
1648	0.983984	9.097321	36.455527	4.596910e+02
1696	0.941365	10.364525	41.531434	4.692641e+02
1744	1.022224	10.378232	41.584338	4.866494e+02
1792	1.402369	8.206946	32.882743	4.954003e+02
1840	1.201316	10.371133	41.552169	5.048999e+02
1888	1.286984	10.458338	41.899824	5.153721e+02
1936	1.388576	10.451447	41.870569	5.383513e+02
1984	1.550995	10.070350	40.342309	5.441071e+02

BS = 4

Dimension	Time	Gflop/s	GB/s	Error
4	0.376417	5.313256	37.192792	1.930508e+07
52	0.364523	5.486649	23.212746	1.294194e+05
100	0.317827	6.299015	25.951944	3.279681e+04
148	0.319967	6.261363	25.553131	1.488839e+04
196	0.304124	6.585667	26.745871	8.132360e+03
244	0.305241	6.567593	26.593369	5.150747e+03

292	0.311638	6.551064	26.473476	3.653748e+03
340	0.311827	6.554306	26.448550	2.678501e+03
388	0.324041	6.489292	26.157869	2.105263e+03
436	0.335658	6.420007	25.856727	1.675396e+03
484	0.314064	6.498163	26.153765	1.309623e+03
532	0.324576	6.494517	26.124562	1.105332e+03
580	0.359575	6.511427	26.180428	1.011378e+03
628	0.380578	6.507813	26.155604	9.125553e+02
676	0.379880	6.505552	26.137693	7.759236e+02
724	0.353200	6.446822	25.894143	6.326553e+02
772	0.426639	6.470576	25.982883	6.613743e+02
820	0.341261	6.462710	25.945416	4.724340e+02
868	0.407783	6.414901	25.748288	4.941193e+02
916	0.483479	6.358711	25.518146	5.210746e+02
964	0.679066	5.276902	21.173295	5.516814e+02
1012	0.329557	6.289861	25.234028	2.870719e+02
1060	0.377798	6.305040	25.291537	2.958791e+02
1108	0.432964	6.283445	25.201830	3.127530e+02
1156	0.494686	6.245599	25.047227	3.248432e+02
1204	0.559511	6.238796	25.017365	3.360267e+02
1252	0.630086	6.229361	24.977150	3.500426e+02
1300	0.713514	6.158254	24.689859	3.640703e+02
1348	0.802933	6.101271	24.459399	3.794245e+02
1396	0.927079	5.869074	23.526748	3.892219e+02
1444	1.085201	5.549087	22.242464	3.994090e+02
1492	1.115541	5.954575	23.866191	4.176378e+02
1540	1.234421	5.917372	23.715599	4.284149e+02
1588	1.364255	5.870645	23.526944	4.393153e+02
1636	1.517443	5.771219	23.127210	4.523815e+02
1684	1.661568	5.748283	23.034093	4.711033e+02
1732	1.828779	5.682146	22.767953	4.806263e+02
1780	2.184768	5.162792	20.685973	4.884137e+02
1828	2.192141	5.573013	22.328637	5.033187e+02
1876	2.374111	5.561954	22.283395	5.170853e+02
1924	2.568666	5.545460	22.216429	5.273173e+02
1972	2.776654	5.523686	22.128355	5.442381e+02

BS = 32

Dimension	Time	Gflop/s	GB/s	Error
32	0.151868	13.169543	57.616752	3.454625e+05
64	0.274196	7.294640	30.546306	8.645484e+04
96	0.184151	10.867574	44.828743	3.603143e+04
128	0.173076	11.559530	47.321828	2.028065e+04
160	0.181404	11.063931	45.085520	1.285523e+04
192	0.184923	10.870015	44.159437	8.626524e+03
224	0.185809	10.767056	43.645030	6.241077e+03
256	0.219629	9.166680	37.096409	4.740149e+03
288	0.184567	10.871829	43.940308	3.676900e+03
320	0.290033	7.004779	28.281797	3.014212e+03
352	0.197450	10.160840	40.989752	2.393770e+03

384	0.190698	10.689302	43.091248	2.120384e+03
416	0.185259	10.880723	43.836758	1.722277e+03
448	0.201057	10.733126	43.219997	1.587122e+03
480	0.207519	10.658489	42.900418	1.408996e+03
512	0.248860	8.629283	34.719381	1.211501e+03
544	0.219592	10.263806	41.281632	1.120362e+03
576	0.223570	10.257336	41.243040	1.006499e+03
608	0.217028	10.356085	41.628736	8.804057e+02
640	0.208055	10.079785	40.508135	7.327325e+02
672	0.234784	10.340199	41.545443	7.719212e+02
704	0.205823	10.171267	40.858442	6.036365e+02
736	0.231473	10.334387	41.506043	6.293510e+02
768	0.305577	8.894355	35.716394	6.599235e+02
800	0.286649	7.144630	28.685688	4.588220e+02
832	0.229936	10.018948	40.220298	4.751202e+02
864	0.255755	10.087362	40.489549	4.877100e+02
896	0.295615	9.733256	39.063379	5.102025e+02
928	0.315796	10.122719	40.621772	5.313950e+02
960	0.356766	9.919503	39.802006	5.444863e+02
992	0.430040	9.079997	36.429825	5.618113e+02
1024	0.287092	7.480126	30.008163	2.906769e+02
1056	0.235774	9.989094	40.069890	2.992610e+02
1088	0.251540	10.240243	41.073916	3.058866e+02
1120	0.276178	10.174091	40.805372	3.146530e+02
1152	0.315325	9.696818	38.888280	3.241138e+02
1184	0.330412	10.046835	40.289166	3.320246e+02
1216	0.357537	10.057952	40.331065	3.475650e+02
1248	0.386637	10.054719	40.315555	3.481782e+02
1280	0.491562	8.532604	34.210409	3.566596e+02
1312	0.453977	9.949415	39.888660	3.676665e+02
1344	0.508054	9.556915	38.312989	3.780126e+02
1376	0.525832	9.909195	39.723196	3.912954e+02
1408	0.601612	9.279435	37.196827	3.904002e+02
1440	0.625103	9.553581	38.293937	4.022505e+02
1472	0.640948	9.952460	39.890976	4.082739e+02
1504	0.709962	9.583811	38.411712	4.174498e+02
1536	0.940382	7.707244	30.889189	4.298162e+02
1568	0.777819	9.912652	39.726469	4.390920e+02
1600	0.864818	9.472518	37.961117	4.482625e+02
1632	0.938249	9.265577	37.130438	4.580419e+02
1664	0.944725	9.754040	39.086500	4.590257e+02
1696	1.018764	9.577094	38.376139	4.684472e+02
1728	1.177380	8.764850	35.120265	4.795047e+02
1760	1.136957	9.590117	38.425856	4.839984e+02
1792	1.476047	7.797295	31.241394	4.950327e+02
1824	1.350245	8.988594	36.013513	5.030270e+02
1856	1.361203	9.393795	37.635917	5.111048e+02
1888	1.588570	8.472849	33.945250	5.251387e+02
1920	1.857477	7.620968	30.531504	5.275152e+02
1952	1.927056	7.719251	30.924457	5.386715e+02
1984	1.760171	8.873606	35.548094	5.445766e+02

## OpenMP

Dimension	Time	Gflop/s	GB/s	Error
16	21.989616	0.090952	0.432023	3.953354e+06
64	10.820056	0.184857	0.774087	3.407899e+05
112	6.299466	0.317585	1.304368	1.017575e+05
160	4.549159	0.441189	1.797846	4.831103e+04
208	2.931880	0.687530	2.789786	2.857763e+04
256	2.729928	0.737479	2.984487	1.885309e+04
304	2.219474	0.911388	3.681528	1.294787e+04
352	2.255125	0.889642	3.588895	9.752712e+03
400	2.212199	0.925776	3.730875	7.563525e+03
448	2.248084	0.959915	3.865372	6.295509e+03
496	2.170400	1.011993	4.072458	5.234917e+03
544	1.922359	1.172439	4.715620	4.415464e+03
592	1.876354	1.105733	4.445347	3.395948e+03
640	1.454913	1.441428	5.792738	2.976941e+03
688	1.801948	1.445816	5.808481	3.154993e+03
736	2.127876	1.124186	4.515075	2.518677e+03
784	2.547320	1.135052	4.557583	2.674568e+03
832	2.191626	1.051147	4.219749	1.954231e+03
880	2.397333	1.137050	4.563706	1.998178e+03
928	2.966563	1.077582	4.324262	2.118935e+03
976	3.756269	0.990040	3.972333	2.226589e+03
1024	5.866915	0.366033	1.468421	1.138714e+03
1072	2.142776	1.149840	4.612233	1.212227e+03
1120	2.504422	1.121958	4.499853	1.277311e+03
1168	2.872558	1.109404	4.449014	1.326886e+03
1216	3.399094	1.057956	4.242264	1.369478e+03
1264	1.808404	2.233447	8.954993	1.415789e+03
1312	3.937808	1.147036	4.598634	1.464268e+03
1360	4.444765	1.131874	4.537482	1.518108e+03
1408	3.766467	1.482190	5.941391	1.555060e+03
1456	5.734748	1.076465	4.314731	1.619364e+03
1504	5.921955	1.148969	4.605044	1.671654e+03
1552	6.466117	1.156276	4.634045	1.727057e+03
1600	3.535840	2.316847	9.284763	1.772262e+03
1648	6.699633	1.336136	5.354272	1.836995e+03
1696	8.012183	1.217746	4.879600	1.876319e+03
1744	4.605605	2.303471	9.229734	1.946598e+03
1792	9.842674	1.169313	4.685084	1.980012e+03
1840	10.425893	1.195006	4.787819	2.019545e+03
1888	11.363430	1.184477	4.745435	2.061347e+03
1936	12.348966	1.175210	4.708124	2.153405e+03
1984	13.307310	1.173721	4.701981	2.176428e+03