Priyanka Karuppuch Samy

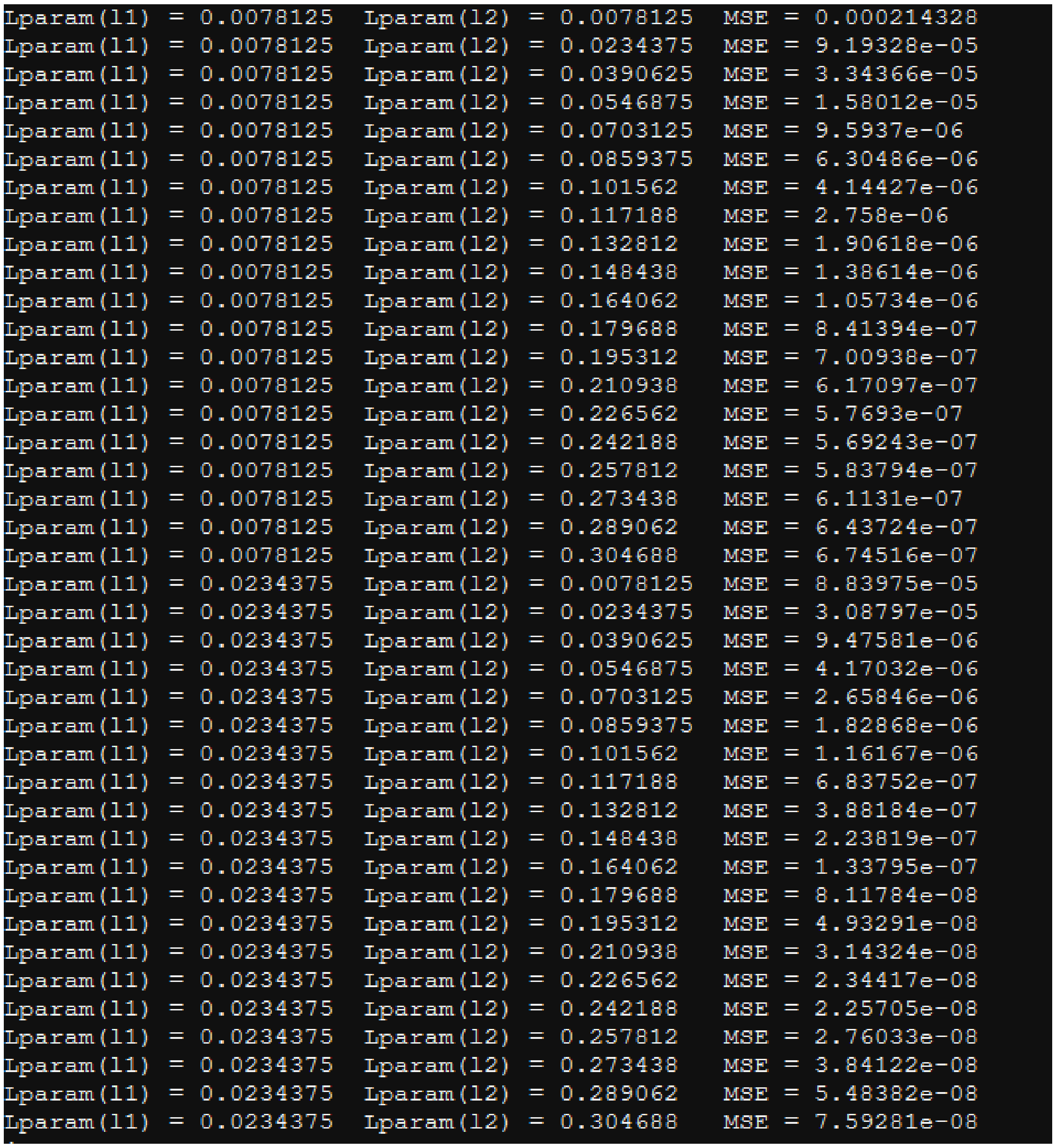
729007151

Major Project – GPR with Hyperparamers to find min MSE

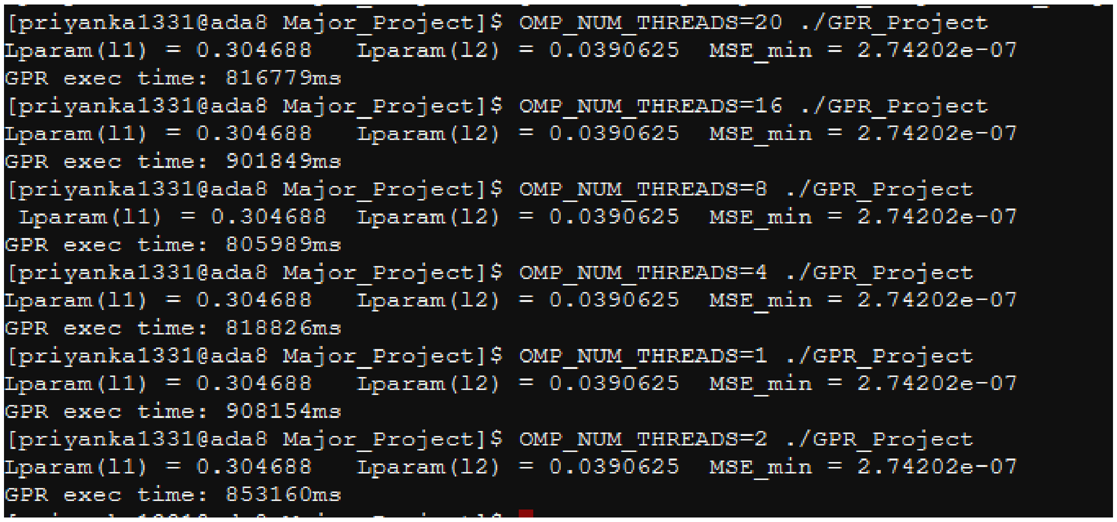
1. Compilation and execution
   - The zip file contains a GPR_Project.cpp,GPR_Project executable and also GPR_Proj_diff_data file. The compilation is done using the command g++ -O3 -o GPR_Project GPR_final.cpp -fopenmp.
   - The other executable(GPR_Proj_diff_data) I have attached is for Q3. This can be compiled similar to above, but by replacing just the executables.
   - The output screenshot is also attached.
   - When compiled, a new a.out will be created and it can be run using the command OMP_NUM_THREADS=#n ./GPR_Project.
   - Eg: OMP_NUM_THREADS=20 ./GPR_Project
   - “m” was set to default given by professor in MATLAB file as 32 initially.
   - “m” can be changed inside the main function in line 579.
   - The changes for generating different dataset(Q3) can be done in lines 391 and 397 by changing 0.25 to 0.5 and 2 to 1.

2. Strategy to Parallelize
   - LU factorization produces an upper triangular and lower triangular matrix. To do this there are 3 loops required. The outer loop can’t be parallelized. Thus, I just parallelized inner loops using #pragma omp for.
   - Similarly, for the function comp_pred_val, the function does operations like calculating inverse and multiplication to do backward and forward substitution. This function is implemented similar to MATLAB implementation provided by professor. Inverse is calculated using Gauss Jordan inverse and matrix multiplication is also parallelized.
   - All the dependent functions are also parallelized based on the functionality, for example a function that returns a 0 matrix can be parallelized completely. So, I allowed nested parallelizing by using #pragma omp parallel for.
   - Creating random matrix could also be parallelized. This would give different prediction values at the output although all of them are correct for the random matrix generated. I preferred to keep it serial so that the output looks consistent.
   - Observed data initialization is also parallelized, creating an identity matrix is parallelized on the outer loop and parallelizing the inner loop were inconsistent because of dependency.

- The challenging part was to parallelize the loops that finds mean square error. The execution should be in order. So, a simple #pragma omp parallel for loop didn’t work. So, I used scheduling static and made the executions to run in order. This significantly improved parallel performance. The output I obtained is as shown below for matrix size 1024*1024. But this strategy worked for matrix sizes 10 and 11. For some reason there was an interference for m = 32 and I ended up removing parallelization for this part and settled with the lower speed up. The improvement with 20 threads was 2-3 minutes which will be significantly higher if m size is increased even more.
- MSE is close to zero implies that the model works good.



```
Lparam(l1) = 0.0078125  Lparam(l2) = 0.0078125  MSE = 0.000214328
Lparam(l1) = 0.0078125  Lparam(l2) = 0.0234375  MSE = 9.19328e-05
Lparam(l1) = 0.0078125  Lparam(l2) = 0.0390625  MSE = 3.34366e-05
Lparam(l1) = 0.0078125  Lparam(l2) = 0.0546875  MSE = 1.58012e-05
Lparam(l1) = 0.0078125  Lparam(l2) = 0.0703125  MSE = 9.5937e-06
Lparam(l1) = 0.0078125  Lparam(l2) = 0.0859375  MSE = 6.30486e-06
Lparam(l1) = 0.0078125  Lparam(l2) = 0.101562   MSE = 4.14427e-06
Lparam(l1) = 0.0078125  Lparam(l2) = 0.117188   MSE = 2.758e-06
Lparam(l1) = 0.0078125  Lparam(l2) = 0.132812   MSE = 1.90618e-06
Lparam(l1) = 0.0078125  Lparam(l2) = 0.148438   MSE = 1.38614e-06
Lparam(l1) = 0.0078125  Lparam(l2) = 0.164062   MSE = 1.05734e-06
Lparam(l1) = 0.0078125  Lparam(l2) = 0.179688   MSE = 8.41394e-07
Lparam(l1) = 0.0078125  Lparam(l2) = 0.195312   MSE = 7.00938e-07
Lparam(l1) = 0.0078125  Lparam(l2) = 0.210938   MSE = 6.17097e-07
Lparam(l1) = 0.0078125  Lparam(l2) = 0.226562   MSE = 5.7693e-07
Lparam(l1) = 0.0078125  Lparam(l2) = 0.242188   MSE = 5.69243e-07
Lparam(l1) = 0.0078125  Lparam(l2) = 0.257812   MSE = 5.83794e-07
Lparam(l1) = 0.0078125  Lparam(l2) = 0.273438   MSE = 6.1131e-07
Lparam(l1) = 0.0078125  Lparam(l2) = 0.289062   MSE = 6.43724e-07
Lparam(l1) = 0.0078125  Lparam(l2) = 0.304688   MSE = 6.74516e-07
Lparam(l1) = 0.0234375  Lparam(l2) = 0.0078125  MSE = 8.83975e-05
Lparam(l1) = 0.0234375  Lparam(l2) = 0.0234375  MSE = 3.08797e-05
Lparam(l1) = 0.0234375  Lparam(l2) = 0.0390625  MSE = 9.47581e-06
Lparam(l1) = 0.0234375  Lparam(l2) = 0.0546875  MSE = 4.17032e-06
Lparam(l1) = 0.0234375  Lparam(l2) = 0.0703125  MSE = 2.65846e-06
Lparam(l1) = 0.0234375  Lparam(l2) = 0.0859375  MSE = 1.82868e-06
Lparam(l1) = 0.0234375  Lparam(l2) = 0.101562   MSE = 1.16167e-06
Lparam(l1) = 0.0234375  Lparam(l2) = 0.117188   MSE = 6.83752e-07
Lparam(l1) = 0.0234375  Lparam(l2) = 0.132812   MSE = 3.88184e-07
Lparam(l1) = 0.0234375  Lparam(l2) = 0.148438   MSE = 2.23819e-07
Lparam(l1) = 0.0234375  Lparam(l2) = 0.164062   MSE = 1.33795e-07
Lparam(l1) = 0.0234375  Lparam(l2) = 0.179688   MSE = 8.11784e-08
Lparam(l1) = 0.0234375  Lparam(l2) = 0.195312   MSE = 4.93291e-08
Lparam(l1) = 0.0234375  Lparam(l2) = 0.210938   MSE = 3.14324e-08
Lparam(l1) = 0.0234375  Lparam(l2) = 0.226562   MSE = 2.34417e-08
Lparam(l1) = 0.0234375  Lparam(l2) = 0.242188   MSE = 2.25705e-08
Lparam(l1) = 0.0234375  Lparam(l2) = 0.257812   MSE = 2.76033e-08
Lparam(l1) = 0.0234375  Lparam(l2) = 0.273438   MSE = 3.84122e-08
Lparam(l1) = 0.0234375  Lparam(l2) = 0.289062   MSE = 5.48382e-08
Lparam(l1) = 0.0234375  Lparam(l2) = 0.304688   MSE = 7.59281e-08
```



```
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=20 ./GPR_Project
Lparam(l1) = 0.304688   Lparam(l2) = 0.0390625  MSE_min = 2.74202e-07
GPR exec time: 816779ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=16 ./GPR_Project
Lparam(l1) = 0.304688   Lparam(l2) = 0.0390625  MSE_min = 2.74202e-07
GPR exec time: 901849ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=8 ./GPR_Project
 Lparam(l1) = 0.304688  Lparam(l2) = 0.0390625  MSE_min = 2.74202e-07
GPR exec time: 805989ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=4 ./GPR_Project
Lparam(l1) = 0.304688   Lparam(l2) = 0.0390625  MSE_min = 2.74202e-07
GPR exec time: 818826ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=1 ./GPR_Project
Lparam(l1) = 0.304688   Lparam(l2) = 0.0390625  MSE_min = 2.74202e-07
GPR exec time: 908154ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=2 ./GPR_Project
Lparam(l1) = 0.304688   Lparam(l2) = 0.0390625  MSE_min = 2.74202e-07
GPR exec time: 853160ms
```

- Speed up and efficiency achieved:

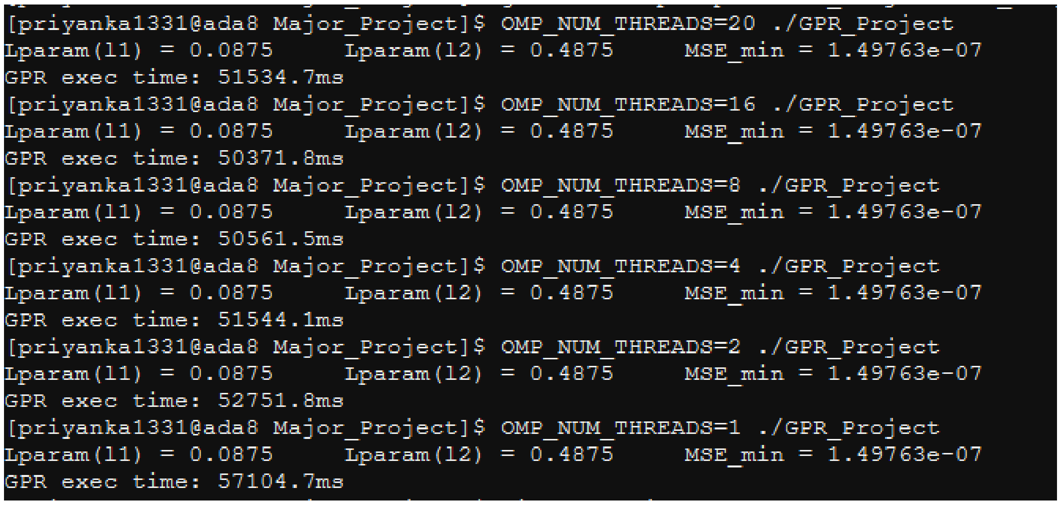
  m = 32

  #threads = 1, time : 908.2 s

  #threads = 8, time : 805.9 s

  Speed up = 1.12

  Efficiency = 14%



```
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=20 ./GPR_Project
Lparam(l1) = 0.0875     Lparam(l2) = 0.4875     MSE_min = 1.49763e-07
GPR exec time: 51534.7ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=16 ./GPR_Project
Lparam(l1) = 0.0875     Lparam(l2) = 0.4875     MSE_min = 1.49763e-07
GPR exec time: 50371.8ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=8 ./GPR_Project
Lparam(l1) = 0.0875     Lparam(l2) = 0.4875     MSE_min = 1.49763e-07
GPR exec time: 50561.5ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=4 ./GPR_Project
Lparam(l1) = 0.0875     Lparam(l2) = 0.4875     MSE_min = 1.49763e-07
GPR exec time: 51544.1ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=2 ./GPR_Project
Lparam(l1) = 0.0875     Lparam(l2) = 0.4875     MSE_min = 1.49763e-07
GPR exec time: 52751.8ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=1 ./GPR_Project
Lparam(l1) = 0.0875     Lparam(l2) = 0.4875     MSE_min = 1.49763e-07
GPR exec time: 57104.7ms
```

m = 20

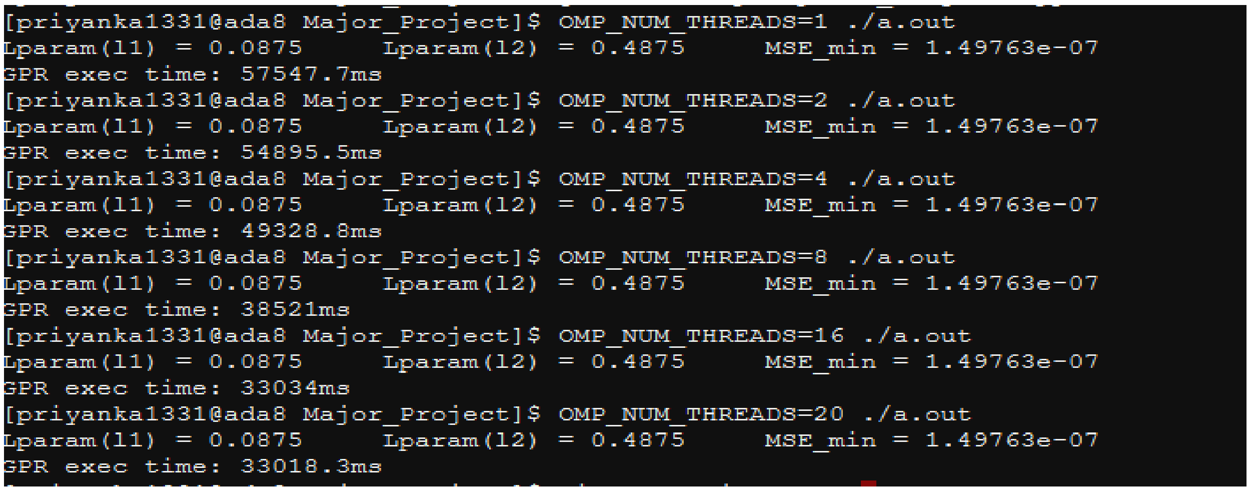
#threads = 1, time : 57.104 s

#threads = 8, time :50.561 s

Speed up = 1.14

Efficiency = 14.25%

The below output is if I enable #pragma omp parallel for ordered scheduled(static,2).



```
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=1  ./a.out
Lparam(l1) = 0.0875      Lparam(l2) = 0.4875      MSE_min = 1.49763e-07
GPR exec time: 57547.7ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=2  ./a.out
Lparam(l1) = 0.0875      Lparam(l2) = 0.4875      MSE_min = 1.49763e-07
GPR exec time: 54895.5ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=4  ./a.out
Lparam(l1) = 0.0875      Lparam(l2) = 0.4875      MSE_min = 1.49763e-07
GPR exec time: 49328.8ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=8  ./a.out
Lparam(l1) = 0.0875      Lparam(l2) = 0.4875      MSE_min = 1.49763e-07
GPR exec time: 38521ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=16 ./a.out
Lparam(l1) = 0.0875      Lparam(l2) = 0.4875      MSE_min = 1.49763e-07
GPR exec time: 33034ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=20 ./a.out
Lparam(l1) = 0.0875      Lparam(l2) = 0.4875      MSE_min = 1.49763e-07
GPR exec time: 33018.3ms
```

m = 20

#threads = 1, time : 575.4ms

#threads = 16, time :330.3 ms

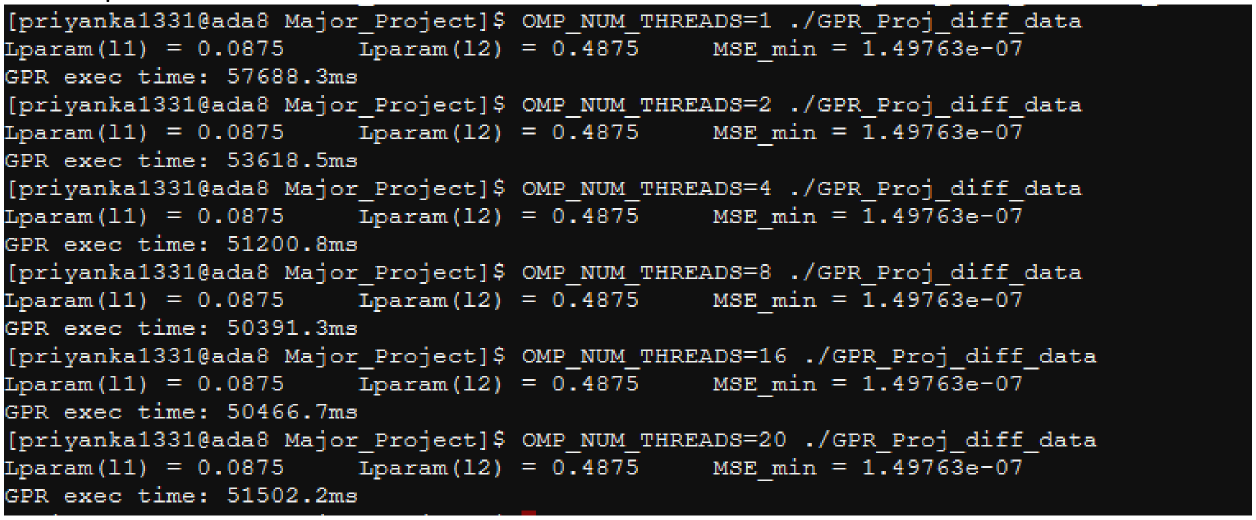
Speed up = 1.742

Efficiency = 11%

The algorithm performs consistently for the second dataset as well.

3. Created new dataset by changing f = f + kernel(XY, [0.25,0.25], [2; 2]/m) + XY * [0.2; 0.1] to f = kernel(XY, [0.5,0.5], [1; 1]/m) + XY * [0.2; 0.1]. The MSE is also very low similar to previous dataset.

The output is as shown below for m = 20.



```
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=1 ./GPR_Proj_diff_data
Lparam(l1)  = 0.0875      Lparam(l2)  = 0.4875      MSE_min = 1.49763e-07
GPR exec time: 57688.3ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=2 ./GPR_Proj_diff_data
Lparam(l1)  = 0.0875      Lparam(l2)  = 0.4875      MSE_min = 1.49763e-07
GPR exec time: 53618.5ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=4 ./GPR_Proj_diff_data
Lparam(l1)  = 0.0875      Lparam(l2)  = 0.4875      MSE_min = 1.49763e-07
GPR exec time: 51200.8ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=8 ./GPR_Proj_diff_data
Lparam(l1)  = 0.0875      Lparam(l2)  = 0.4875      MSE_min = 1.49763e-07
GPR exec time: 50391.3ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=16 ./GPR_Proj_diff_data
Lparam(l1)  = 0.0875      Lparam(l2)  = 0.4875      MSE_min = 1.49763e-07
GPR exec time: 50466.7ms
[priyanka1331@ada8 Major_Project]$ OMP_NUM_THREADS=20 ./GPR_Proj_diff_data
Lparam(l1)  = 0.0875      Lparam(l2)  = 0.4875      MSE_min = 1.49763e-07
GPR exec time: 51502.2ms
```

The performance is very similar to the above dataset for same size, m = 20.

m = 20

#threads = 1, time : 57.6 s

#threads = 8, time :50.3 s

Speed up = 1.15

Efficiency = 14.3%