Programming Assignment #5

1. Hardware

The CSL machine was used for compilation, execution, and benchmarking.

- Processor model: Intel® Core™ i5-11500
- Number of cores: 6
- Number of threads: 12
- 50 GB/s max memory bandwidth

2. Operating System and Compilation:

The operating system was Ubuntu 22.04.2 LTS

The intel compiler on the CSL machine was activated using the command:

```
source /s/intelcompilers-2019/bin/iccvars.sh intel64
```

Commands used for compilation and execution:

```
icc *.cpp -mkl -qopenmp -o main
./main
```

3. Code Changes

Part A:

For the Minimum Degree Algorithm, only iparm[1] needed to be changed. For the “no permutation” scenario, a vector was created containing numbers 0 to n - 1 (where n is XDIM*YDIM*ZDIM). This was then passed into every call to pardiso and iparm[4] was set to 1 to indicate that the user is providing a permutation.

Part B:

A parameter called K_NUM_RHS was created for the number of right hand sides. Two new arrays were created: xMultiple and bMultiple which were of size (XDIM*YDIM*ZDIM*K_NUM_RHS). These were then filled with the values of x and f that are passed into the direct solver. The same values were repeatedly filled K_NUM_RHS times as this allowed assessment of the performance impact of this modification. The variable nrhs was set to K_NUM_RHS.

4. Performance Statistics

The parameters were changed to XDIM=32, YDIM=32, ZDIM=32

Part A:

Scenario	Number of non-zeroes in L	GLOPS for numerical factorization	GLOPS/s for numerical factorization	Time for solve step (s)
Nested Dissection	3835218	2.10	24.19	0.0198
Minimum Degree Algorithm	6043324	5.77	10.22	0.0503
No permutation	23561947	20.99	3.46	0.0640

Part B:

The reordering was done by nested dissection (default). “idum” was passed in instead of “perm” and the values of iparm were reset to what they were orignally. The parameters were changed to XDIM=64, YDIM=64 ZDIM=64

Number of rhs (k)	Time for solve step (s)
1	0.0417
10	0.1533
100	0.3521

5. Comments

Part A:

As expected, the nested dissection reordering was the best for every metric. The minimum degree algorithm had twice as many non-zeroes and was about 2.5x worse for every other metric. Also, doing no permutation yielded the worst results for every metric: 6x worse sparsity, 10x worse GLOPS for factorization and 3x more time for solve step.

Part B:

As expected, the time for the solve step increasing sub linearly as the number of rhs was changed. When increasing from 1 to 10, the time increased by about 4x. When increasing from 10 to 100, the time increasing by about 2.5x. This shows that PARDISO takes advantage of the extra rhs and makes the execution more compute bound rather than memory bound.