

Project1 Final Report

This report is in continuation of “M2 Graphs and Docs” in milestone 2 folder.

project1final.c and project1varvel.c are the final files for the normal conditions and condition with variable velocity respectively.

In the code, for the inputs, 0th rank is taking all the inputs and broadcasting it to the other ranks. All ranks are working on their small chunks and at the end of the code, all the ranks are sending data to 0th rank for it to print to the file. In addition to this, parallelization has also been added to cater to for loops for each time stamp.

Sample code for executing the file –

```
mpicc -fopenmp -DPARALLEL -o test ./project1final.c -lm  
mpirun -np 4 ./test 100 20000 1 1e6 5e-7 2.85e-7 4 1
```

Here, 4 after np denotes the number of processors.

./test is the compiled file

The parameters in the middle are in the same order as directed in the first milestone of the assignment

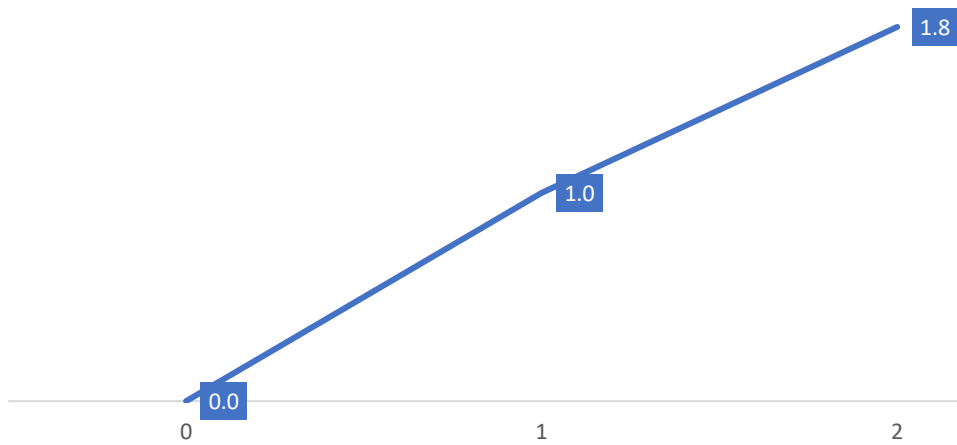
The second last parameter is the number of parallel threads

Last parameter is 1 for Lax method, 2 for first order and 3 is for second order.

Below is the strong scaling graphs for 4 parallel threads and N = 10000.

Ranks	1	4	16
Log(Ranks)	0	1	2
Time (s)	5982.1	1489.2	491.7
Strong Scaling	1.0	4.0	12.2
Log(Strong Scaling)	0.0	1.0	1.8
Grind Rate (B)	0.3	1.3	4.1

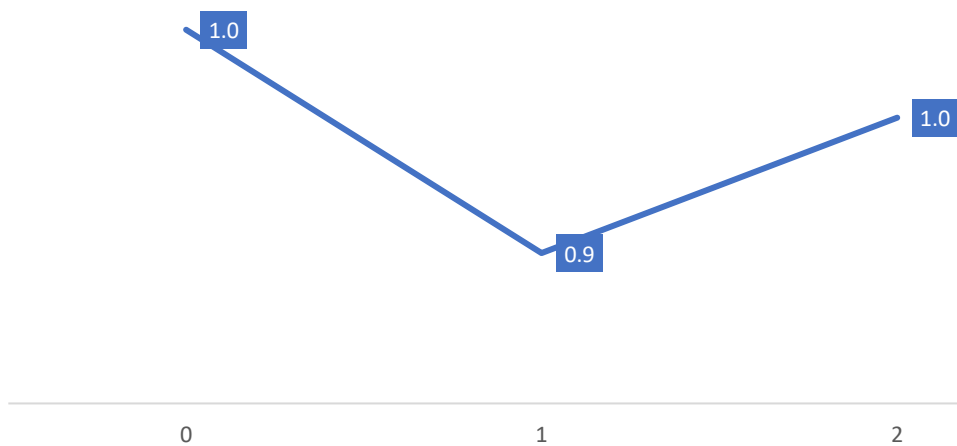
Strong Scaling, $N = 10,000$: $\log(S_n)$ vs $\log(\text{Ranks})$



Below is the weak scaling graph for 4 parallel threads for values of N mentioned in the table

Ranks	1	4	16
Log(Ranks)	0	1	2
N	4000	8000	16000
Time (s)	1524.5	1360.5	1460.0
S_n	1	0.9	1.0

Weak Scaling: $\log(S_n)$ vs $\log(\text{Ranks})$



From 1 MPI Rank per core to no use of OpenMP, the time decreases by about 60%. For 4 parallel threads, the time taken for 1 MPI Rank per core is around 6k seconds, where as without OpenMP, its around 3.6k seconds.

For the non uniform, I have taken u proportional to i and v proportional to j in our grid