Report: **Project 3- MPI** (CSE 6230)

-- by Harsh Shrivastava (gtID: 903241811)

Report is divided into 4 parts:

Part A: Correctness

1. The code always converges with all possible mesh arrangements
2. Code is scalable as the max size allocated via malloc is (m+2*grow)^2
3. Please refer the graphs for execution timings

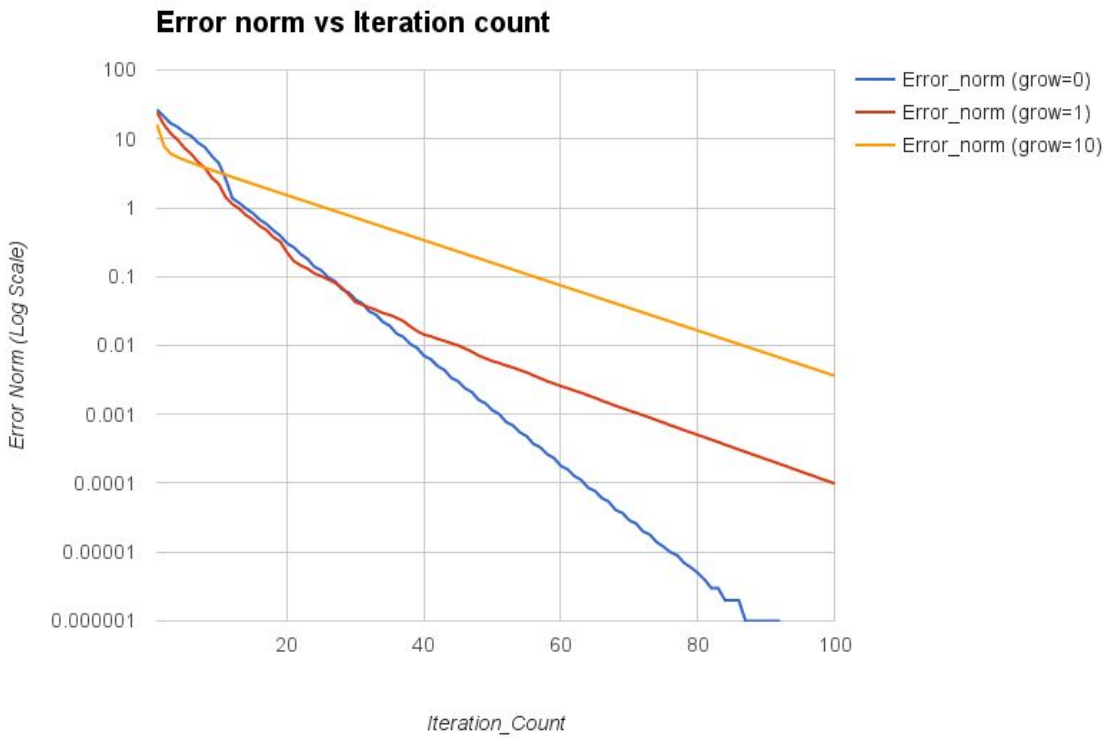
(Results shown are obtained from running on **'mic2'** coprocessor)

NOTE: Run using the following steps

1. Set M, P, Q, GROW in 'jacobi_schwarz.c' file
2. Set NUM_PROCESS in 'makefile'
3. make jacobi_schwarz_mic
4. Make runmic

Part B: Error Norm Graphs

M=100, P=10, Q=6, Grow={0,1,10}



For the grow = {1, 10} cases, we can observe a considerable dip in the error norm plot during the initial iterations. But, after few iterations the gradient decreases and slope for error reduction becomes more flat.

Part C: Table of time and number of iterations to reduce error norm to 10^-3

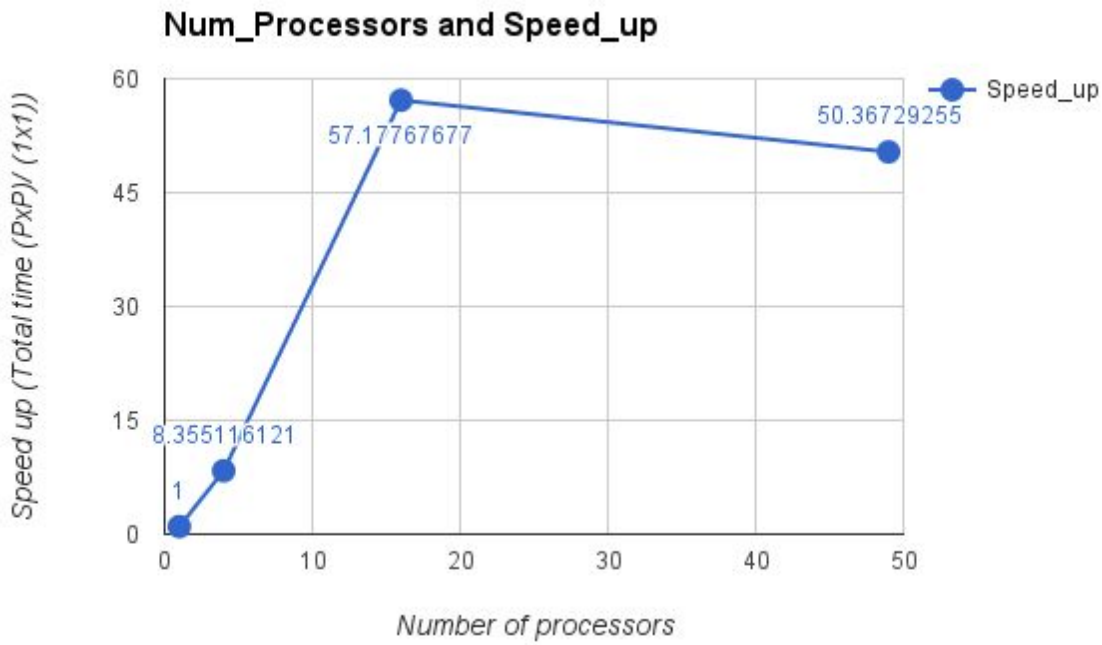
(pxq), m, grow	Time per Iteration (secs)	Number of Iterations	Total runtime (secs)
(1x1), 700, 0	5124.516837	1	5124.516837
(2x2), 350, 1	37.588231	9	338.294082
(4x4), 175, 1	1.267523	39	49.433389
(7x7), 100, 1	0.935292	60	56.117496

The (1 by 1) mesh case takes a considerable amount of time. Some possible reasons for this is

1. Maybe the mic2 was not completely available
2. DSS_unsym.c solver used for sparse matrix calculations may not be optimized enough to handle large matrix sizes efficiently.
3. The time taken for the laplacian2D function maybe the bottleneck.

Part D: Speed up graph

Relative to 1 process: For calculation of speed up, total runtime was considered.
Speed up = total_time(1 by 1 mesh) / total_time(‘p’ by ‘p’ mesh) [to reduce to 10^-3 error]



The speed up is going over 49 times. Probably the overhead time to run the process with 1x1 mesh is quite high, hence the results. Also, ideally **speedup should be over the runtime of best serial algorithm** and I’m pretty sure that there can be better implementation for serial version compared to my code. This is probably the reason, the speedup is more than the number of processors used.