Computational Fluid Dynamics Lab
Worksheet 3
Group B

1 Validation

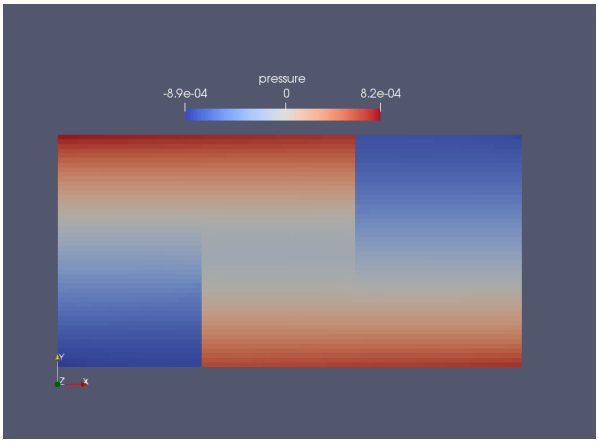
1.1 Convergence rate check:

Case	Size	iproc	jproc	Average number of the iterations needed for convergence
Lid-driven cavity	300x300	1	1	
Lid-driven cavity	300x300	2	2	
Lid-driven cavity	300x300	1	4	
Lid-driven cavity	300x300	2	3	
Fluid Trap	100x50	1	1	4
Fluid Trap	100x50	2	3	10

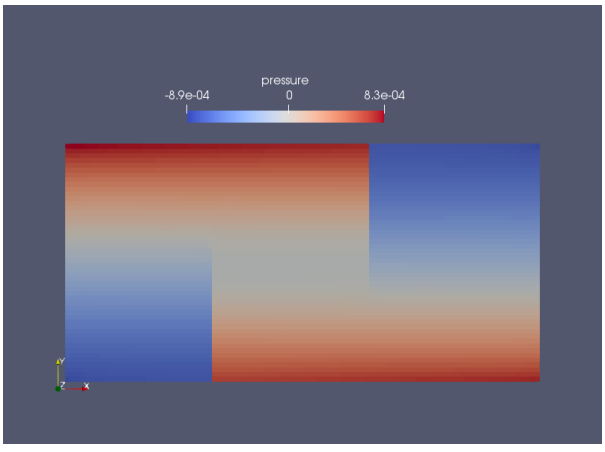
We observed that the number of time steps it take to go towards convergence increase with the number of threads. In other words, more and more time steps issued the warning that the SOR iterations exceeded the maximum limit when we increase the number threads. This should be due to error(s) introduced when the algorithm was converted to work in parallel, which needs to be investigated.

1.2 Figures

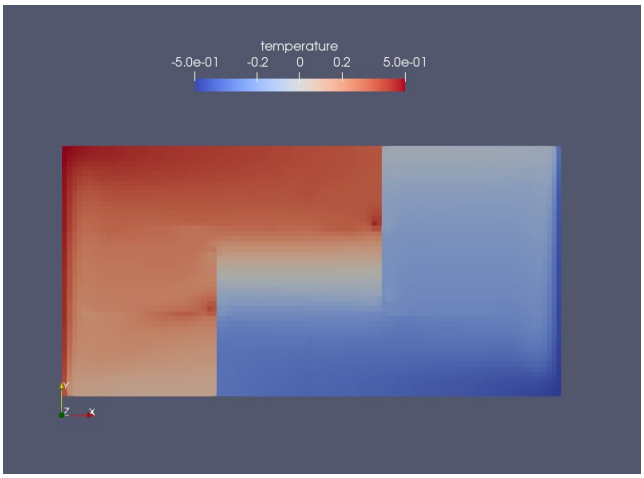
1.2.1 Fluid trap results



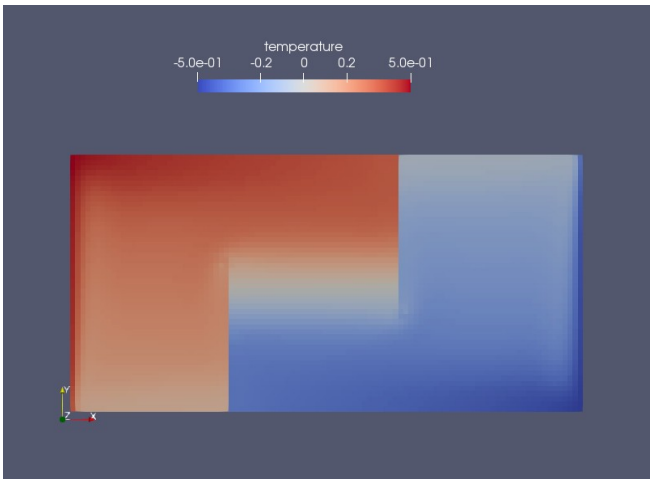
Pressure field (iproc, jproc) = (2, 3)



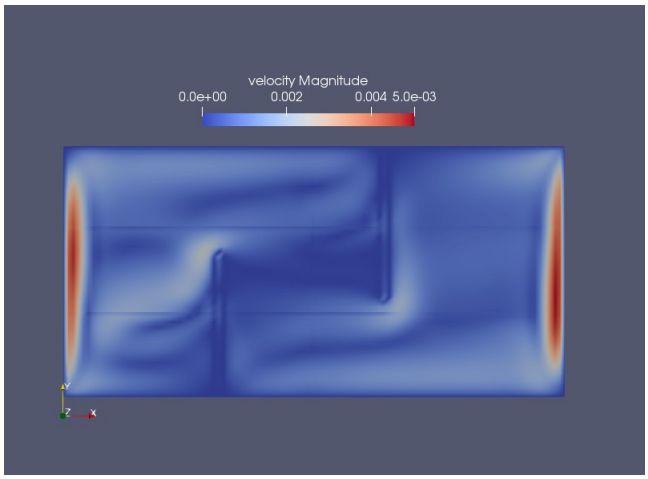
Pressure field (iproc, jproc) = (1, 1)



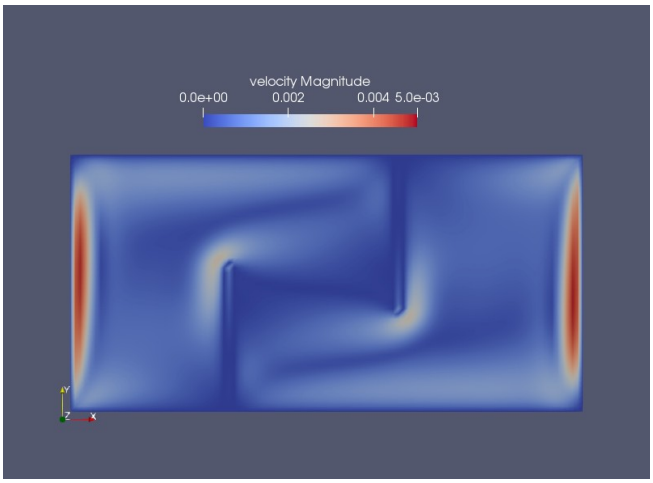
Temperature field (iproc, jproc) = (2, 3)



Temperature field (iproc, jproc) = (1, 1)



Velocity field (iproc, jproc) = (2, 3)



Velocity field (iproc, jproc) = (1, 1)

2 Perfomance Scaling

We are unable to run the Rayleigh-Benard with a parallelization because of the bug that we have not yet figured out. It causes a huge residual, thus also number of iteration and make the code slower. To test our parallelization for the strong scaling analysis, we use Lid-Driven Cavity instead.
In this case, the SOR solver does not converge fast enough. Therefore it took a long to time to get the results for different sizes, especially, for 100x100 and finer grids. Due to the lack of time, we could not get all the results to compare.

1.2 Strong Scaling Analysis from the Lid Driven Cavity

Size	Threads	Time (s)	Speedup	Parallel Efficiency
50x50	1	266.52	-	-
50x50	2	130.65	2.04	1.02
50x50	4	111.05	2.40	0.60

1.3 Weak Scaling Analysis from the Lid Driven Cavity

Size	Threads	Time (s)	Parallel Efficiency	% Time Increase
50x50	1	1.04	-	1
100x100	4	10.01	0.42	2.4
200x200	16	210.15	0.08	12.62
300x300	36	781.25	0.05	20.84
400x400	64	1702.05	0.04	25.54

The MPI implementation for the given problem doesn’t perform well with weak scaling. This is due to the increasing overhead for communication compared to computation when the problem is scaled.