Readme for Excercise 3 of CFD Master Praktikum (Group G)

Yue Zhu, Yingqiang Gao and Nikhil Agarwal

June 4, 2018

Contents

1 Instructions about compilation, execution, and report generation

1.1 Compilation and Execution

Please start a terminal and switch to where the `main.c` is. Build it first by typing `make` in the terminal. Then if everything is right, find the executable `sim` being created and run with the following command:

- `mpiexec -n <N> ./sim` for `MPICH` where `N` is the product of the number of MPI processes along x and y axis.
- `mpirun --hostfile my-hosts -np <N> ./sim` for `Open-MPI` where `N` is the product of the number of MPI processes along x and y axis.

Example: To execute the binary with 2 process along x direction and 3 along y with MPICH, set these values in `cavity100.dat` and run as follows

```
mpiexec -n 6 ./sim
```

1.2 Report Generation

The markdown file (and generated html) loads image if the screenshot generated from the `Paraview` are stored at `.\..\output` location relative to README.rmd.

Use a markdown file viewer to view the report or it can be converted to html or pdf using suitable tools. `Github` automatically generates the view from markdown.

2 Problem:

2.1 Parameters

imax = 300	jmax = 300	xlength = 10	ylength = 10
dt = 0.01	t_end = 1	tau = 0.5	dt_value = 2.0
eps = 0.01	omg = 1.7	alpha = 0.5	itermax = 100
GX = 0.0	GY = 0.0	Re = 10	
UI = 0.0	VI = 0.0	PI = 0.0	
iproc = 2	jproc = 3		

2.2 Geometry

Driven cavity of square cross-section area. The domain is a container filled with a fluid with the container lid (a band or a ribbon) moving at a constant velocity. No-slip conditions are imposed on all four boundaries, with the exception of the upper boundary, along which the velocity u in x-direction is not set to zero, but is equal to 1, in order to simulate the moving lid.

2.3 Pressure

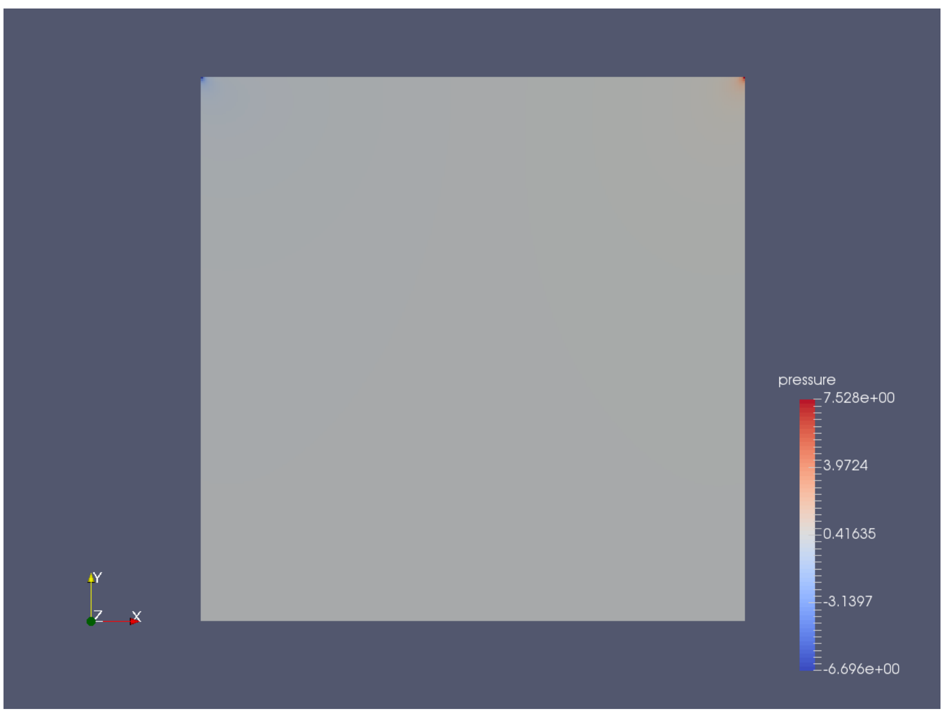


Figure 1: Pressure

Observation: The pressure is highest at the top right while lowest at top left, is consistent with the direction of fluid flow in the absence of thermal effects.

2.4 Streamlines

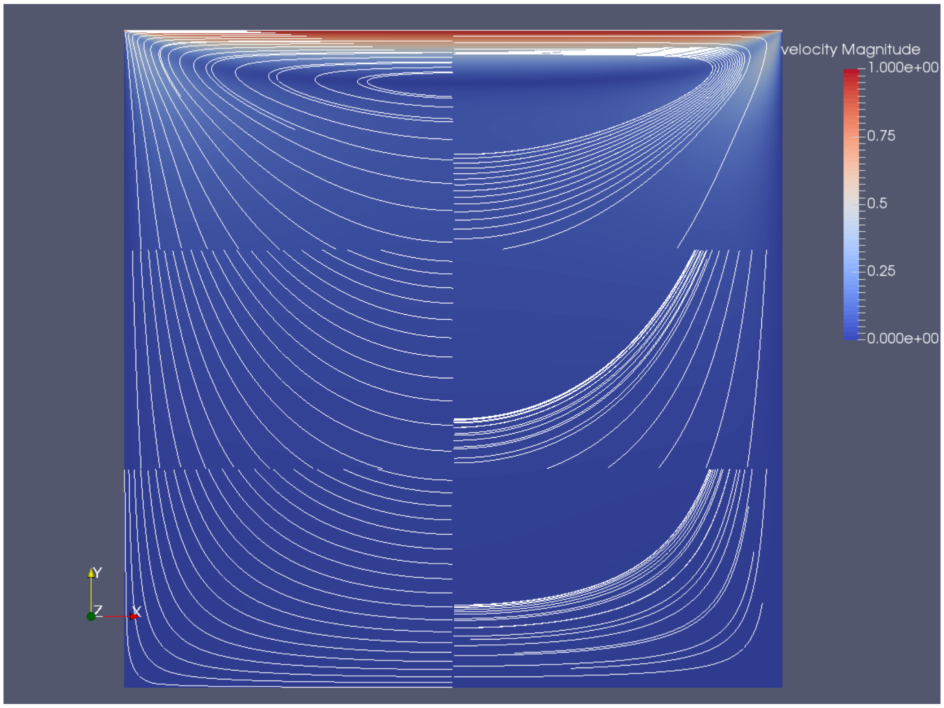


Figure 2: Streamlines

Observation: The streamlines are ploted for each subdomain independently. The path traced out by a massless particle as it moves with the flow, is matching at the interface of each subdomain. So, we can conclude that the streamlines are consistent with other subdomains.

2.5 Velocity

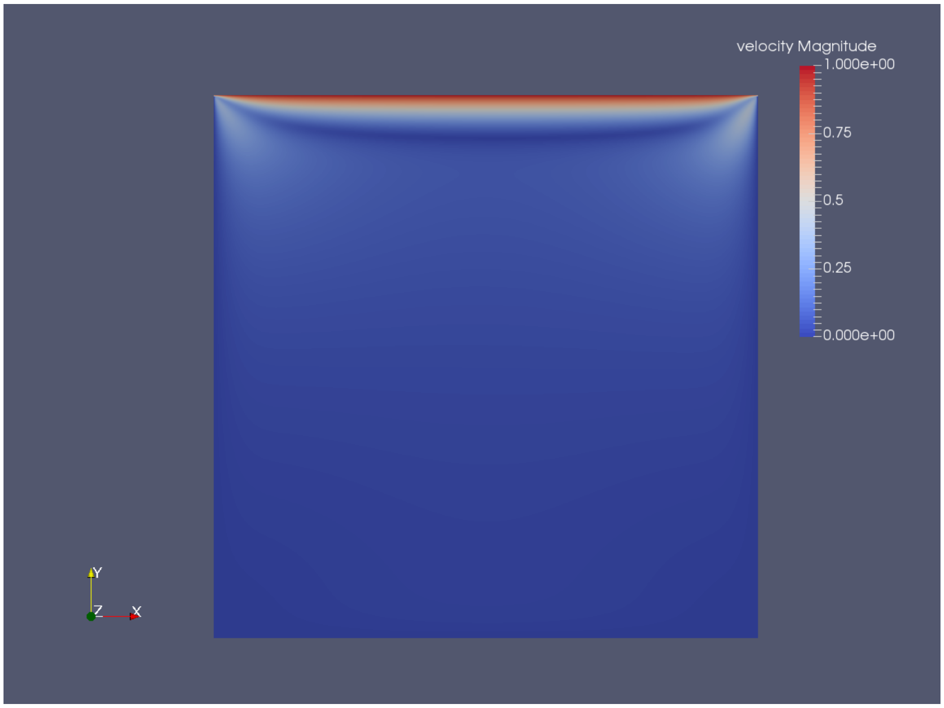


Figure 3: Velocity

Observation: As expected velocity is maximum at the top. Due to low Reynold's number there is less turbulence in the system (**when compared with Worksheet1**).

2.6 Performance

- Speedup, `S(p) := T(1)/T(p)`
- Parallel efficiency `E(p) := T(1)=(p * T(p)) = S(p)/p`

On Intel Processor i7-8550U (*henceforth refered to as **new machine***) with

- Nominal Frequency = 1.80GHz
- Single Core Frequency = 4.0 GHz
- # Core = 4
- # Threads/core = 2
- L3 Cache = 8 MB
- Memory Types = DDR4-2400, LPDDR3-2133
- Bus Speed 4 GT/s OPI

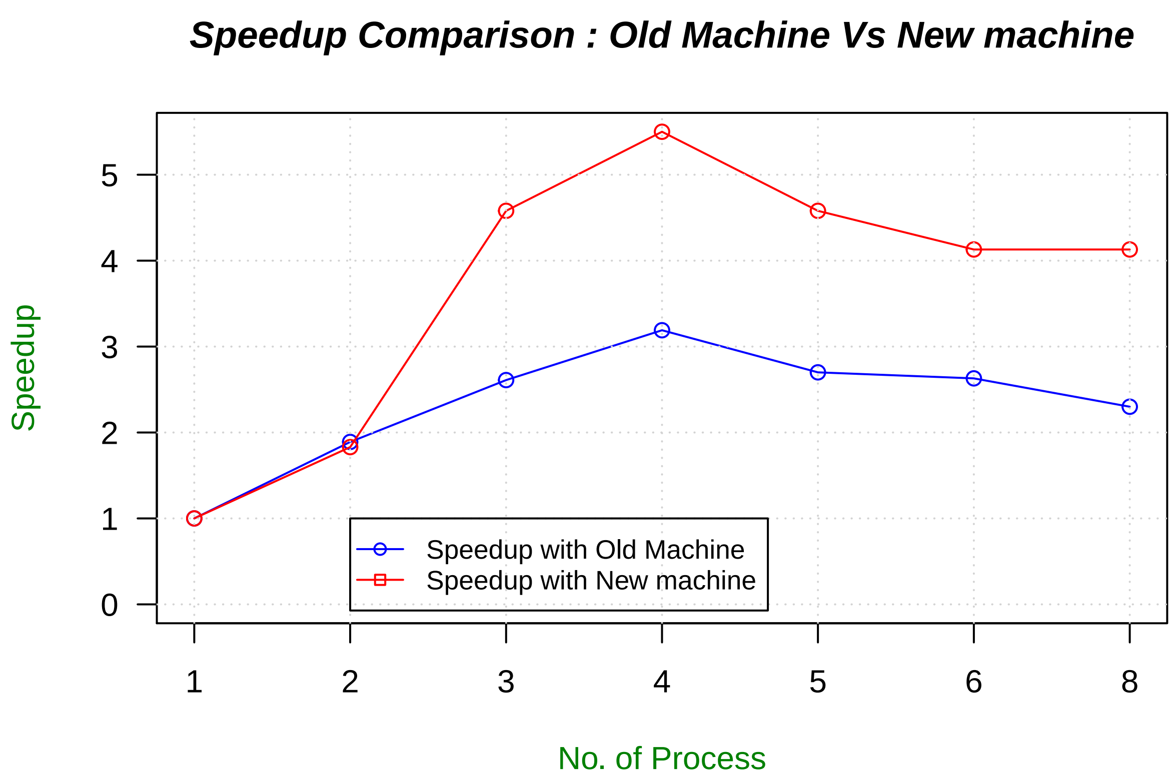
# Process (x)	# Process (y)	Total Process	time1 (s)	time2 (s)	time3 (s)	mean (s)	Std. Dev	Speedup	Efficiency
1	1	1	164.37	165.75	165.27	165.00	0.70	1.00	1.00
2	1	2	88.52	88.86	91.43	90.00	1.59	1.83	0.92
1	3	3	35.19	36.28	36.16	36.00	0.59	4.58	1.53
1	4	4	31.51	30.12	26.85	30.00	2.39	5.50	1.38

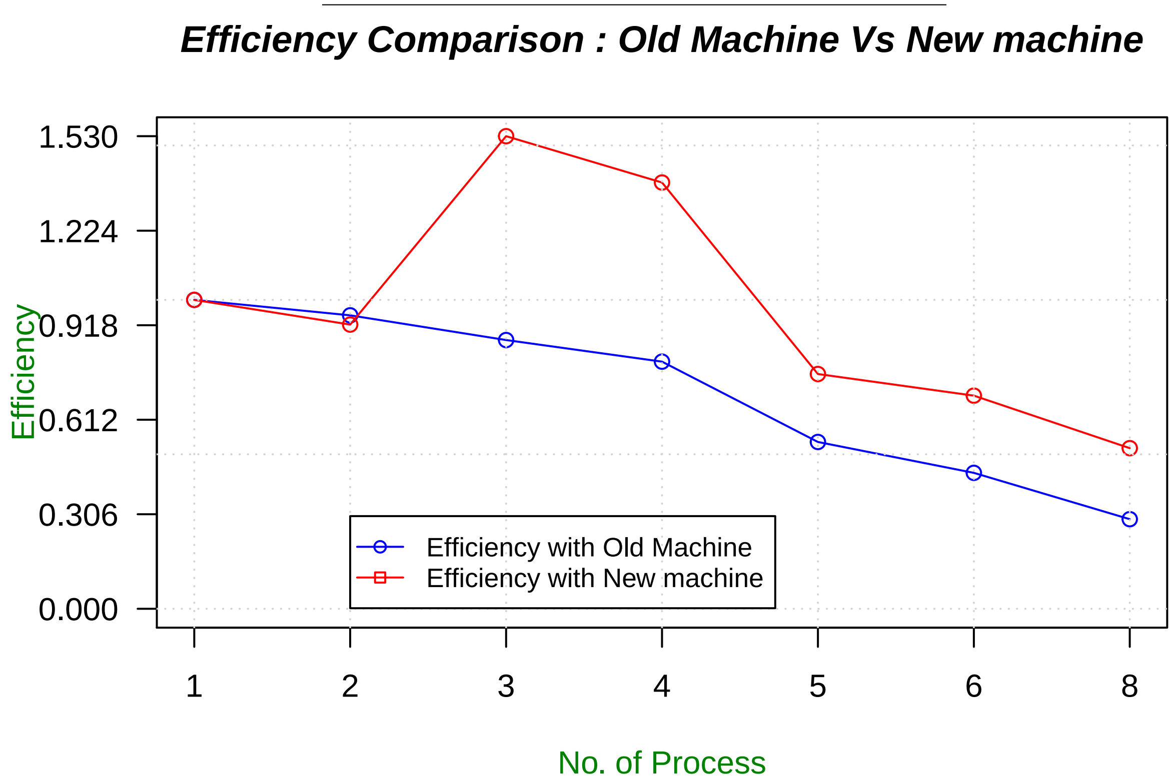
# Process (x)	# Process (y)	Total Process	time1 (s)	time2 (s)	time3 (s)	mean (s)	Std. Dev	Speedup	Efficiency
1	6	6	35.37	36.66	35.91	36.00	0.65	4.58	0.76
2	3	6	39.48	39.18	40.58	40.00	0.74	4.13	0.69
2	4	8	35.51	45.52	39.96	40.00	5.02	4.13	0.52
2	2	4	53.78	52.98	55.17	54.00	1.11	3.06	0.76
3	4	12	very slow						
4	5	20	very slow						
6	6	36	very slow						

On Intel Processor i7-4720HQ (*henceforth refered to as* ***old machine***) with

- Nominal Frequency = 2.60GHz
- Single Core Frequency = 3.6GHz
- # Core = 4
- # Threads/core = 2
- L3 Cache = 6 MB
- Memory Types = DDR3L 1333/1600
- Bus Speed 5 GT/s DMI2

# Process (x)	# Process (y)	Total Process	time1 (s)	time2 (s)	time3 (s)	mean (s)	Std. Dev	Speedup	Efficiency
1	1	1	115.13	112.08	114.09	113.77	1.55	1.00	1.00
2	1	2	60.56	58.85	61.00	60.13	1.14	1.89	0.95
1	3	3	42.97	43.86	44.00	43.61	0.56	2.61	0.87
1	4	4	36.93	33.60	36.60	35.71	1.83	3.19	0.80
1	5	5	39.74	42.55	44.18	42.16	2.24	2.70	0.54
1	6	6	40.89	42.86	45.99	43.25	2.57	2.63	0.44
2	4	8	45.42	50.26	52.96	49.55	3.82	2.30	0.29
2	2	4	36.34	34.51	36.50	35.78	1.11	3.18	0.79


Speedup Comparison : Old Machine Vs New machine
Speedup
0
1
2
3
4
5
1
2
3
4
5
6
8
No. of Process
Speedup with Old Machine
Speedup with New machine


Efficiency Comparison : Old Machine Vs New machine
Efficiency
0.000
0.306
0.612
0.918
1.224
1.530
1
2
3
4
5
6
8
No. of Process
Efficiency with Old Machine
Efficiency with New machine

Observation	Explanation
Single Core performance of *Old machine* with lesser frequency, lesser cache, slow RAM but faster Bus is better than that of the *Newer machine*	There should me further CPU characteristics (beyond CPU, Cache, RAM, Bus) that should have caused this result
When the total of Process becomes more the performance takes a severe hit!	As the resources are limited, *oversubscription of process* leads to *severe performance penalty.*
Max. Speed up is achieved when iproc = 1 and jproc = 4	Communication overhead and resource contention affects speedup for higher number of MPI process.
Superlinearity in efficiency seen with the execution with the *newer machine*	Probably due to more cache available in the machine (*Reference*: How to measure, present, and compare parallel performance - IEEE)

---

3 Addendum

3.1 Domain decomposition Visualized

See Figure *Domain Decomposition Visualized*

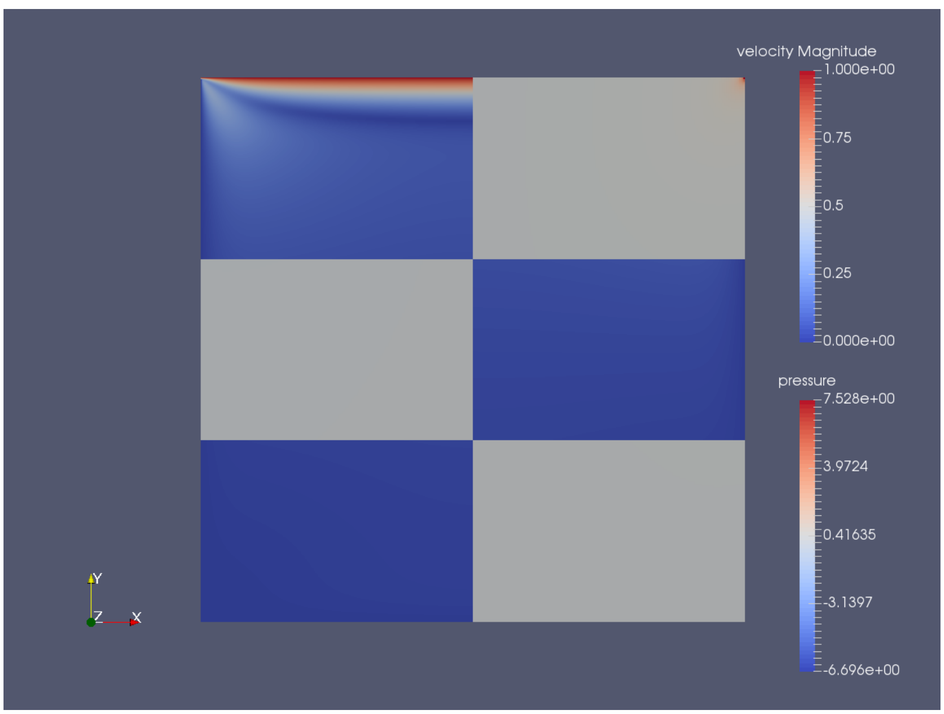


Figure 4: Domain Decomposition Visualized

3.2 Same Problem Visualized for Iproc = 2 and Jproc = 2

See Figure *Same Problem Visualized for Iproc = 2 and Jproc = 2*

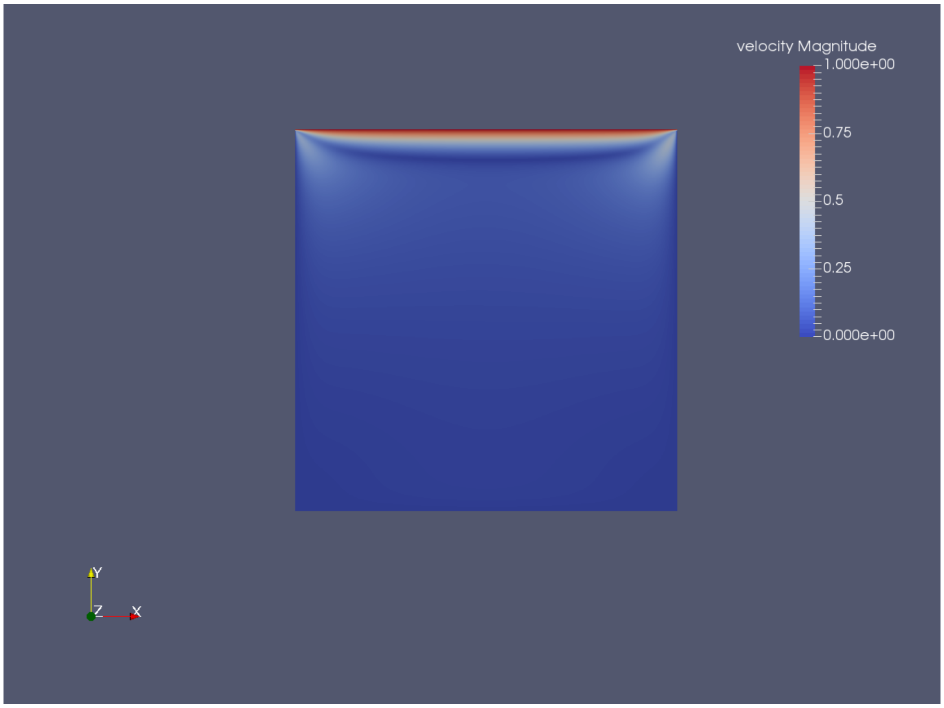


Figure 5: Same Problem Visualized for Iproc = 2 and Jproc = 2

3.3 Same Geometry but Parameters from Worksheet 1 Visualized for Iproc = 2 and Jproc = 2

See Figure *Same Geometry but Parameters from Worksheet 1 Visualized*

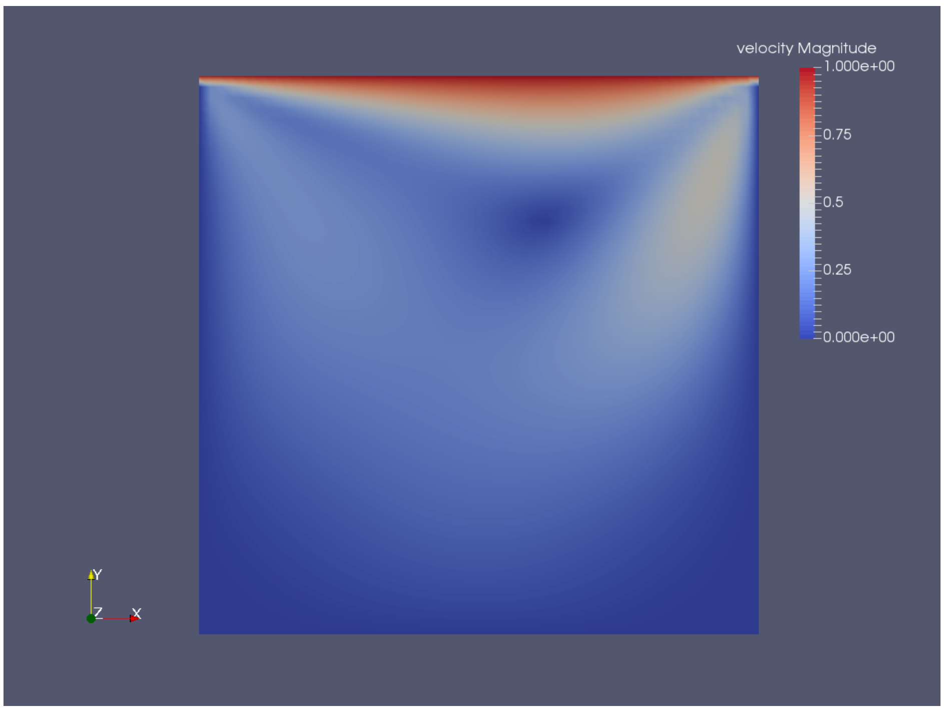


Figure 6: Same Geometry but Parameters from Worksheet 1 Visualized