

---

# **Lattice Boltzmann Method Solver Documentation**

*Release 0.0.1*

**Sayop Kim**

May 03, 2016



<b>1</b>	<b>Code Instruction</b>	<b>3</b>
1.1	Quick instruction for running the simulation . . . . .	3
<b>2</b>	<b>Results</b>	<b>5</b>
2.1	Problem1 - a . . . . .	5
2.2	Problem1 - b . . . . .	7
2.3	Problem1 - c . . . . .	12
2.4	Problem1 - d . . . . .	13
2.5	Problem1 - e . . . . .	13
2.6	Problem1 - f . . . . .	14
2.7	Problem1 - g . . . . .	16



This document is online available. The *pdf* format of this documentation may have some inappropriate image size and not-well organized order of image and its description. You are recommended to see this documentation visiting <http://cfm05.readthedocs.org>.

Contents:



---

## Code Instruction

---

The present project is aimed to develop a computer program for solving a steady solution with Lattice Boltzmann Method (LBM). The code being used for answering all the question here is written with Python language. This program is to run with simple command:

```
$ python main.py
```

### 1.1 Quick instruction for running the simulation

The Python code used for this project can be cloned from *github.com* repository:

```
$ git clone https://github.com/sayop/CFM05
```

You can also see the code directly by visiting the website: <https://github.com/sayop/CFM05> If you clone the code, you will see the following set of files and directories:

```
$ sayop@reynolds:~$ ls CFM05/  
docs  README.md  src
```

*docs* contains the document files set for the current project using *Sphinx* software. This *pdf* document is online available at: <http://cfm05-gatech.readthedocs.org>. The Python script for this simulation is stored in *src* folder.

Before running the simulation, you need to open the file named *input.in* using editor for example, VI on unix system:

```
$ vi input.in
```

Then, you should be able to see the following set of simulation parameters:

```
#grid dimension  
iDim      61  
jDim      61  
xmin      0  
xmax      60  
ymin      0  
ymax      60  
#boundary conditions  
uLeft     0.0  
vLeft     0.0  
uRight    0.0  
vRight    0.0  
uBottom   0.0  
vBottom   0.0  
uUp       10.0
```

```
vUp          0.0
# fluid kinematic properties
nu           1.0
pInit        10.0
#simulation setup
pCorr        1
alpha        1.0
pResidual    0.01
maxIter      1
Courant      0.5
dtInit       0.0001
Beta         0.5
residualMin   0.00005
#Post-Process
nIterWrite   100
```

The parameter's name above will literally tell you what every single variables indicates in the simulation. For the post-processing as requested in this project, *nIterWrite* will write a solution plot and CSV file at speicified interval of time integration number.



## Results

## 2.1 Problem1 - a

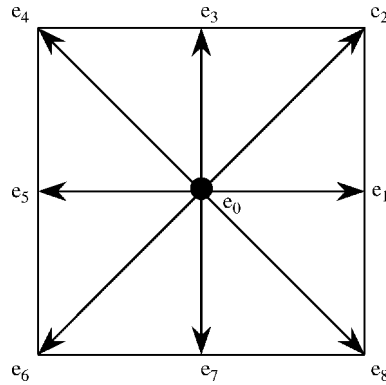
Describe the essential steps of the solution method. Include the discretized equations and implementation of boundary conditions.

### 2.1.1 Solution method

This project aims to use Lattice Boltzmann Method for solving the lid driven flow problem and assess its prediction accuracy and several numerical issues as stated in the problem description. In this project, D2Q9 lattice is employed to describe the stream the distribution fuctions along the specified directions. The directions in which 9-bit lattice BGK model evoloves defines following 9 discrete velocities:

$$e_i = \begin{cases} (0, 0) & i = 0 \\ (\cos[(i-1)\pi/2], \sin[(i-1)\pi/2]) & i = 1, 2, 3, 4 \\ \sqrt{2}(\cos[(i-5)\pi/2 + \pi/4], \sin[(i-5)\pi/2 + \pi/4]) & i = 5, 6, 7, 8 \end{cases}$$

The description of the lattice arrangement and the directions that particle streams are illustrated in the diagram below. Having 9 different velocity vectors, each of the 9 particles (fluid molecules) is moved to neighboring lattices.



Along the defined direction, 9 independent fluid particles are moved and re-located then update their distribution function in time. The evolution equation of the density distribution function is integrated in time as shown below. This step represents the collision process that changes the distribution function for each of 9 particles after streaming. This process will update the fluid macroscopic properties.

$$f_i(x_i + ce_i\Delta t, t + \Delta t) - f_i(x_i, t) = -\frac{1}{\tau} [f_i(x_i, t) - f_i^{\text{eq}}(x_i, t)]$$

where  $c = \Delta x / \Delta t$ , and  $\Delta x$  and  $\Delta t$  are the lattice spacing and the time step, respectively. Here,  $\tau$  is the dimensionless relaxation time that approximates the temporal rate at which instantaneous distribution function evolves and transitions to the equilibrium states. Given kinematic viscosity is taken into account for determining the relaxation time which is determined from following equation:

$$\nu = \frac{2\tau - 1}{6} \Delta x$$

And the  $f_i^{\text{eq}}$  expresses the equilibrium density function, which is determined by:

$$f_i^{\text{eq}}(x_i, t) = \omega_i \rho + \rho s_i(u_i(x_i, t))$$

where

$$s_i(u_i) = \omega_i \left[ e \frac{(e_i u_i)}{c} + 4.5 \frac{(e_i u_i)^2}{c^2} - 1.5 \frac{u_i u_i}{c^2} \right]$$

with the weight coefficient:

$$\omega_i = \begin{cases} \frac{4}{9} & i = 0 \\ \frac{1}{9} & i = 1, 2, 3, 4 \\ \frac{1}{36} & i = 5, 6, 7, 8 \end{cases}$$

After updating the distribution function, we are then to find fluid density and velocity from the integrated distribution function by assuming the relationship between the distribution function and macroscopic fluid properties:

$$\rho = \sum_i f_i$$

$$\rho u = \sum_i c e_i f_i$$

## 2.1.2 Boundary conditions

Treating boundary conditions for Lattice Boltzmann Method is very different from the way with other typical CFD simulation's approach that directly states flow properties at the boundary. Since the LBM plays with distribution function and derives the macroscopic flow properties from it, the boundary condition should also be treated with specific manipulation of distribution at the boundary lattices.

The issue that arise with the LBM is that there is no further lattices out of the computational domain that accomodate the streamed particles. To cope with this problem, a mid grid bounce-back boundary condition is used to replace the normal 9 direction streaming. On the edge lattices, we assume there is an imaginary lattices right next to the boundary, and when the particles move towards these lattices, it will be bounce back to their original lattice with inversed direction.

For the specific boundary condition of our moving lid on the upper wall, an equation of Zou-He velocity boundary conditions are employed and the specific treatment on this lattice can be made by manipulating the distribution functions as:

$$\begin{aligned} f_2 + f_5 + f_6 &= \rho - (f_4 + f_7 + f_8 + f_0 + f_1 + f_3) \\ f_5 - f_6 &= \rho u - (f_1 + f_8 - f_3 - f_7) \\ f_2 + f_5 + f_6 &= \rho v + (f_4 + f_7 + f_8) \\ \rho &= \frac{1}{1-v} ((f_0 + f_1 + f_3) - 2(f_4 + f_7 + f_8)) \end{aligned}$$

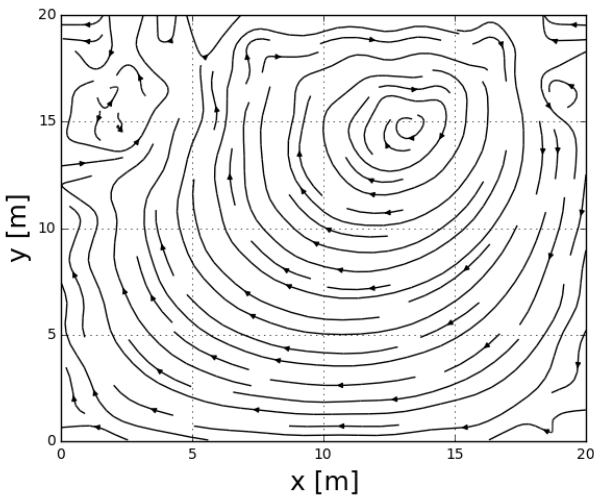
## 2.2 Problem1 - b

Consider the case when  $H = W$  (a square cavity). Here, the Reynolds number,  $Re = UW/\nu$ , characterizes the flow patterns. Compute the steady state solutions for both  $Re = 100$  and  $Re = 500$ . Plot the flow streamlines and centerline profiles ( $u$  vs.  $y$  and  $v$  vs.  $x$  through the center of the domain). For  $Re = 100$ , validate your method by comparing your results to data from given literature.

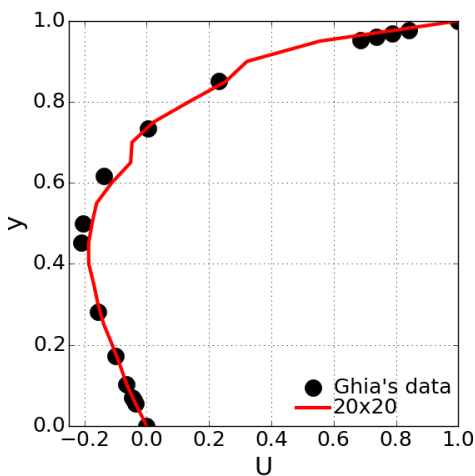
### 2.2.1 Re = 100

In this test, the lid cavity's velocity is set to make the Reynolds number set to 100. To see the qualitative effect of different grid spacing, four different grid resolution conditions is employed and compared together in this page.

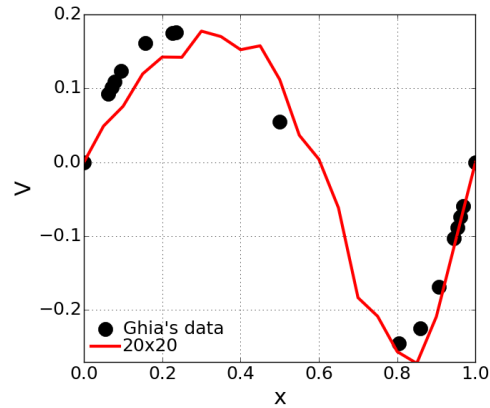
- $N \times N = 20 \times 20$



<Streamlines of 20x20 case runs>



<Centerline u-velocity compared with Ghia's numerically resolved data>

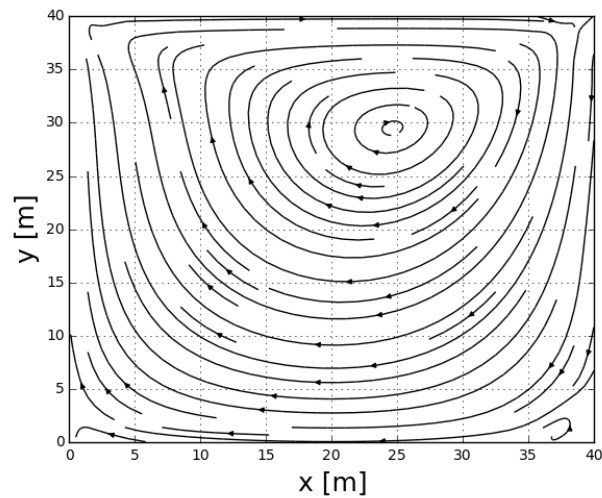


<Centerline v-velocity compared with Ghia's numerically resolved data>

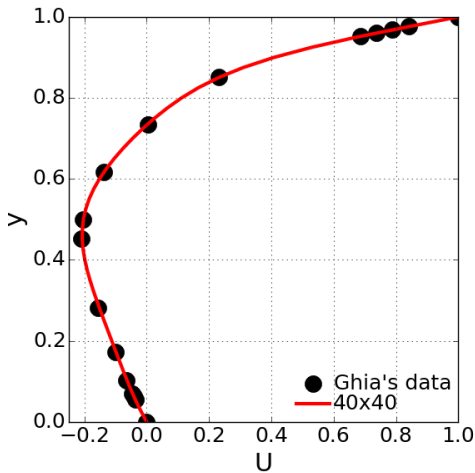
- **Observation**

- Coarser grid is horrible at achieving well-defined velocity field due to numerical errors and unstable solution.
- Despite this instability and error, the qualitative flow pattern is well resolved.
- The possible explanation of this inaccurate solution may be made with an argument that particle streaming should be made across lattices closely apart each other.
- Far distance between neighboring lattices may not represent proper particle streaming process.

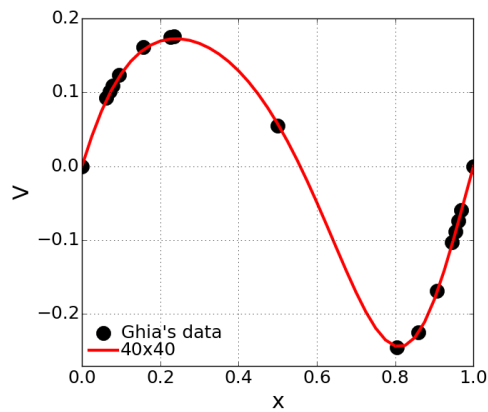
- $N \times N = 40 \times 40$



<Streamlines of 40x40 case runs>



<Centerline u-velocity compared with Ghia's numerically resolved data>

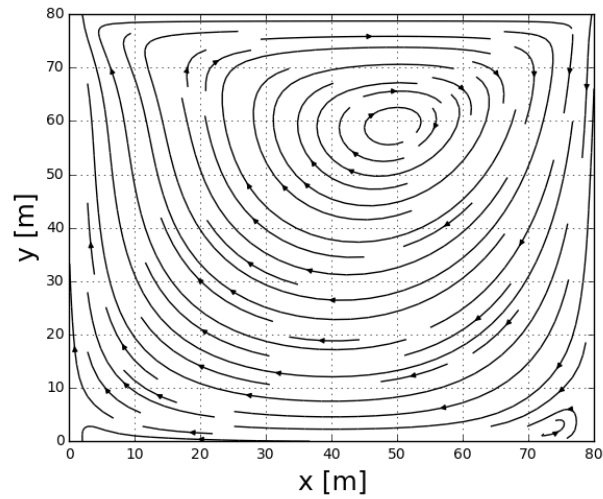


<Centerline v-velocity compared with Ghia's numerically resolved data>

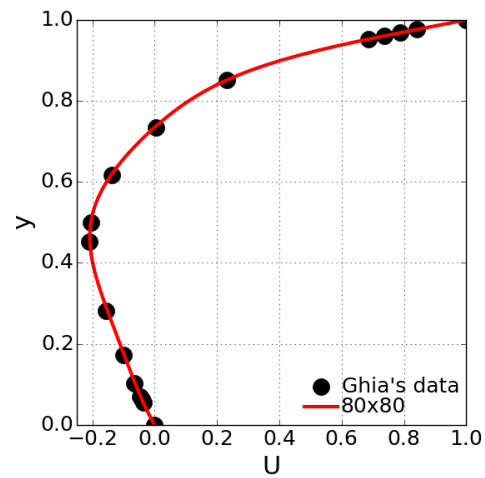
- **Observation**

- Grid resolution for this problem should be made up with 40x40 at least in order to have well-resolved flow properties with reasonable accuracy.
- The simulation data is in a good agreement with the reference data produced by Ghia et al.

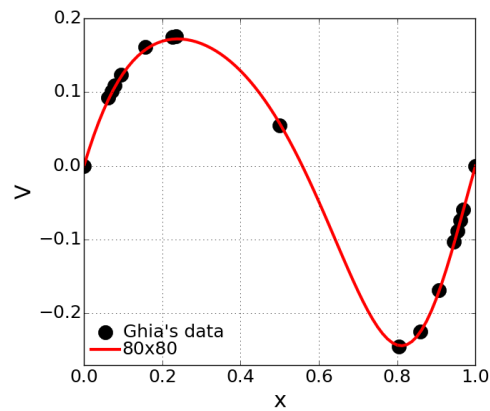
- $N \times N = 80 \times 80$



<Streamlines of 60x60 case runs>



<Centerline u-velocity compared with Ghia's numerically resolved data>



<Centerline v-velocity compared with Ghia's numerically resolved data>

- **Observation**

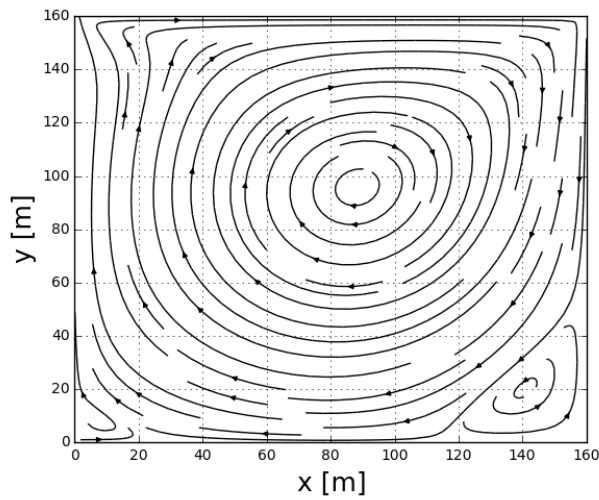
- Having resolution of 80x80 makes finally the resolved data looks very close to the reference data.
- We observed the denser grid size produces the more well-matching data with reference data.

### 2.2.2 Re = 500

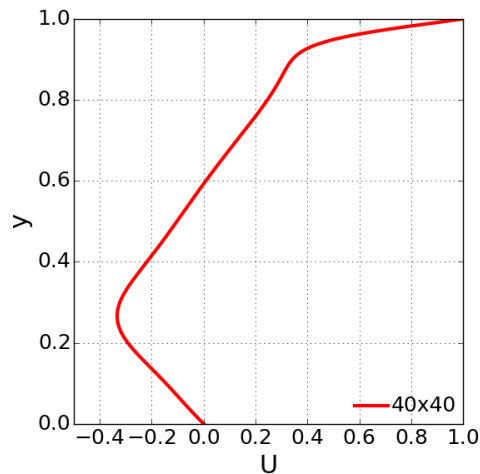
In this test, the lid cavity velocity is set to 50 m/s to make the Reynolds number 500. Here, only one single set of grid space was employed for effective simulation running, because this high Reynolds number case was strongly sensitive to the grid resolution, so it was not successful with less grid points than this to have such that stable solution. Again, due to the computational time issue, having more than the current grid points were not effective either.

The chosen grid resolution is 160x160 in x and y directions. The computation time taken for this setup is 32151.1 sec.

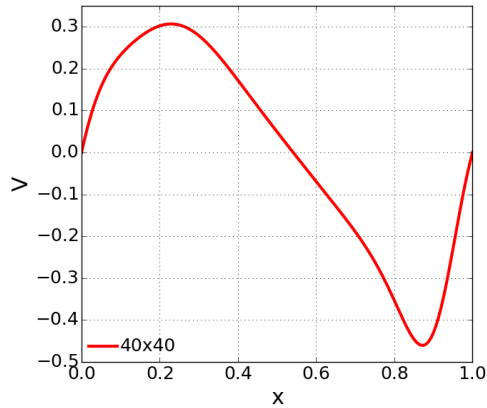
- $N_x N_y = 160 \times 160$



<Streamlines of 160x160 case runs>



<Centerline u-velocity>



<Centerline v-velocity>

- **Observation**

- As compared to previous condition of  $Re=100$ , the higher Reynolds number lid moving speed makes more strong vorticity on the bottom corner.
- The centered vortex relocates its position further down and so it moves toward the center of domain.

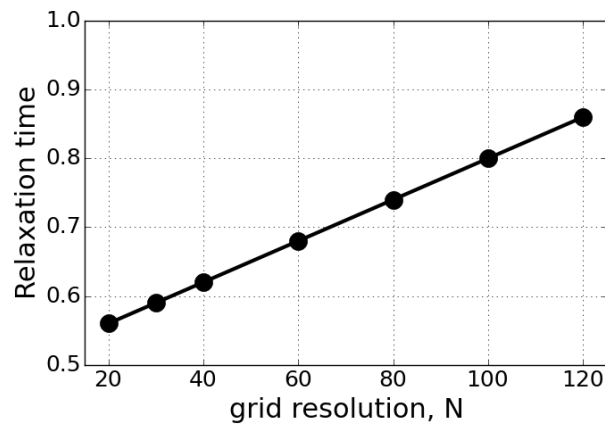
## 2.3 Problem1 - c

Examine the effect of the relaxation time  $\tau$  and the Mach number on method stability using your simulations, and compare your results to the stability criteria. Discuss how these parameters relate to the required grid size and physical time step size.

Given equation as stated in the previous section, the relaxation time is strongly coupled with kinematic viscosity  $\nu$ .

$$\nu = \frac{2\tau - 1}{6} \Delta x$$

In this simulation, grid spacing and time step are consistently set to unity for various grid resolution conditions in order to avoid complexity that may arise with non-unity  $\Delta t$  and  $\Delta x$ . Therefore, the grid spacing doesn't change the numerics of  $\Delta$ . Instead, to be consistent with  $Re=100$ , kinematic viscosity  $\nu$  is to be changed, resulting in change of  $\tau$ . The following figure illustrates the variable relaxation time allows the stable solution with different grid spacing.





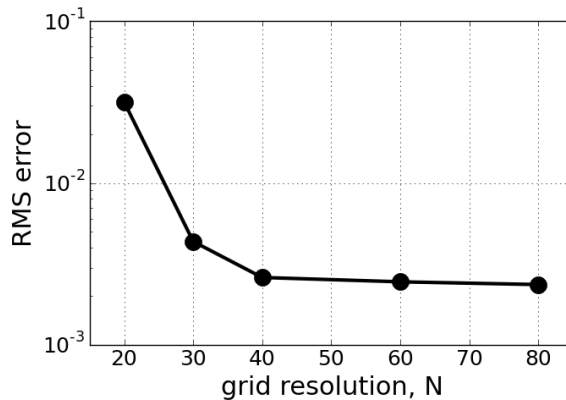
## 2.4 Problem1 - d

Compare your results with different grid resolutions to evaluate the numerical error and the order of the scheme.

The qualitative comparison between different grid spacing for  $Re=100$  case has been made in the previous section. As observed, the less grid point is, the less accurate and the more unstable solution were achieved. Please see section b.

In order to evaluate the numerical accuracy, the numerically resolved LBM data is compared to the reference data in terms of u-velocity component along the centered axial location. Assuming the reference data to be target data that is achieved when the numerical data is accurate, the root mean square (RMS) values were evaluated from the different grid spacing setups.

- Effect of grid spacing on the numerical accuracy



Overall, the accurate solution can be achieved with finer grid spacing setup. It means again that the smaller grid size contributes to less numerical error. As stated in the previous section, smaller grid spacing is more favorable condition for LBM because original Boltzmann equation was derived in molecular scales of length and time and so it can be more reliable when it is employed in the small spaced lattice configuration.

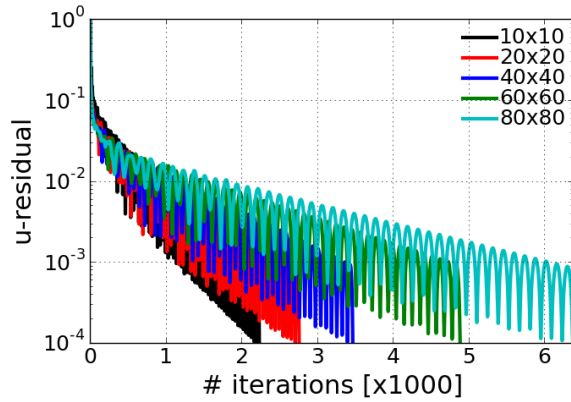
In addition, the RMS error seems to be stabilized as the grid resolution goes beyond 60 or more. Since the current Python script running for LBM is not such efficient, all the grid spacing analysis and fully resolved solution were discussed based on 60x60 resolution.

## 2.5 Problem1 - e

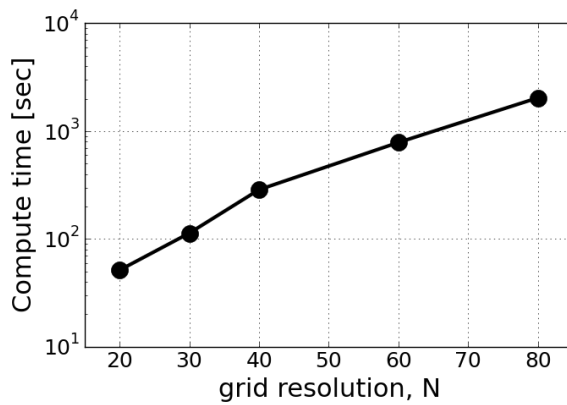
Discuss how the simulation speed depends on the grid resolution and time step.

The simulation for different grid resolutions were conducted with consistent convergence criterion as 0.01 % of normalized residual. The evolution of u-velocity residual with different grid spacing is placed on top of each other in the following plot. As the grid spacing become smaller, the more time integration was required to achieve the pre-specified convergence criterion. It also means the more grid points requires more computational time, which is illustrated as shown in the second figure.

- Effect of grid spacing on the convergence history



- Effect of grid spacing on the computational time to achieve the steady solution.



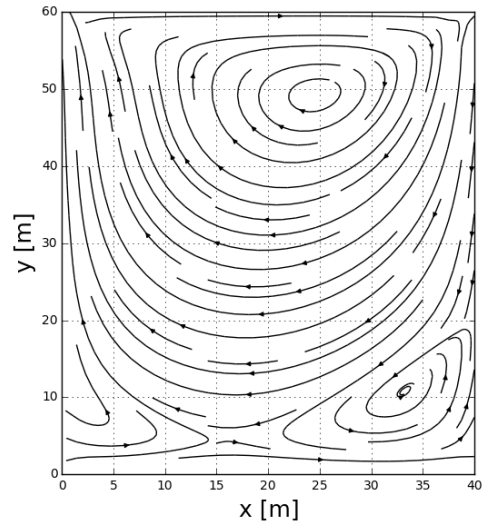
As observed in the previous homework problem, above observations are very typical in the fact that more iterations and more computational time are required for the higher grid resolutions. Specifically, in such a high resolution case, temporal updated quantity in distribution function is too small to rapidly approach to the steady solution, and it leads to more required iterations and again more computational time.

## 2.6 Problem1 - f

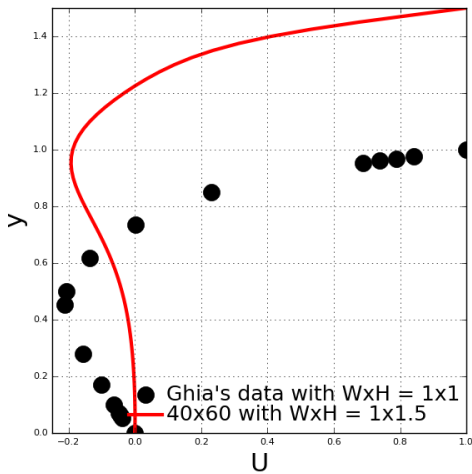
Repeat parts b and c for the case of  $H = 1.5W$  (except for validation).

### 2.6.1 Re = 100

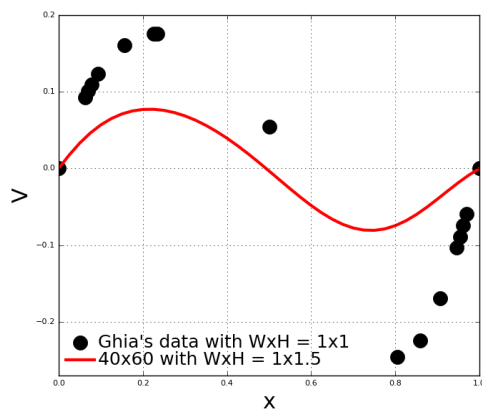
- $N \times N = 40 \times 60$



<Streamlines for 40x60 case>



<Centerline u-velocity compared with Ghia's numerically resolved data>



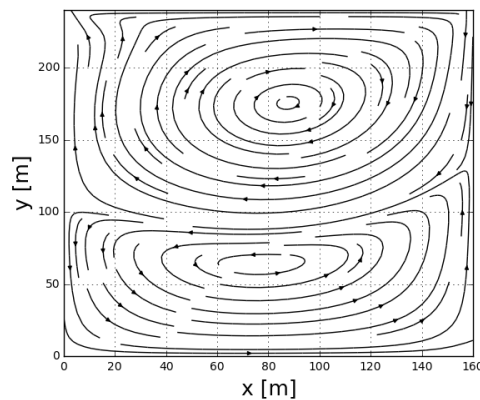
<Centerline v-velocity compared with Ghia's numerically resolved data>

- Observations

- In  $Re=100$ , the main vortex is formed in a very similar distance from the lid location.
- Because of fixed viscosity even in the long depth, the momentum transferred from the moving lid penetrates by the same amount, so it is observed that the vortex is created in a very same way as the previous domain configuration.
- This observation can be confirmed qualitatively in the centerline u-velocity plot: The curved shape is shifted upward.
- However, in the bottom clock-counterwise rotating vortices are observed on both bottom corners.

## 2.6.2 $Re = 500$

- $N \times N = 160 \times 240$



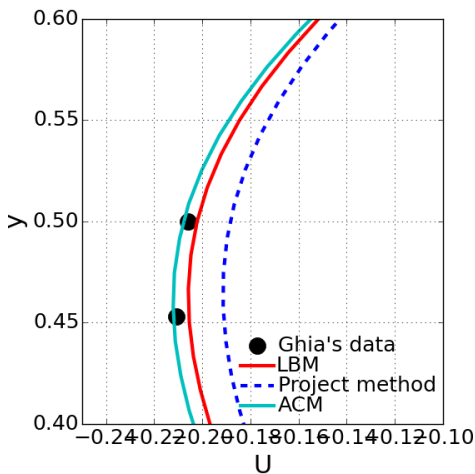
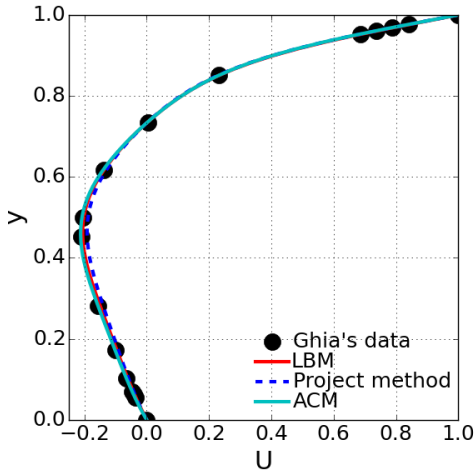
<Streamlines for 160x240 case under higher Resolution number condition>

- **Observations**
  - Higher resolution condition makes numerical solution unstable: The required grid spacing needs to be small enough to have stable solution. This case running took a couple of hours with current Python script.
  - There is distinctive layer separation in between two counter-rotating vortices: This was identically observed in the previous homework problems under the same condition.

## 2.7 Problem1 - g

Compare the results obtained in homework 3,4, and 5. Discuss the advantages and disadvantages of the methods.

In this test,  $Re=100$  case has been chosen for the desired comparison because this case doesn't not require impractically higher grid resolution such that the test can be performed very efficiently. The selected grid resolution is  $60 \times 60$  and three different solution methods were tested under the same condition. All the simulation for these solution methods were conducted with the same convergence rate of 0.01% for u-velocity residual.



<Comparison on the calculated u-velocity along the centerline with different solution method: bottom image is the magnified one for noticing the discrepancy>

#### • Observations

- Overallly all the solution methods were successfully running to achieve the reliable flow behaviors.
- However, there are noticeable discrepancy when there are compared to each other: Projection method is the worst at accuracy.
- It seems that more iterations may be needed to have accurate solution for Projection method.
- ACM gives the most accurate solution. However, in order to assess the solution method's accuracy, further investigation needs to be done for reliable assessment.

	LBM	Projection method	ACM
RMS of u-velocity	0.00413982355618	0.00857537642521	0.00132103968968
Iterations for convergence	4889	3496	37017
Compute time for convergence	784.86	4145.3	1711.31

Above table describes the computational performance among different solution methods. Here the projection method again turns out to be worst case in terms of computational time. This is because the projection method is to solve Laplacian equation for pressure correction terms which is the most expensive part in the whole process. Given grid resolution, the projection method requires more than 150 iterations for SOR method in the pressure correction step.

On the otherhand, LBM was the best way to have accurate solution with less computational effort. However, this comparison may not be reliable because the computational time is also very strongly sensitive to how the efficiently and effectively simulation code is made.