Immersed Boundary Lattice-Boltzmann Method for Complex geometries

Arpit Tiwari, R. Samala and S. P. Vanka
University of Illinois at Urbana-Champaign

Lattice Boltzmann Method

- LBM: Introduction
  - Statistical Mesoscale Method
  - Intermediate approach between Macroscopic and Microscopic Simulation

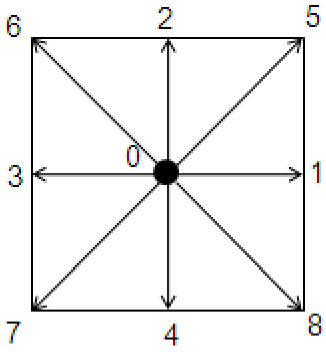
- From Molecular Dynamics
  - Simplified Kinetic Model with fewer details is constructed from the Molecular Boltzmann Equation
  - Fictitious particles moving in discrete space and time with discrete velocities
  - Velocity space is finite and consists of a particular set of values

- To Macroscopic Dynamics
  - Kinetic model is developed such that the Macroscopic Navier-Stokes equation is recovered in low Mach number limit
  - Linear convective operator
  - No pressure-velocity decoupling
  - Simplified boundary conditions
  - Localized non-linearities
  - Ease in Parallelization

LBM – Steps: Standard LBGK - D2Q9 model

1. Set the direction vectors

$$\mathbf{v}_i = \begin{cases} (0,0), \quad i=0, \\ \left(\cos\left((i-1)\frac{\pi}{2}\right), \sin\left((i-1)\frac{\pi}{2}\right)\right), \quad i=1,2,3,4, \\ \left(\cos\left((i-5)\frac{\pi}{2}+\frac{\pi}{4}\right), \sin\left((i-5)\frac{\pi}{2}+\frac{\pi}{4}\right)\right)\sqrt{2}, \quad i=5,6,7,8, \end{cases}$$



2. Find equilibrium distribution

$$f_i^{eq} = w_i \rho \left(1 + \frac{\mathbf{v}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{v}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2}\right),$$

3. Stream and Collide

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \Omega_i, \quad i = 1, 2, \ldots m, \qquad \Omega_i = \frac{f_i - f_i^{eq}}{\tau},$$

LBM – Steps: Standard LBGK - D2Q9 model

4. Density and velocity given by

$$\rho = \sum_{i=0}^{8} f_i, \quad \rho \mathbf{u} = \sum_{i=1}^{8} f_i \mathbf{v}_i.$$

5. Applying Chapman-Enskog procedure in low Mach number limit gives N-S equations

$$p = \rho c_s^2, \quad \nu = \left(\tau - \frac{1}{2} \right) c_s^2 \Delta t.$$

6. Boundary conditions: Discussed later

LBM: Advantages

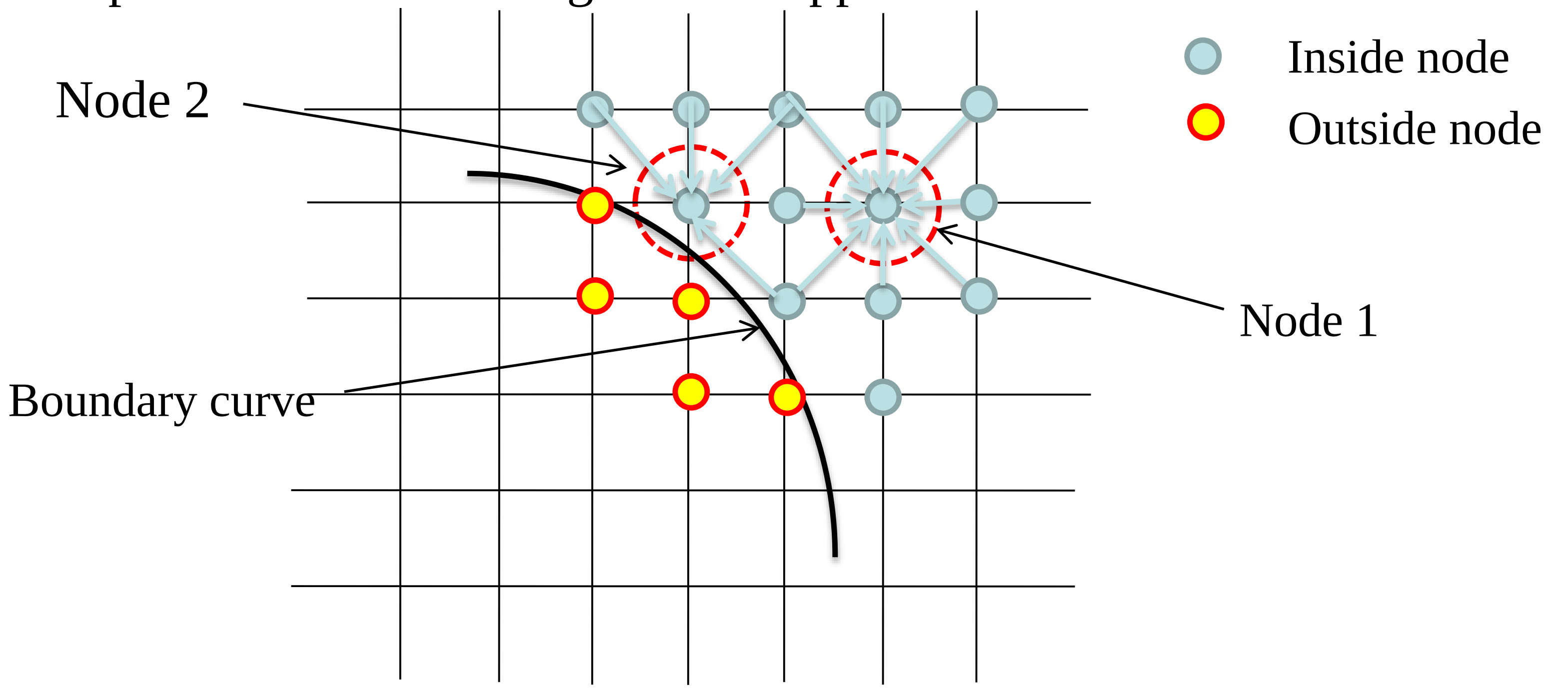
- Linear streaming operator: Conventional CFD suffers from the linearization issues of the convection operator
- Collision non-linearities are localized: Highly desirable for parallelization
- Ease in handling complex geometries: No slip is incorporated by a simplified and easy to implement condition which is the bounce back boundary condition
- Multiphase-flow analysis: LBM, being a simplification of the Boltzmann equation offers relative ease in incorporation of micro-scale physics

Boundary issues in LBM

- Conventional bounce-back condition is only first order accurate
- Bounce-back condition can not be applied to moving boundaries and pressure conditions
- Higher order accurate BC has always been a topic of research in LBM
- Need a boundary condition that gives more accurate results while preserving the simplicity of this method

Boundary issues in LBM

- LBM is based upon streaming and collision. So before streaming, each inside node must have 8 particle distribution functions pointing towards it
- As an example, in the figure below, node 1 has all the surrounding fluid nodes, but node 2 has only 5 surrounding interior nodes
- So, we need to somehow get the remaining components of node 2 by using the appropriate boundary conditions
- The novel method that we have proposed obtains the remaining components based on the ghost cell approach



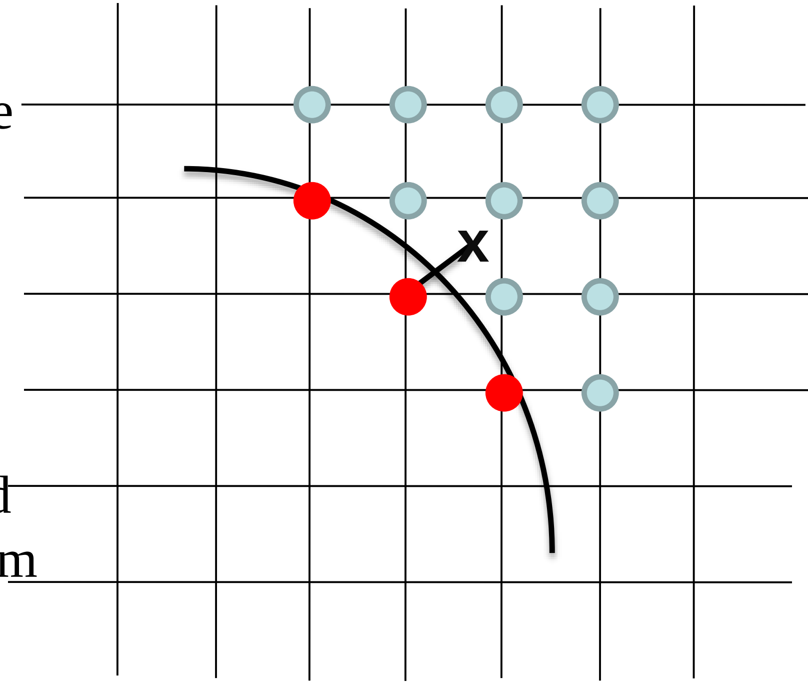
Ghost Cell Based IBM: Overview

- All the outside nodes that are required by the inside nodes are identified as ghost nodes
- Then particle distribution functions of the ghost nodes are estimated as:

$$f_{ghost,i} = f_{ghost,i}^{eq} + f_{ghost,i}^{neq}$$

- To obtain the equilibrium distribution function, we first need to estimate velocity and density values at the ghost nodes
- To do this:

1. Ghost node is projected normally inside the fluid domain



○	Inside node
●	Ghost node
x	Projected ghost node

2. Velocity and density values at projected node are computed by interpolating from the surrounding fluid modes

Ghost Cell Based IBM: Overview

- Then velocities of the ghost nodes are obtained based on the boundary condition as:

$$u_{boundary} = \frac{u_{ghost} + u_{projected}}{2} \Rightarrow u_{ghost} = 2u_{boundary} - u_{projected}$$

- Density of the ghost node is taken to be same as that of the projected node (zero normal gradient condition)
- Upon the calculation of density and velocity values of the ghost nodes, their equilibrium distribution functions can be easily computed using

$$f_i^{eq} = w_i \rho \left(1 + \frac{\mathbf{v}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{v}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2} \right),$$

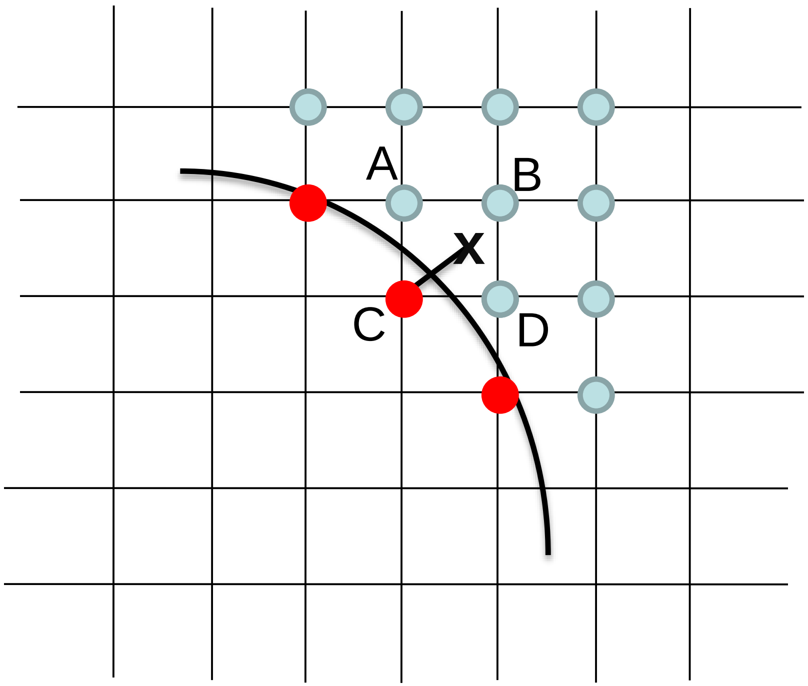
- Non-equilibrium part is calculated by the procedure similar to that adopted for the computation of density

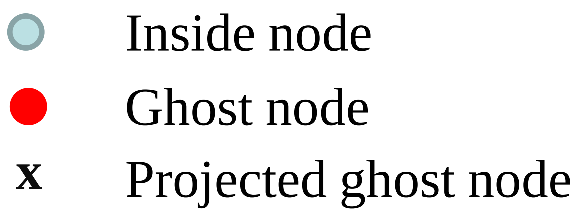
Bilinear Interpolation Details

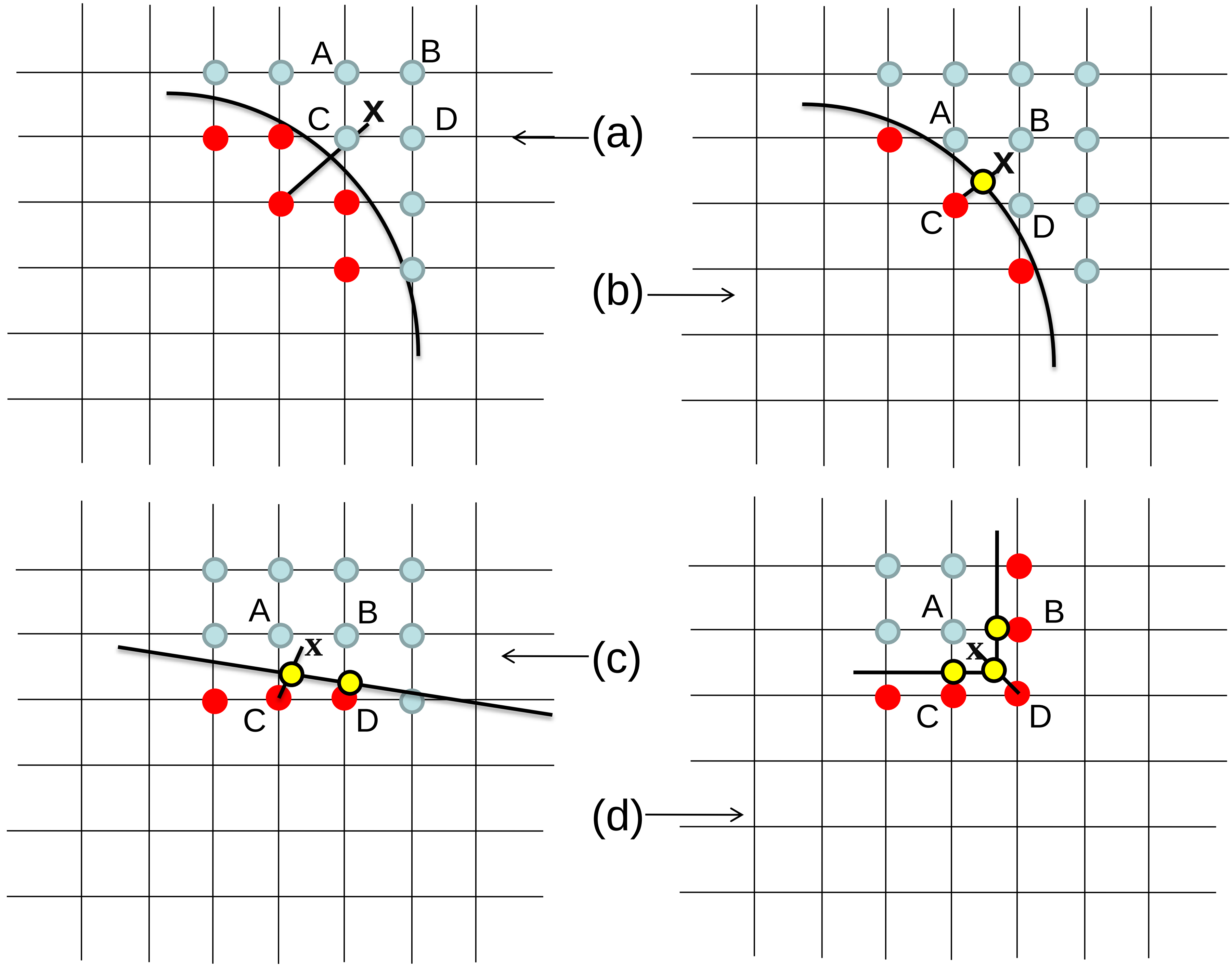
Depending upon where the projected ghost point lie, various cases may arise. We construct following cases by observing the nodes lying on the vertices of the square inside which the projected point is located:

Steps:

1. Find the square ABCD inside with the projected ghost node lie







A
B
C
x
D
(a)
(b)
(c)
(d)

Interpolation Matrix: Velocity

Using: $$\phi = ax + by + cxy + d$$

We have :

$$\begin{bmatrix} x_1 & y_1 & x_1 y_1 & 1 \\ x_2 & y_2 & x_2 y_2 & 1 \\ x_3 & y_3 & x_3 y_3 & 1 \\ x_4 & y_4 & x_4 y_4 & 1 \end{bmatrix} \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} = \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{Bmatrix}$$

Interpolation Matrix: Velocity

$$\phi_0 = \{x_0 \quad y_0 \quad x_0 y_0 \quad 1\} \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix}$$

$$\phi_0 = \{x_0 \quad y_0 \quad x_0 y_0 \quad 1\} \begin{bmatrix} x_1 & y_1 & x_1 y_1 & 1 \\ x_2 & y_2 & x_2 y_2 & 1 \\ x_3 & y_3 & x_3 y_3 & 1 \\ x_4 & y_4 & x_4 y_4 & 1 \end{bmatrix}^{-1} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{Bmatrix}$$

Interpolation Matrix: Density

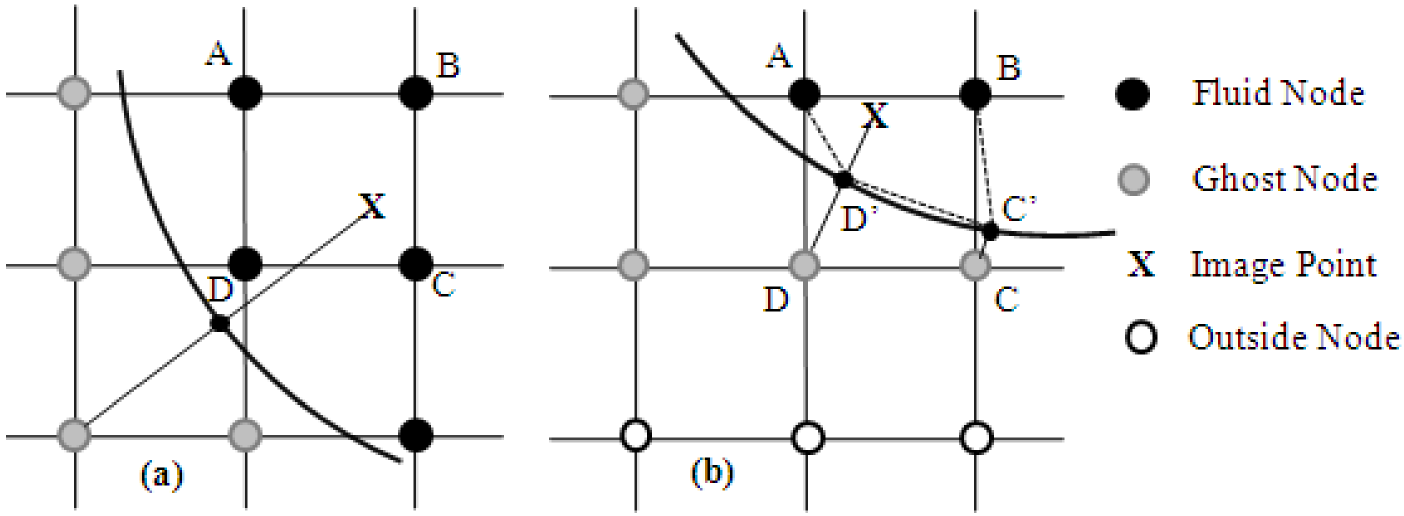
Using: $\phi = ax + by + cxy + d$ at nodal points

& $$\frac{\partial \phi}{\partial n} = an_x + bn_y + c(xn_y + yn_x) = 0$$ at wall points

We construct matrix, which (as an example)for case (c) looks like :

$$\begin{bmatrix} x_1 & y_1 & x_1 y_1 & 1 \\ x_2 & y_2 & x_2 y_2 & 1 \\ n_{x3} & n_{y3} & x_3 n_{y3} + y_3 n_{x3} & 0 \\ n_{x4} & n_{y4} & x_4 n_{y4} + y_4 n_{x4} & 0 \end{bmatrix} \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} = \begin{Bmatrix} \phi_1 \\ \phi_2 \\ 0 \\ 0 \end{Bmatrix}$$

Ghost Fluid IBM: Summary



Conditions: $\phi = ax + by + cxy + d, \quad \frac{\partial \phi}{\partial n} = an_x + bn_y + c\left(xn_y + yn_x\right) = 0$

Velocity: $a\left(\alpha_i x_i + (1-\alpha_i)x_i^{'}\right) + b\left(\alpha_i y_i + (1-\alpha_i)y_i^{'}\right) + c\left(\alpha_i x_i y_i + (1-\alpha_i)x_i^{'} y_i^{'}\right) + d$

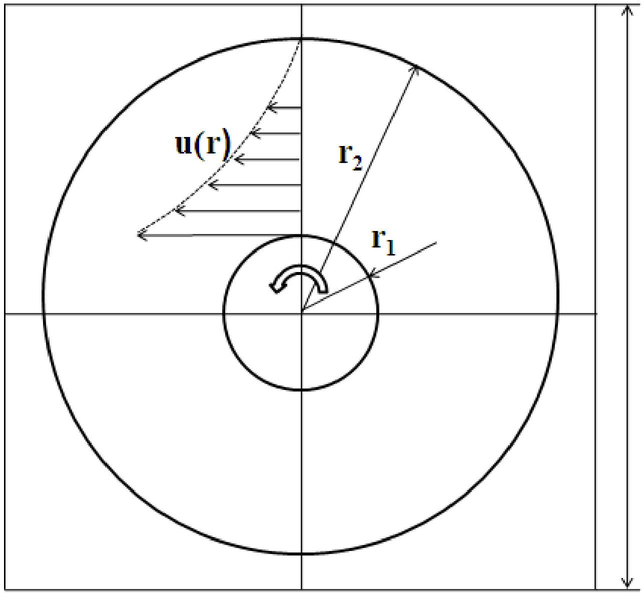
$= \alpha_i \mathbf{u}_i + (1-\alpha_i)\mathbf{u}_i^{'}, \quad i = 1, 2, 3, 4,$

Density: $a\left(\alpha_i x_i + (1-\alpha_i)n_{xi}\right) + b\left(\alpha_i y_i + (1-\alpha_i)n_{yi}\right) + c\left(\alpha_i x_i y_i + (1-\alpha_i)\left(x_i n_{yi} + y_i n_{xi}\right)\right)$

$+\alpha_i d = \alpha_i \rho_i, \quad i = 1, 2, 3, 4.$

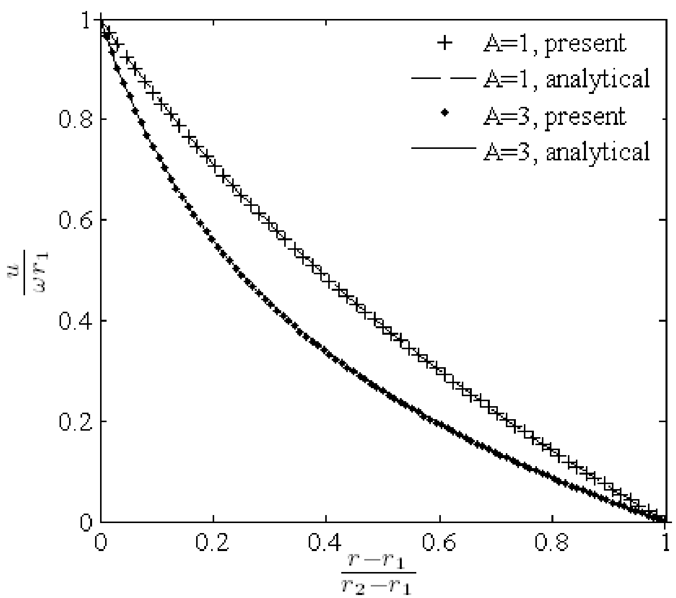
Extrapolation: $\mathbf{u_g} = 2\mathbf{u}_w - \mathbf{u}_{img}; \quad \rho_g = \rho_{img}$

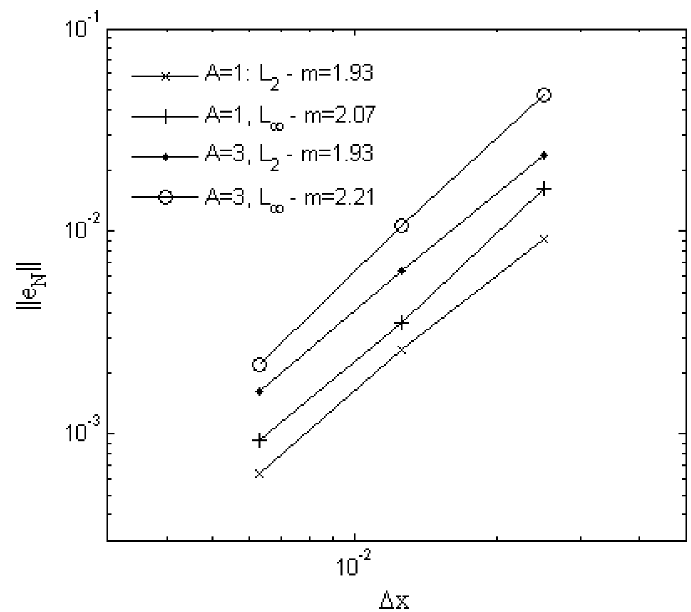
Results: Cylindrical Couette Flow: 321x321



$$\frac{u(r)}{\omega r_1} = \frac{r_1 r_2}{\left(r_2^2 - r_1^2\right)}\left(\frac{r_2}{r} - \frac{r}{r_2}\right).$$

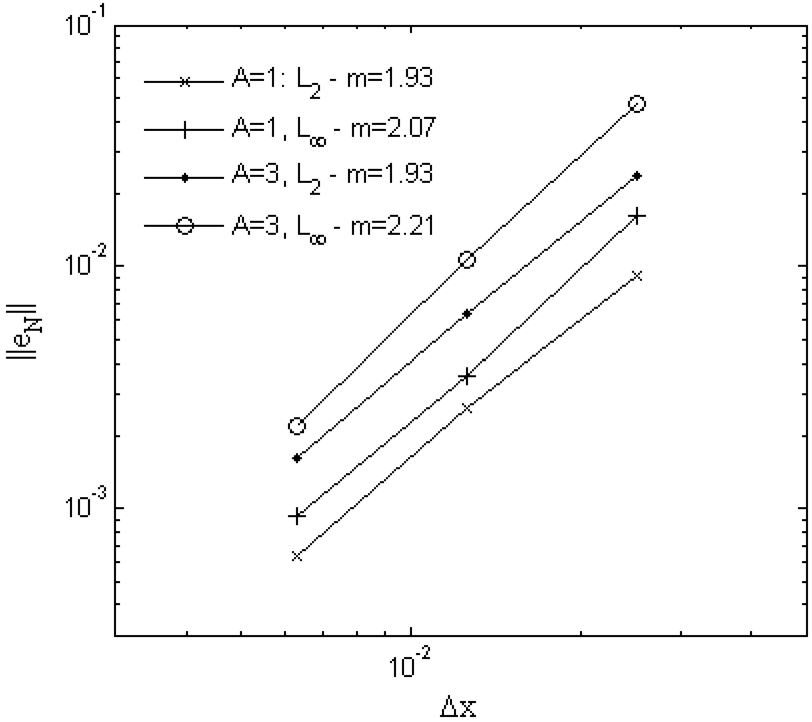
$$\left\|e_N\right\|_2 = \frac{1}{\omega r_1}\sqrt{\frac{\sum_{i=1}^{n}\left(u_i^N - u_i^e\right)^2}{n}}, \quad \left\|e_N\right\|_\infty = \max_{i=1,n}\frac{\left|u_i^N - u_i^e\right|}{\omega r_1},$$



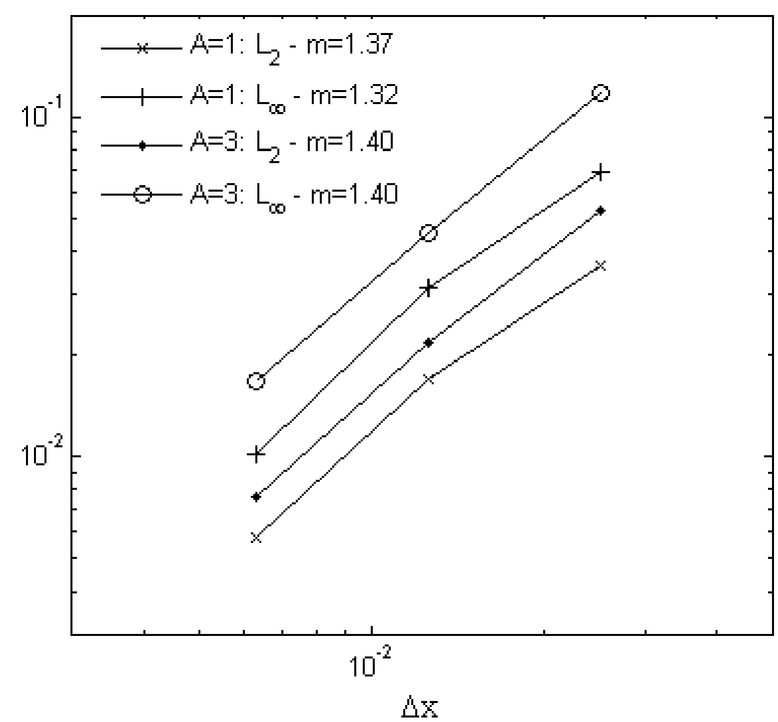


Results: Cylindrical Couette Flow

Second order accuracy of extrapolation of non-equilibrium part

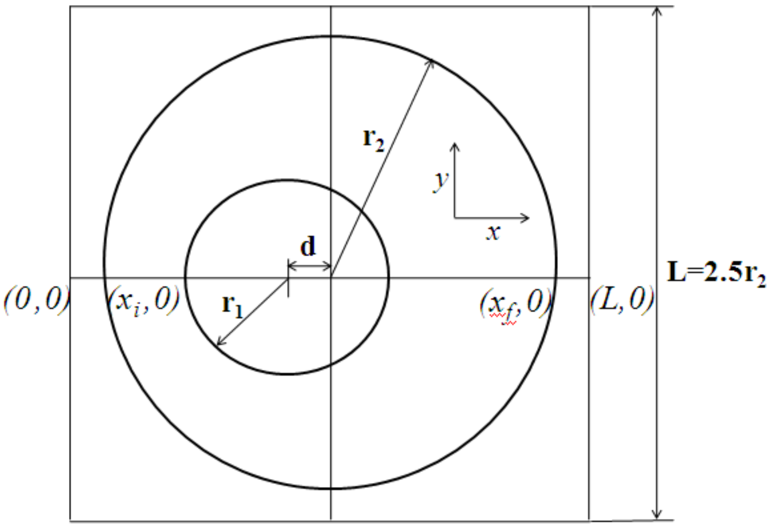


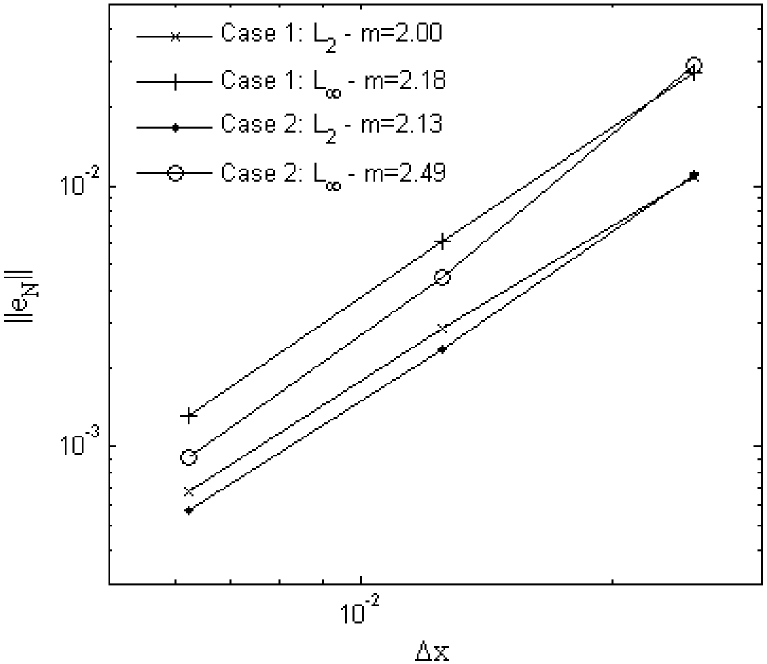
Simulation with extrapolation of non-equilibrium part



Simulation w/o extrapolation of non-equilibrium part

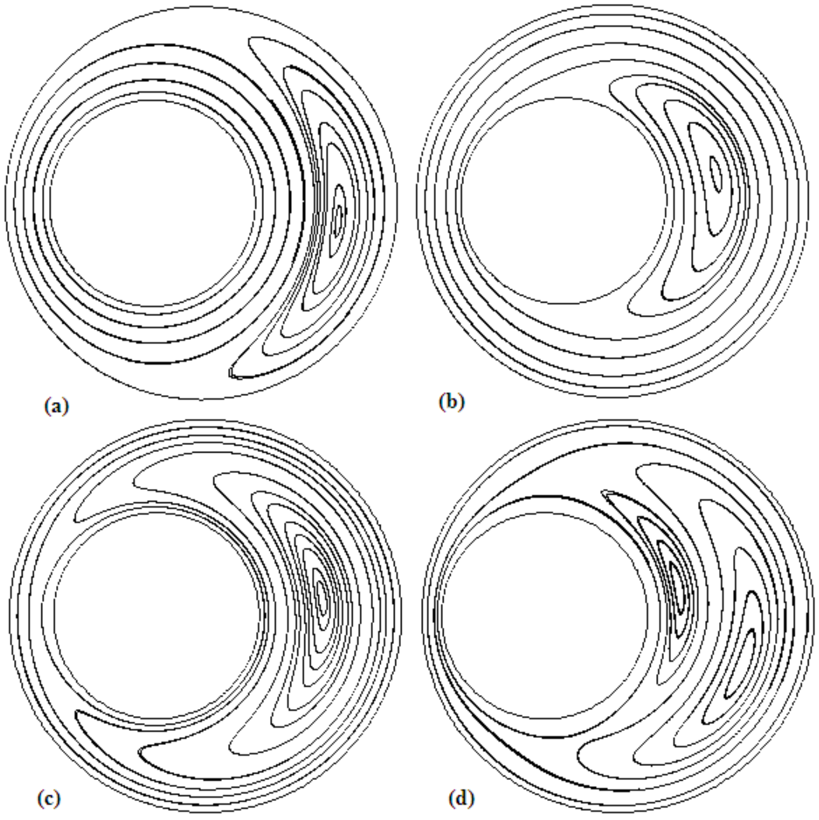
Results: Cylindrical Eccentric Flow



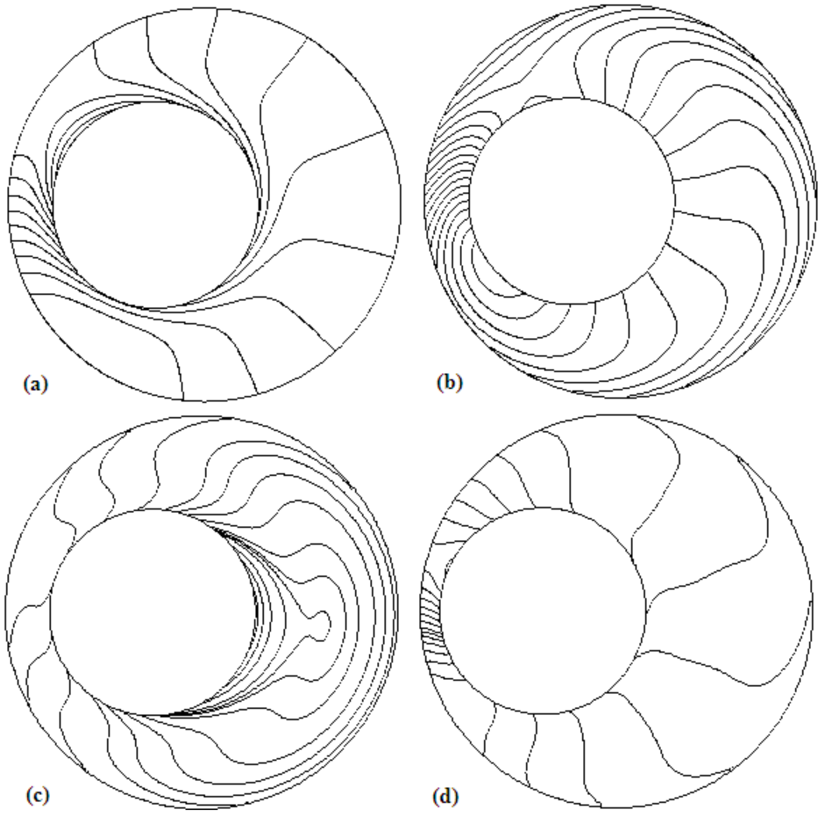


Cases	A	ε	$\omega_1 r_1$	$\omega_2 r_2$
1	2	0.5	-0.1/√3	0
2	2	0.5	0	0.1/√3
3	2	0.5	-0.1/√3	0.1/√3
4	2	0.725	-0.1/√3	-0.1/√3

Results: Cylindrical Eccentric Flow



Streamlines

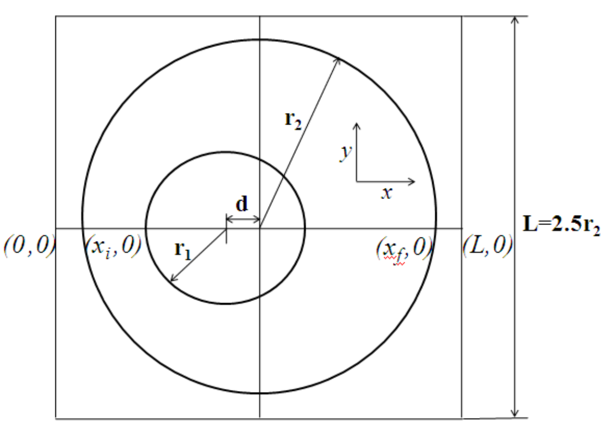


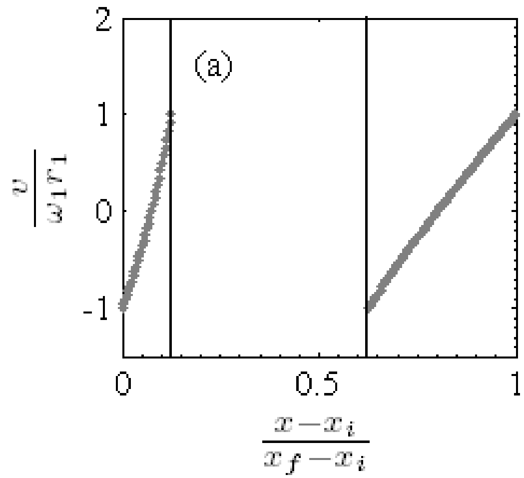
Pressure Contours

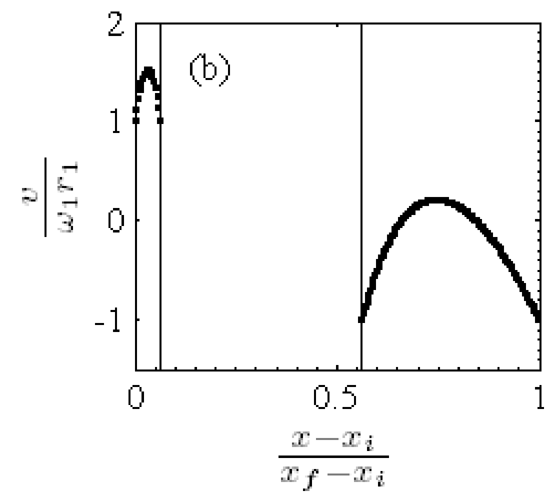
Results: Cylindrical Eccentric Flow: 321x321

$$F_x = \int_S \left(\rho v \frac{\partial u_t}{\partial n} n_y - P n_x \right) dS, \quad c_h = \frac{|2F_x|}{\bar{\rho} U^2 D}, \qquad F_y = -\int_S \left(\rho v \frac{\partial u_t}{\partial n} n_x + P n_y \right) dS, \quad c_v = \frac{|2F_y|}{\bar{\rho} U^2 D},$$

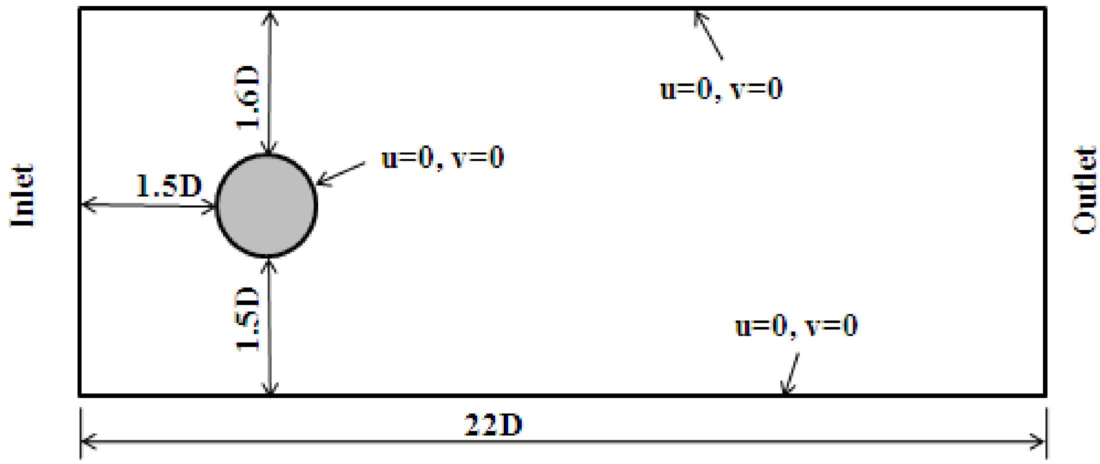
Simulation	(GF-IB-LBM)	(GF-IB-LBM)	FLUENT	FLUENT
Case 1	0.1348	0.5517	0.1372	0.5525
Case 2	0.1614	1.1065	0.1618	1.1114
Case 3	0.8941	0.5391	0.9087	0.5418
Case 4	0.8940 (0.9509)	2.8724 (2.8560)	1.0003	2.8701



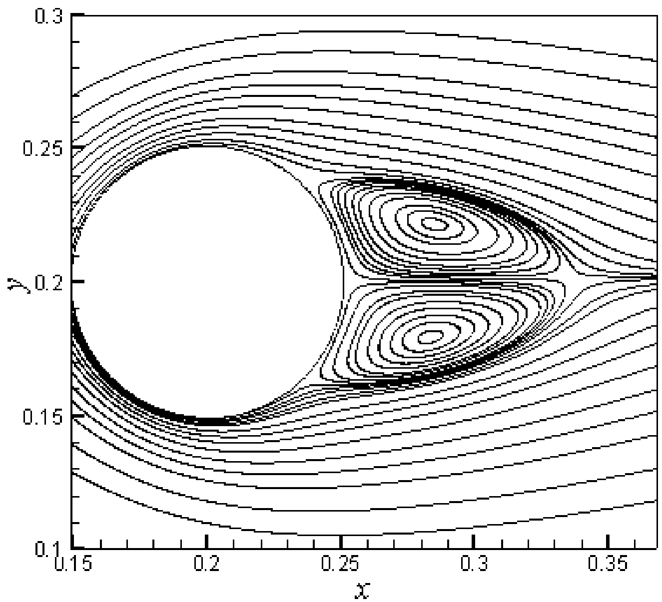


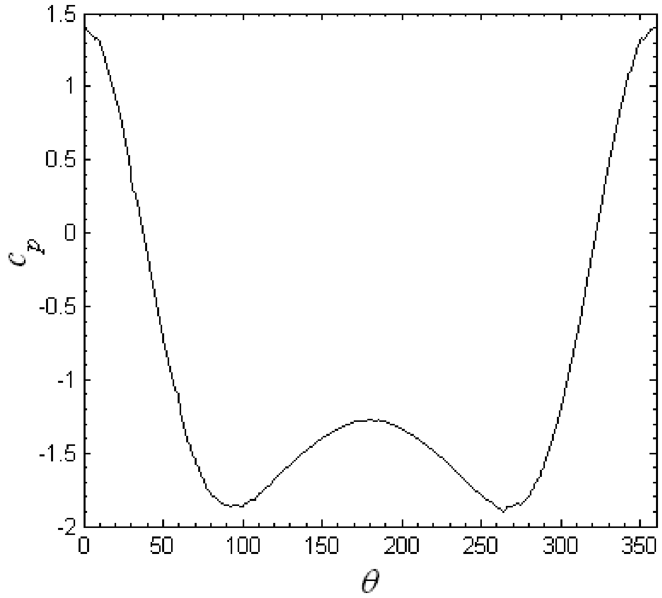


Results: Flow over cylinder in a channel: n=64



$$c_p = \frac{2\left(p - \bar{p}_{in}\right)}{\rho_\infty u_{max}^2},$$

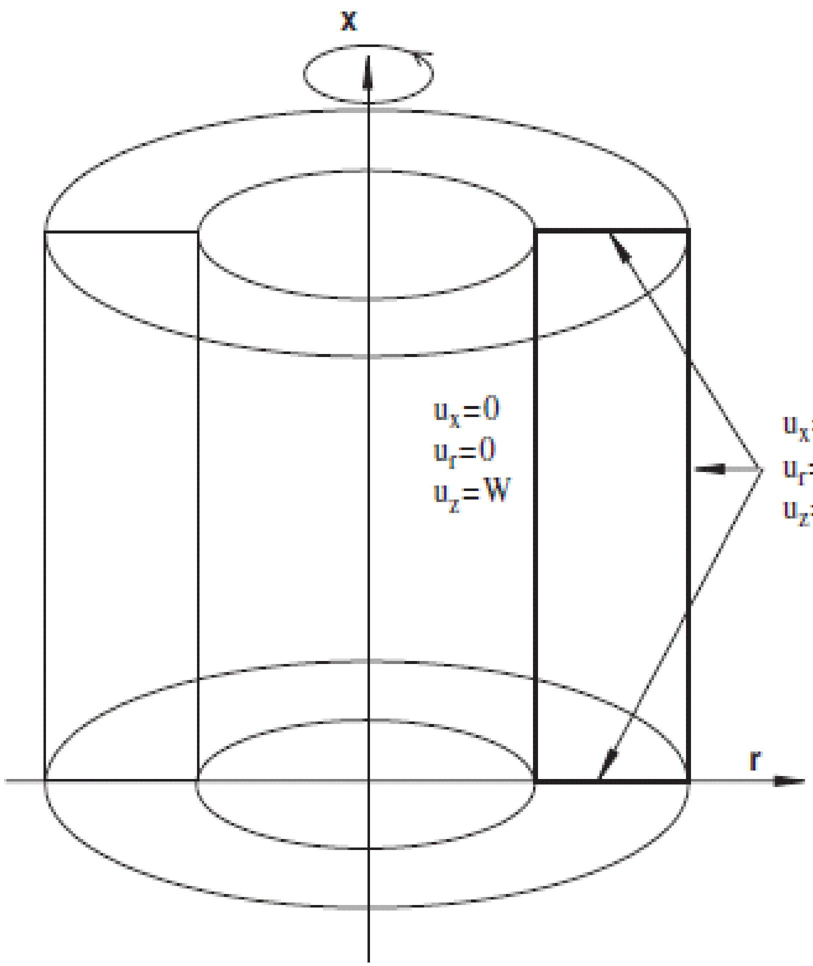




Results: Flow over cylinder in a channel

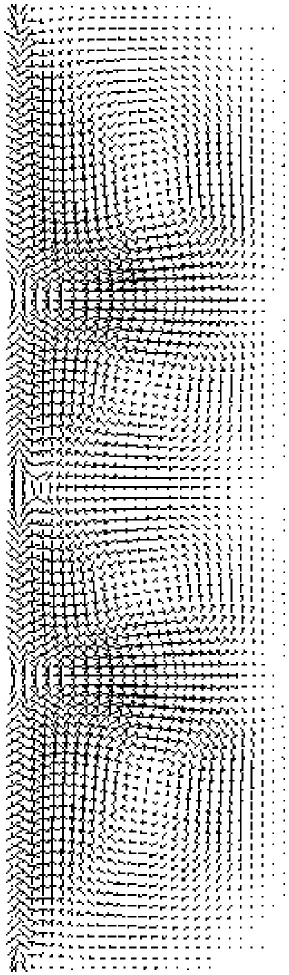
	n	16	32	64	Mussa et al. (n=80) [51]	Schäfer and Turek [50]
c_d	I	5.3203 (5.3449)	5.4772 (5.5025)	5.5799 (5.6057)		5.5700-5.5900
	II	5.3577 (5.3853)	5.515 (5.5434)	5.6182 (5.6471)	5.5584	
c_l	I	0.0488 (0.0490)	0.0141 (0.0142)	0.0101 (0.0101)		0.0104-0.0110
	II	0.0493 (0.0496)	0.0143 (0.0144)	0.0103 (0.0103)	0.0113	
$\Delta\bar{p}$	I	6.1878 (6.2164)	5.9088 (5.9361)	5.9498 (5.9773)		5.8600-5.8800
	II	6.2292 (6.2614)	5.9489 (5.9796)	5.9897 (6.0205)	5.8091	
$\bar{L}$	I	0.8436	0.8629	0.8676		0.8420-0.8520
	II	0.8342	0.853	0.8577	0.8402	

Results: 3D Taylor Couette Flow

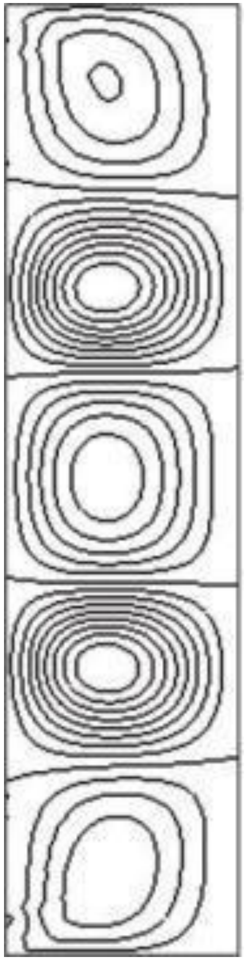


Re = 100
Re = WD/ν
W = Inner cylinder Velocity,
D = Gap Width
Aspect Ratio = 3.8
Grid Size = 125 X 125 X 95

Results: 3D Taylor Couette Flow



Velocity Vectors



U velocity contours

Shan and Chen (Single Component Multiphase)

$$f_\alpha\left(\mathbf{x}+e_\alpha\delta t, t+\delta t\right)= f_\alpha\left(\mathbf{x},t\right)-\frac{1}{\tau}\left[f_\alpha\left(\mathbf{x},t\right)-f^{eq}{}_\alpha\left(\mathbf{x},t\right)\right]$$

$$\alpha = 0,1,.....N$$

$$f_\alpha{}^{eq} = \rho w_\alpha\left[1+\frac{3}{c^2}\left(\mathbf{e}_\alpha\cdot\mathbf{U}\right)+\frac{9}{2c^4}\left(\mathbf{e}_\alpha\cdot\mathbf{U}\right)^2-\frac{3}{2c^2}\left(\mathbf{U}\cdot\mathbf{U}\right)\right]$$

$$\mathbf{F}_{\text{int}}\left(\mathbf{x}\right)=-\psi\left(\mathbf{x}\right)\sum_{x'}G\left(\mathbf{x},\mathbf{x}'\right)\psi\left(\mathbf{x}'\right)\left(\mathbf{x}'-\mathbf{x}\right)$$

$$G\left(\mathbf{x},\mathbf{x}'\right)=\begin{cases} g, & \left|\mathbf{x}-\mathbf{x}'\right|\le c, \\ 0, & \left|\mathbf{x}-\mathbf{x}'\right|> c, \end{cases}$$

Calculation of density, velocity and pressure

$$\rho = \sum_{\alpha=0}^{N} f_\alpha = \sum_{\alpha=0}^{N} f^{eq}{}_\alpha, \qquad \nu = \left(\tau - \frac{1}{2}\right) c^2{}_s \delta t$$

$$\rho(\mathbf{x})\mathbf{U} = \rho(\mathbf{x})\mathbf{u} + \frac{1}{2}\mathbf{F}_{\text{int}}$$

$$p = c^2{}_s \rho + \frac{c_0}{2} g \left[\psi(\rho)\right]^2$$

p is calculated using a equation of state

For D2Q9 and D3Q19, $c_s = 1/\sqrt{3}$

$$\psi(\rho) = \sqrt{\frac{2\left(p - c_s^2 \rho\right)}{c_0 g}}$$

He and Chen (Multicomponent single phase)

$$f_\alpha(x+e_\alpha\delta t, t+\delta t) - f_\alpha(x,t) = -\frac{f_\alpha(x,t) - f_\alpha^{eq}(x,t)}{\tau} - \frac{2\tau-1}{2\tau}\frac{(e_\alpha-u).\nabla\psi(\phi)}{c_s^2}\Gamma_\alpha(u)\delta t$$

$$g_\alpha(x+e_\alpha\delta t, t+\delta t) - g_\alpha(x,t) = -\frac{g_\alpha(x,t) - g_\alpha^{eq}(x,t)}{\tau} + \frac{2\tau-1}{2\tau}(\mathrm{e}_\alpha\text{-u}).[\Gamma_\alpha(\mathrm{u})(F_s+G) - (\Gamma_\alpha(\mathrm{u}) - \Gamma_\alpha(0))\nabla(p - c_s^2\rho)]\delta t$$

LBM Multiphase

Equilibrium values $$f_\alpha^{\text{eq}} = w_\alpha \phi \left[1 + \frac{3\mathbf{e}_\alpha \cdot \mathbf{u}}{c^2} + \frac{9(\mathbf{e}_\alpha \cdot \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}^2}{2c^2} \right]$$

$$g_\alpha^{\text{eq}} = w_\alpha \left[p + \rho \left(\frac{3\mathbf{e}_\alpha \cdot \mathbf{u}}{c^2} + \frac{9(\mathbf{e}_\alpha \cdot \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}^2}{2c^2} \right) \right]$$

$$\Gamma_\alpha(\mathbf{u}) = w_\alpha \left[1 + \frac{3\mathbf{e}_\alpha \cdot \mathbf{u}}{c^2} + \frac{9(\mathbf{e}_\alpha \cdot \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}^2}{2c^2} \right]$$

Equation of State: $$\psi(\phi) = c_s^2 \left[\frac{1 + \phi + \phi^2 - \phi^3}{(1-\phi)^3} - 1 \right] - a\phi^2,$$

LBM Multiphase

$$\phi = \sum_{\alpha} f_{\alpha}$$

$$\rho \mathbf{u} c_s^2 = \sum_{\alpha} g_{\alpha} e_{\alpha} + \frac{c_s^2}{2}\left(F_s + G\right)\delta t$$

$$g = f RT + \psi(\rho)\Gamma(0)$$

$$p = \sum_{\alpha} g_{\alpha} \quad - \frac{1}{2}\mathbf{u}.\nabla(p - c_s^2 \rho)\delta t$$

$$\rho(\phi) = \rho_l + \frac{\phi - \phi_l}{\phi_h - \phi_l}\left(\rho_h - \rho_l\right)$$

$$\nu(\phi) = \nu_l + \frac{\phi - \phi_l}{\phi_h - \phi_l}\left(\nu_h - \nu_l\right)$$

$$\nu = \left(\tau - \frac{1}{2}\right) RT\delta t$$

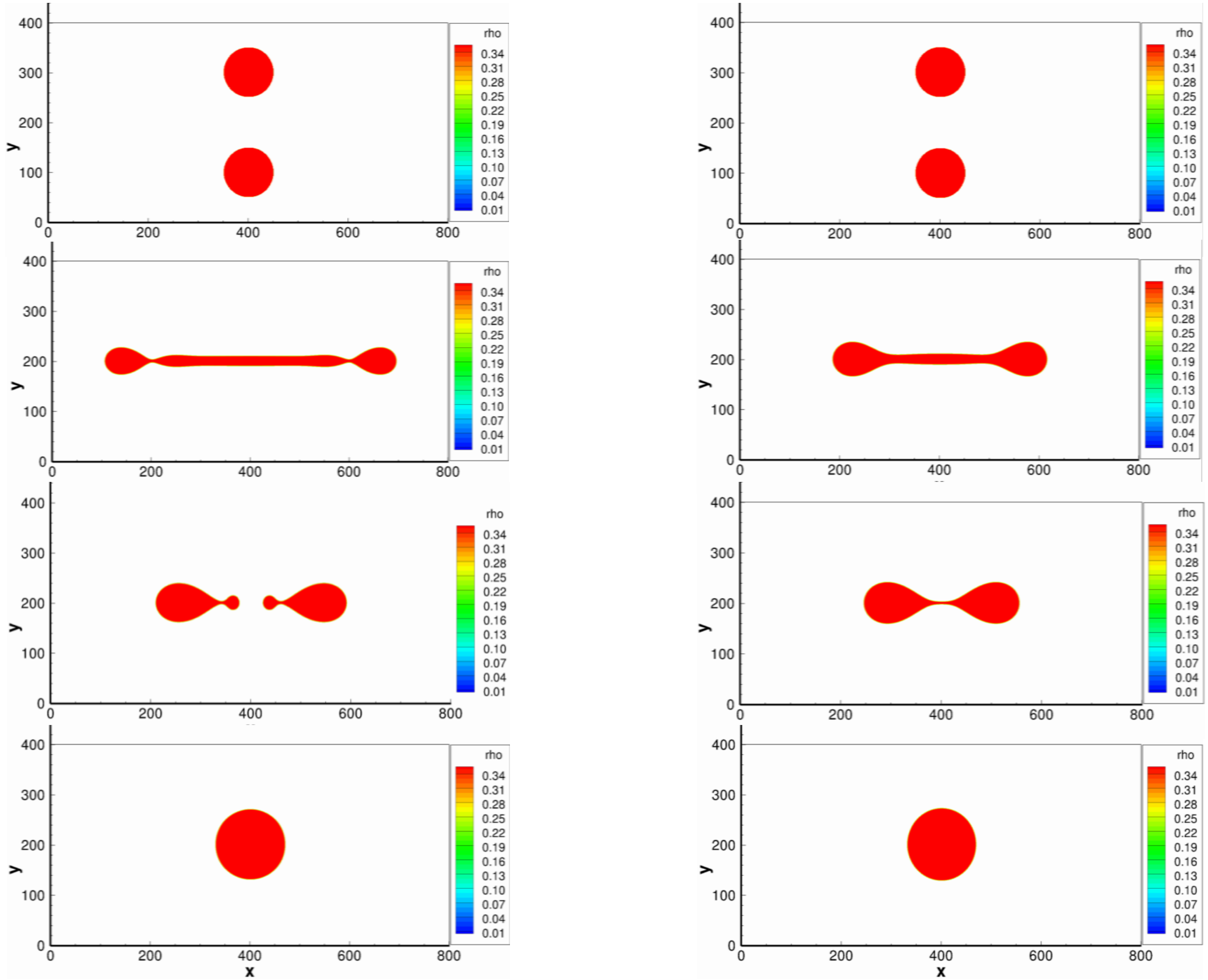
Droplet Collision / Impingement

- 2 droplets approach with identical velocity, collide and coalesce.
- Density ratio = 60, viscosity ratio = 100 (left) and 120(right)
- Droplet velocity = 0.05 (lattice units)

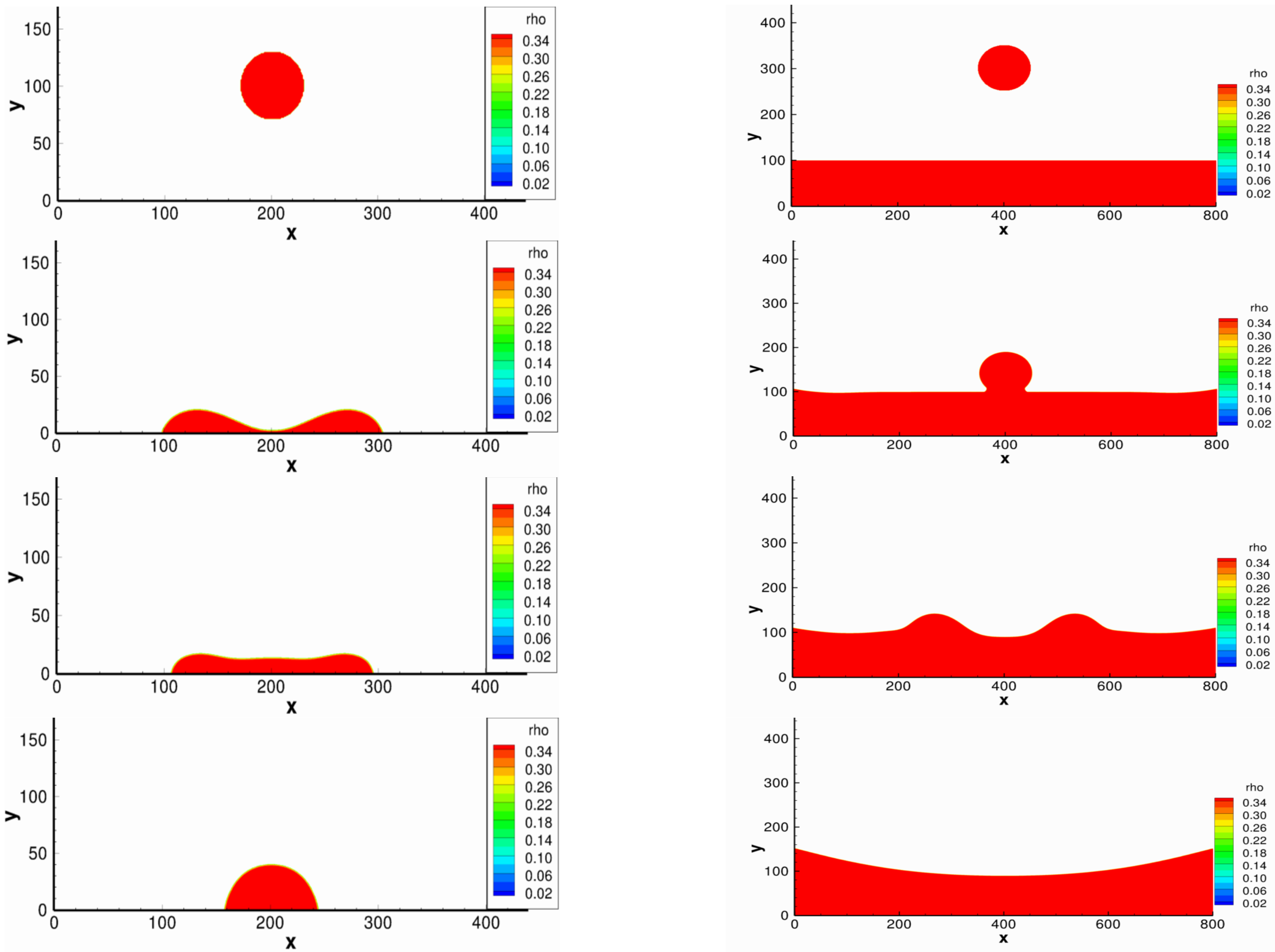
- Droplet impingement on a solid surface (left), and liquid pool (right)

- Density ratio = 60, viscosity ratio = 100,

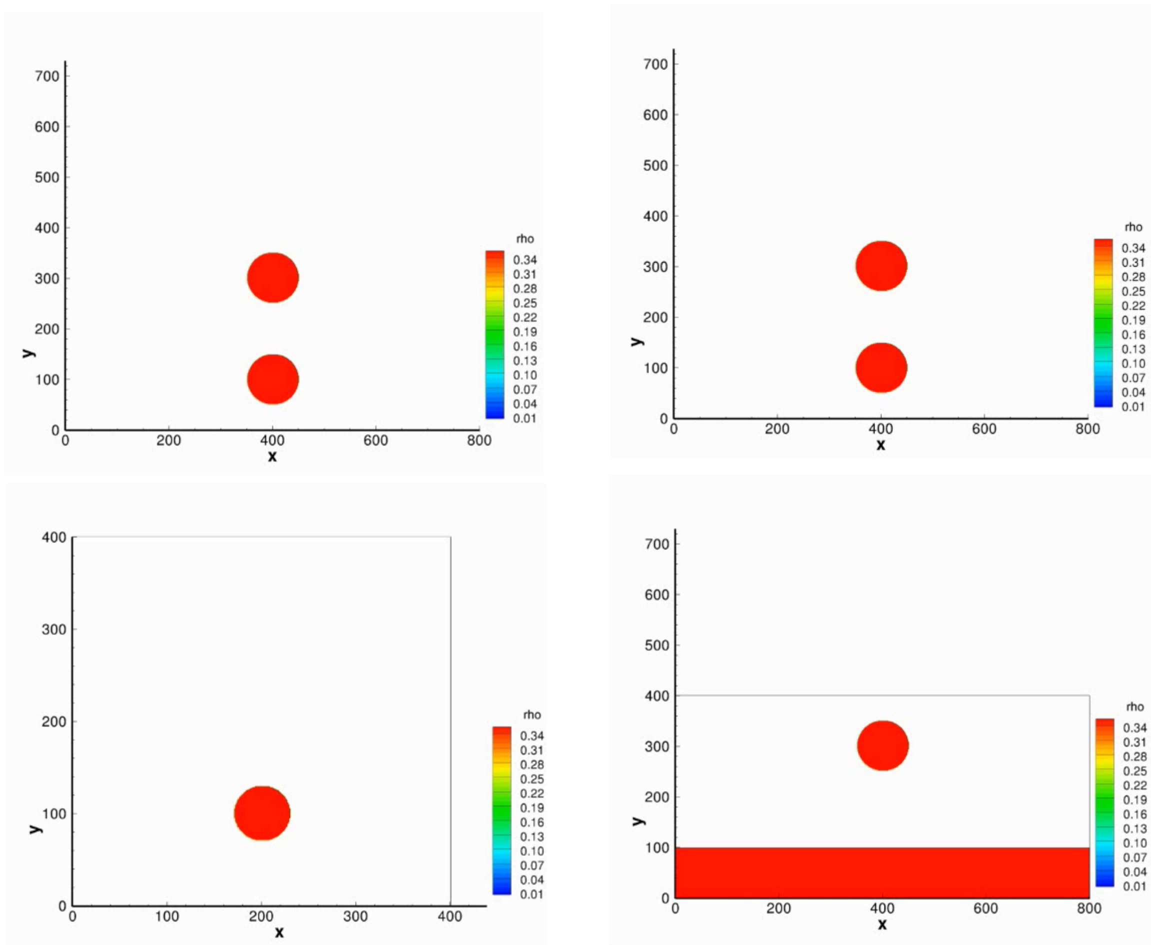
Droplet Collision



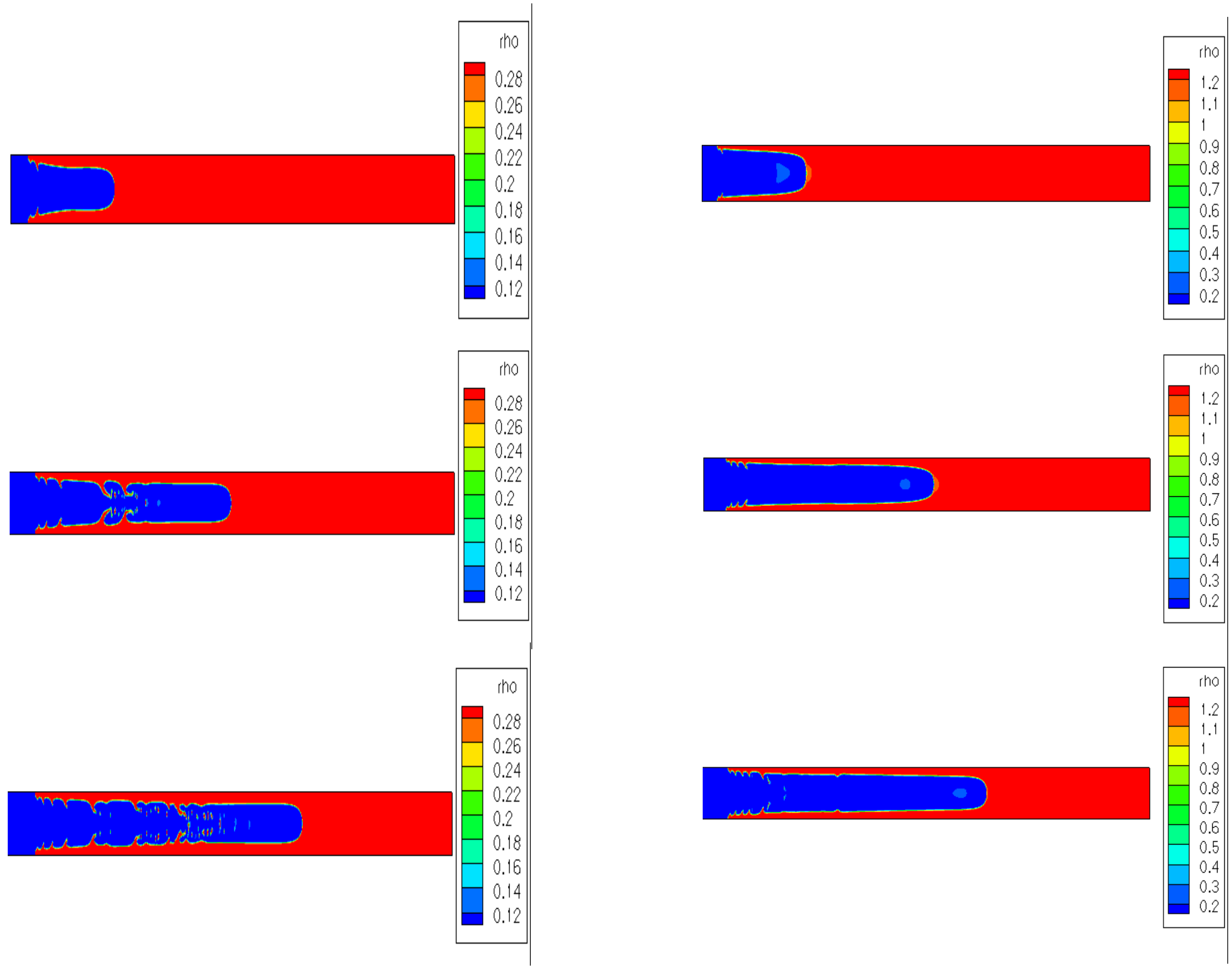
Droplet Impingement



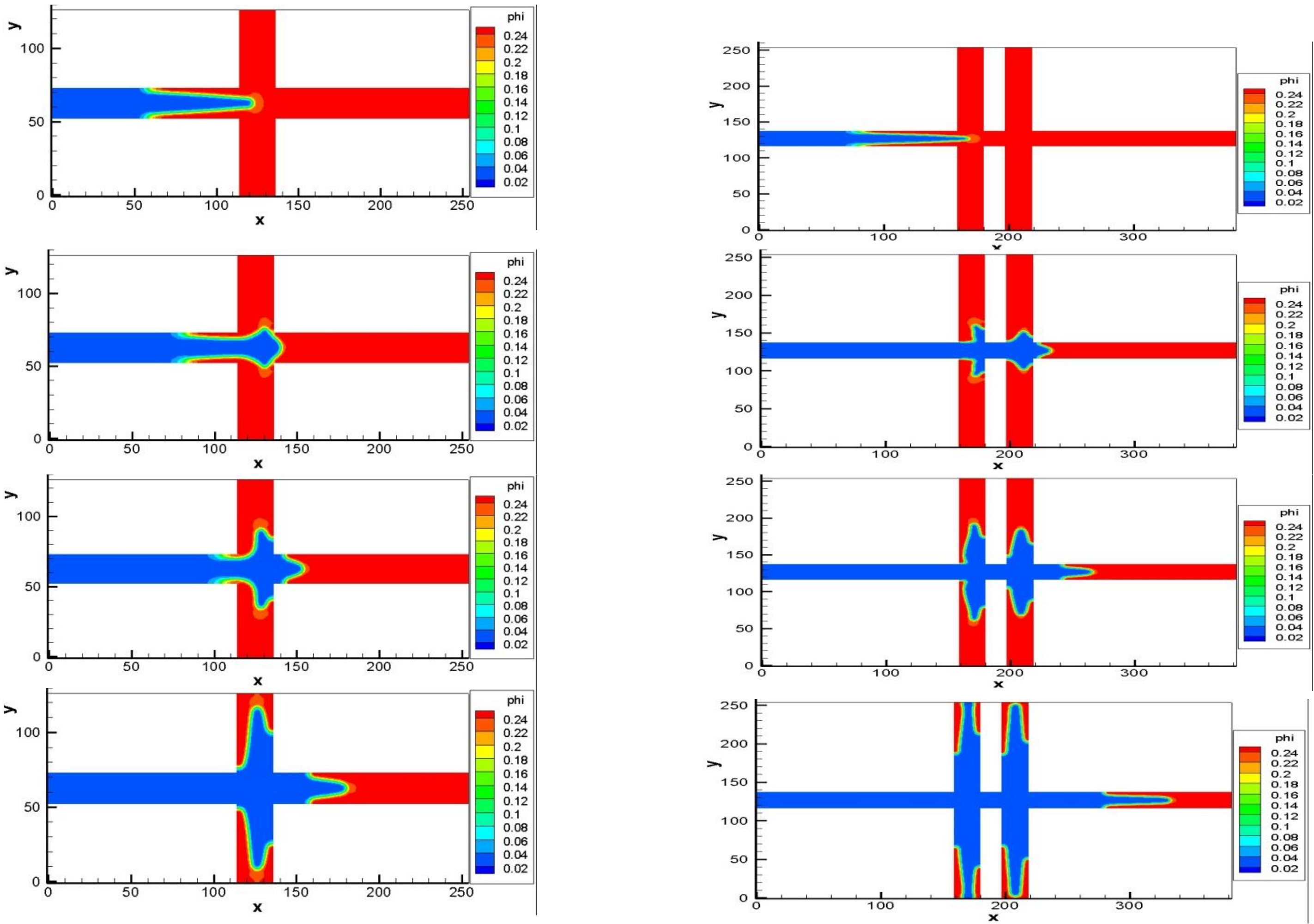
Droplet dynamics



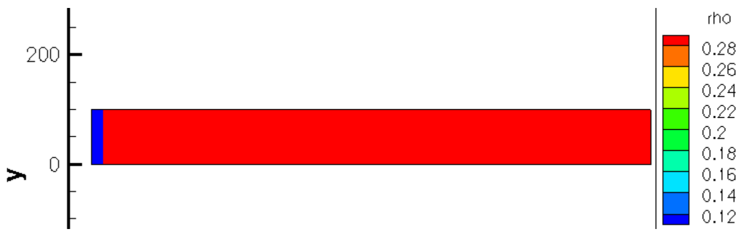
Displacement Flow

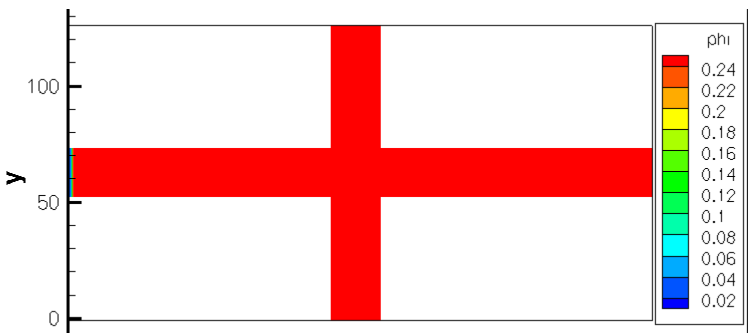


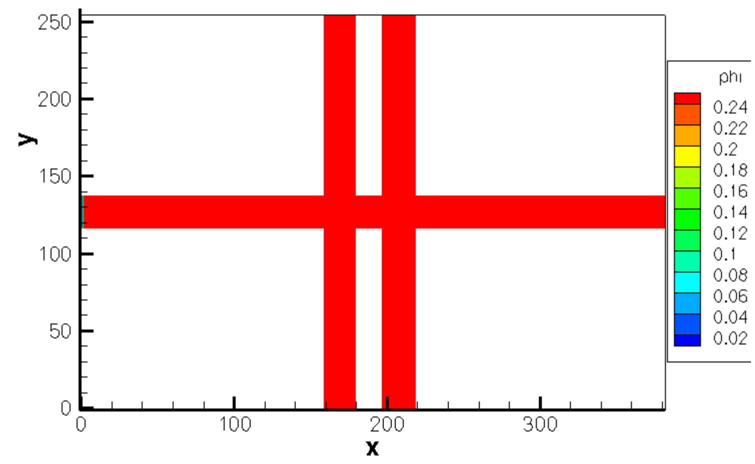
Flow in Complex Geometries



Displacement flow in channel and header







Works Cited

[1] X. He, S. Chen, and R. Zhang, ‘‘A lattice Boltzmann scheme for incompressible multiphase flow and its application in simulation of Rayleigh-Taylor instability,’’ J. Comput. Phys. **152,** 642–663 (1999).

[2] X. Shan and H. Chen, “Lattice Boltzmann model for simulating flows with multiple phases and components,”*Phys. Rev. E* **47**, 1815 (1993).

[3] P. L. Bhatnagar, E. P. Gross, and M. Krook, “A model for collision process in gases. I. Small amplitude processes in charged and neutral one-component system,” *Phys. Rev.* **94**, 511 (1954).

[4] X. He, R. Zhang, S. Chen, and G. D. Doolen, “On the three-dimensional Rayleigh–Taylor instability,” *Phys. Fluids E* **11**, 1143 (1999).

[5] M. A. Rahman, A.M. Jacobi, “Experimental study of wetting anisotropy and Condensate Drainage Enhancement on Microgrooved Aluminum surface,” IMECE2011-64247 (2011).

[6] M.A. Rahman, A.M. Jacobi, “Wetting Behavior and Drainage of Water Droplets on Microgrooved Brass Surface,” IMECE2011-64255 (2011).

[7] M.A. Rahman, A.M. Jacobi, “Condensation, Frost Formation and Condensate Retention Characteristics of
Microgrooved Surfaces under Natural Convection Condition,” *Manuscript.*

Works Cited

[8] "NVIDIA CUDA C Programming Guide," Version 3.2, NVIDIA.

[9] J. Chao, R. Mei, R. Singh and W. Shyy, "A filter-based, mass-conserving lattice Boltzmann method for immiscible multiphase flows," *Int. J. Numer. Meth. Fluids* **66**, 622 (2011).

[10] R. Zhang, X. He, S. Chen, "Interface and surface tension in incompressible lattice Boltzmann multiphase model," Computer Physics Communications **129**, 121 (2000)

[11] P.K. Jain, A. Tentner, Rizwan-uddin, "Lattice Boltzmann Simulation of Liquid Drop Coalescence Driven by Surface Tension," Transactions of the American Nuclear Society, **101**, 870 (2009)

[12] M. Cheng, J. Hua, J. Lou, "Simulation of bubble-bubble interaction using a lattice Boltzmann method" Computers and Fluids, **39**, **2**, 260 (2010)