

Chapter 1

Fluid flow modelling

1.1 Fluid simulation using Lattice Boltzmann Method (LBM)

The Navier-Stokes equation describes the motion of a non-turbulent Newtonian fluid. The equation is obtained by applying Newton's second law to the fluid motion, together with an assumption that the fluid stress is the sum of the viscous term, proportional to the gradient of velocity, and the pressure term. The Lattice Boltzmann equation (LBM) Method (?) is an alternative approach to the classical Navier-Stokes solvers for fluid flow and works on an equidistant grid of cells, called lattice cells, which interact only with their direct neighbours. The Lattice Boltzmann Method evolved out of the Lattice-Gas Cellular Automata (LGCA), a statistical model based on the kinetic theory of gases, which simulates gas as particles at discrete points in space (e.g., Cartesian or hexagonal mesh) represented by Boolean variables, which means that the mass of a particle is fixed to 1. The LGCA and the LBM are categories of *Cellular Automata*, which is characterized by: (a) lattice, (b) a definition of state variable at every point in the lattice, and (c) an update rule based on local and neighbouring information. The important aspect, however is the *discretization of the velocity*, which means that the particle velocities are restricted to a predefined set of orientations. The LB method has evolved as a comprehensive fluid solver and its theoretical aspects link well with the conventional central finite difference scheme (?). The systems simulated using cellular automata approaches exhibit fluid-like behaviour. The process of averaging the density and the momentum over some regions of the space, i.e. coarse graining, produces some useful fluid simulation. The LBM replaces the boolean variables of the LGCA (?) with discretized probability distribution functions, $f_i(\vec{x})$, thereby eliminating the need for ensemble averaging. A simplified collision operator described in ? is implemented in the LBM approach.

1.1.1 Formulation

The Lattice Boltzmann Method is a 'micro-particle' based numerical time-stepping procedure for the solution of incompressible fluid flows. Consider a 2D incompressible fluid flow with density ρ and kinematic viscosity ν , in a rectangular domain D . The fluid domain is divided into a rectangular grid or lattice, with the same spacing h in both the x - and the y -directions, as shown in Figure ???. These lattices are usually classified in the literature using the $D\alpha Q\beta$ -notation, where α is an integer number denoting the space dimensionality and β is the number of discrete velocities (but including the possibility of having particle at rest) within the momentum discretization. The most common lattices are the $D2Q9$ and the $D3Q19$ -models, see ?. The present study focuses on two-dimensional problems, hence the $D2Q9$ momentum discretization is adopted, where the fluid particles at each node are al-

lowed to move to their eight intermediate neighbours with eight different velocities $e_i (i = 1, \dots, 8)$. A particle can remain at the node, which is equivalent to moving with zero velocity e_0 . Referring to

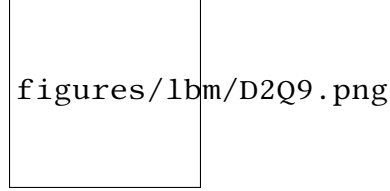


Fig. 1.1 The Lattice Boltzmann discretisation and D2Q9 scheme: (a) a standard LB lattice; (b) D2Q9 model (?)

the numbering system shown in Figure ??, these nine discrete velocity vectors are defined as:

$$\begin{cases} e_0 = (0, 0) \\ e_1 = C(1, 0); e_2 = C(0, 1); e_3 = C(-1, 0); e_4 = C(0, -1); \\ e_5 = C(1, 1); e_6 = C(-1, 1); e_7 = C(-1, -1); e_8 = C(1, -1); \end{cases} \quad (1.1)$$

in which C is the lattice speed, which is defined as:

$$C = h/\Delta t \quad (1.2)$$

where Δt is the discrete time step. The primary variables in the Lattice Boltzmann formulation are called the *fluid density distribution functions*, f_i , each relating the portable amount of fluid particles moving with the velocity e_i along the i^{th} direction at each node. The macroscopic variables are defined as functions of the particle distribution functions, according to:

$$\begin{aligned} \rho &= \sum_{i=0}^{\beta-1} f_i \quad (\text{macroscopic fluid density}) \\ \text{and} \\ \vec{u} &= \frac{1}{\rho} \sum_{i=0}^{\beta-1} f_i \vec{e}_i \quad (\text{macroscopic velocity}) \end{aligned} \quad (1.3)$$

where $i \in [0, \beta - 1]$ is an index spanning the discretized momentum space. There are nine fluid density distribution functions, $f_i (i = 0, \dots, 8)$, associated with each node in the $D2Q9$ model. The evolution of the density distribution function at each time step for every lattice point is governed by:

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{1}{\tau} [f_i(\mathbf{x}, t) - f_i^{\text{eq}}(\mathbf{x}, t)] \quad (i = 0, \dots, 8) \quad (1.4)$$

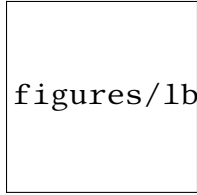
where for any grid node \mathbf{x} , $\mathbf{x} + \mathbf{e}_i \Delta t$ is its nearest node along the direction i . τ is a non-dimensional relaxation time parameter, which is related to the fluid viscosity; and f_i^{eq} is termed as the equilibrium distribution function, and is defined as:

$$\begin{cases} f_0^{\text{eq}} = w_0 \rho (1 - \frac{3}{2C^2} \mathbf{v} \cdot \mathbf{v}) \\ f_i^{\text{eq}} = w_i \rho (1 + \frac{3}{C^2} \mathbf{e}_i \cdot \mathbf{v} + \frac{9}{2C^2} (\mathbf{e}_i \cdot \mathbf{v})^2 - \frac{3}{2C^2} \mathbf{v} \cdot \mathbf{v}) \end{cases} \quad (i = 0, \dots, 8) \quad (1.5)$$

in which, w_i represents the fixed weighting values:

$$w_0 = \frac{4}{9}; \quad w_{1,2,3,4} = \frac{1}{9}; \quad w_{5,6,7,8} = \frac{1}{36} \quad (1.6)$$

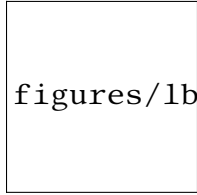
In addition, the right-hand side of Eq. ?? is often denoted by $f_i(\mathbf{x}, t_+)$ and termed the post collision distribution. The LB equation ?? ensures conservation of total mass and total momentum of the fluid particles at each lattice node. It essentially consists of two phases: *collision* and *streaming*. The collision phase computed in the right-hand side of Eq. ??, involves only those variables which are associated with each node \mathbf{x} , and therefore is a local operation. The streaming phase then explicitly propagate the updated distribution functions at each node to its neighbours $\mathbf{x} + \mathbf{e}_i \Delta t$, where no computations are required and only data exchange between neighbouring nodes are necessary. These features, together with the explicit time-stepping nature and the use of a regular grid, make LB computationally efficient, simple to implement and natural to parallelism (?). The streaming step involves the translation of the distribution functions to their neighbouring sites according to the respective discrete velocity directions, as illustrated in Figure ?? in the *D2Q9* model.



figures/lbm/stream.pdf

Fig. 1.2 Illustration of the streaming process on a $D2Q9$ lattice. The magnitude of the distribution functions remains unchanged, but they move to a neighbouring node according to their direction.

The collision step, (illustrated in Figure ??) consists of re-distribution of the distribution functions to the local discretized Maxwellian equilibrium distribution functions, but in such a way that local mass and momentum are invariants. In incompressible flows, the energy conservation is equivalent to momentum conservation (?).



figures/lbm/collision.pdf

Fig. 1.3 Illustration of the collision process on a $D2Q9$ lattice. The local density ρ and velocity \mathbf{v} are conserved, but the distribution functions change according to the relaxation to local Maxwellian rule

The standard macroscopic fluid variables, density ρ and velocity \mathbf{v} , can be recovered from the distribution functions as:

$$\rho = \sum_{i=0}^8 f_i; \quad \rho \mathbf{v} = \sum_{i=0}^8 f_i \mathbf{e}_i \quad (1.7)$$

The fluid pressure field p is determined by the following equation of state:

$$p = C_s^2 \rho \quad (1.8)$$

where C_s is termed the fluid speed of sound and is related to the lattice speed C by

$$C_s = C/\sqrt{3} \quad (1.9)$$

The kinematic viscosity of the fluid ν is implicitly determined by the model parameters h , Δt and τ as:

$$\nu = \frac{1}{3} \left(\tau - \frac{1}{2} \right) \frac{h^2}{\Delta t} = \frac{1}{3} \left(\tau - \frac{1}{2} \right) Ch \quad (1.10)$$

which indicates that these three parameters are related to each other and have to be appropriately selected to represent the correct fluid viscosity. An additional constraint to the parameter selection is the lattice speed C , which must be sufficiently large in comparison with the maximum fluid velocity v_{max} in the simulation, to ensure accuracy of the solution. This is measured by the ‘computational’ Mach number, M_a , defined by

$$M_a = \frac{v_{max}}{C} \quad (1.11)$$

Theoretically, the Mach number is required to be $M_a \ll 1$. In practice, M_a should be, at least smaller than 0.1 (?). From a computational point of view, it is more convenient to choose h and τ as two independent parameters and Δt as the derived parameter, using the following equation.

$$\Delta t = \left(\tau - \frac{1}{2}\right) \frac{h^2}{3\nu} \quad (1.12)$$

It can be observed that τ has to be $\tau > 1/2$ (?). Since there is no a priori estimation available to determine appropriate values of h and τ for a fluid flow problem with given fluid viscosity ν , a *trial and error* approach is employed to obtain results satisfying the requirement of smaller *Mach Number*. This is similar to the choice of an appropriate Finite Element mesh size, without automatic adaptive mesh techniques.

1.1.2 Boundary conditions

Boundary conditions (BC) form an important part of any numerical solutions. In many cases, the boundary conditions can strongly influence the accuracy of the algorithm. The velocity and pressure are not primary variables in the Lattice Boltzmann method, hence the standard pressure, velocity, and mixed boundary conditions cannot be imposed directly. Hence, alternative conditions in terms of the distribution functions are adopted to describe the boundary conditions. Various boundary conditions used in the present study are discussed below.

Periodic boundary condition

The simplest type of boundary condition is the periodic boundary. In this case, the domain is folded along the direction of the periodic boundary pair. From the perspective of submarine landslide modelling, the periodic boundary conditions are useful for preliminary analysis, as they imply higher degree of symmetry of the fluid domain. Further information on the periodic boundary condition can be found in ?.

No-slip boundary condition

The most commonly adopted BC in the Lattice Boltzmann approach is the no-slip BC, especially the simple bounce-back rule, which is quite elegant and surprisingly accurate. The basic idea is that the incoming distribution functions at a wall node are reflected back to the original fluid nodes, but with the direction rotated by π radians. The bounce-back boundary condition is one of benefits of the Lattice Boltzmann method, as it is trivial to implement and it allows one to effortlessly introduce obstacles into the fluid domain. However, the boundary conditions have been proven to be only first-order accurate in time and space (?). A straightforward improvement is to consider the wall-fluid interface to be situated halfway between the wall and fluid lattice nodes (?). It involves, defining the *solid* nodes as those lying within the stationary wall regions, and the *fluid* nodes otherwise. Then if i is the direction between a fluid node n_1 and n_2 to be reflected back along the direction they came from, i.e.

$$f_{-i}(\mathbf{x}, t + \Delta t) = f_i(\mathbf{x}, t_+) \quad (1.13)$$

where $-i$ denotes the opposite direction of i . The bounce back rule is illustrated in Figure ???. This simple rule ensures that no tangential velocity exists along the fluid-wall interface, thereby a non-slip condition is imposed, and can be extended to any shapes or objects in a fluid flow. The slip boundary conditions have similar treatment to the non-slip condition, except that the distribution functions are reflected in the boundary instead of bounce-back (?).

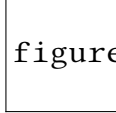

 figures/lbm/bounce.pdf

Fig. 1.4 Half-way bounce back algorithm for the $D2Q9$ model adopted after ?

Pressure and velocity boundary condition

The pressure boundary condition can be imposed in Lattice Boltzmann by specifying a fluid density at the pressure boundary. The velocity boundary condition can be specified in a similar way (?). The pressure and velocity boundary conditions contribute additional equation(s) to determine the unknown distribution functions. In the velocity boundary, the equation is sufficient to determine the unknown distribution functions in $D2Q9$ model, however the pressure boundary conditions require additional constitutive laws to determine the unknown distribution functions.

1.1.3 Turbulence in Lattice Boltzmann Method

The above formulation of Lattice Boltzmann problem has been successfully applied to many fluid flow problems, however it is restricted to flows with low Reynold's number. Modelling fluids with low viscosity like water and air remains a challenge, necessitating very small values of h and/or τ very close to 0.5 (?). The standard Lattice Boltzmann can deal with laminar flows, while practical problems with small kinematic viscosity are often associated with flows having large Reynold numbers, i.e. flows which are turbulent in nature. The turbulent flows are characterized by the occurrence of eddies with multiple scales in space, time and energy.

The Large eddy simulation (LES) is the most widely adopted approach to solve turbulent flow problems. It directly solves the large scale eddies, which carry the predominant portion of the energy, and the smaller eddies are modelled using a sub-grid approach. The separation of scales is achieved by filtering of the Navier-Stokes equations, from which the resolved scales are directly obtained and unresolved scales are modelled by a one-parameter Smagorinski sub-grid methodology, which assumes that the Reynold's stress tensor is dependent only on the local strain rate (?). It involves parametrizing the turbulent energy dissipation in the flows, where the larger eddies extract energy from the mean flow and ultimately transfer some of it to the smaller eddies which, in turn, pass the energy to even smaller eddies, and so on up to the smallest scales. At the smallest scale, the eddies convert the kinetic energy into the internal energy of the fluid. At this scale, the viscous friction dominates the flow (?). Incorporating the LES approach in Lattice Boltzmann technique, the governing equation of the density distribution function in Eq. (??) is modified as:

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{1}{\tau_*} [f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)] \quad (i = 0, \dots, 8) \quad (1.14)$$

The effect of the unresolved scale motion is taken into account by introducing an effective collision relaxation time scale τ_t , so that the total relaxation time τ_* is written as:

$$\tau_* = \tau + \tau_t \quad (1.15)$$

where τ and τ_t are respectively the standard relaxation times corresponding to the true fluid viscosity ν and the turbulence viscosity ν_t , defined by a sub-grid turbulence model. The new viscosity ν_*

corresponding to τ_* is defined as:

$$v_* = v + v_t = \frac{1}{3}(\tau_* - \frac{1}{2})C^2\Delta t = \frac{1}{3}(\tau + \tau_t - \frac{1}{2})C^2\Delta t \quad (1.16)$$

$$v_t = \frac{1}{3}\tau_t C^2\Delta t \quad (1.17)$$

In Smagorinski model adopted in the present study, the turbulent viscosity v_t is calculated from the filtered strain rate tensor as:

$$v_t = (S_c h)^2 \bar{S}; \quad \bar{S} = \sqrt{\sum_{ij} \tilde{S}_{ij} \tilde{S}_{ij}} \quad (1.18)$$

where S_c is the Smagorinski constant found to be close to 0.03 (?). The Smagorinski model is easy to implement and the Lattice Boltzmann formulation remains unchanged, except for the use of a new turbulence-related viscosity τ_* .

1.2 Validation of Lattice Boltzmann Method

The Lattice Boltzmann method implemented in the present study is validated by comparing the LBM simulation of laminar flow of water through a circular pipe with the closed form solution for laminar flow in a pipe. In the present study, water ($\rho = 1000 \text{ kg/m}^3$) is simulated to flow through a circular pipe of diameter 'D' = 0.04 m and length, i.e. perimeter, 'L' = 0.1 m. Periodic boundary conditions are applied at either end of the pipe, which simulates the condition of a continuous flow of fluid in a closed circular pipe. The parameters adopted in the LBM simulation is summarized in Table ???. Sufficient time is allowed for the flow to travel beyond the required development length so as to develop in to a fully laminar flow ?. The development length required for a flow to be fully laminar is:

$$X_D/D = [(0.619)^{1.6} + (0.0567 R_e)^{1.6}]^{1/1.6} \quad (1.19)$$

where X_D is the development length and R_e is the Reynold's number. The velocity profile of water in the simulated segment at the end of the simulation is presented in Figure ??. Figure ?? shows the horizontal velocity profile obtained at L/2 after 50,000 time steps. The maximum horizontal velocity of 0.037863 m/s is observed in the middle of the pipe. The velocity profile obtained (Figure ??), is compared with the closed-form based on the Haygen-Poiseuille's flow equation for no-slip boundary condition (?):

$$v_x = \frac{\Delta P}{2\mu L} [\frac{D^2}{4} - y^2] \quad (1.20)$$

where v_x is the horizontal velocity (m/s); ΔP is the pressure gradient, μ dynamic viscosity of the fluid, The horizontal velocity profile estimated at 'L/2' based on the closed form solution is also plotted in Figure ??. It can be observed from the figure that the Lattice Boltzmann method predicts the laminar flow behaviour, however the maximum horizontal velocity is underestimated by 0.009 % in comparison to the closed form solution. To validate the Lattice Boltzmann code developed in the present study, the results of the laminar flow through a pipe is compared with those obtained from the Computational Fluid Dynamics (CFD) simulations performed using *ANSYS FLUENT*. The CFD analysis involves solving problems associated with fluid flows using numerical approximations

Table 1.1 LBM parameters used in simulating laminar flow through a circular pipe

Parameter	Value
Fluid	Water
Density ρ	1000 kg/m^3
Relaxation parameter τ	0.51
Kinematic viscosity	$1 \times 10^{-6} \text{ m}^2/\text{s}$
Dynamic viscosity	$1.002 \times 10^{-3} \text{ N s/m}^2$
Acceleration	$1 \times 10^{-6} \text{ m/s}$
Maximum number of iterations	50,000
Type of flow	laminar
Diameter of pipe 'D'	0.04 m
Length of the pipe 'L'	0.1 m
Boundary condition	Periodic boundary
Maximum horizontal velocity at L/2	0.037863 m/s
Theoretical maximum horizontal velocity (Poiseuille's flow)	0.0379 m/s
Error in predicting horizontal velocity	0.009 %

to solve Navier-Stokes equations, which defines any single-phase flow. The Finite Volume Method (FVM) is a common approach used in the CFD codes, like FLUENT. It involves solving the governing equation over the discretized control volume. This guarantees the conservation of fluxes over a particular control volume. The finite volume equations yield governing equations of the form:

$$\frac{\partial}{\partial t} \int \int \int Q dV + \int \int F dA = 0 \quad (1.21)$$

where Q is the vector of conserved variables, F is the vector of fluxes in the Navier-Stokes equation, V is the volume of control volume element, and A is the surface area of the control volume element. A 2D rectangular plane of length 1m and height 0.04m is discretized into 400 cells of size 1^{-2} m (see Figure ??). A constant velocity is applied at the inlet. Water ($\rho = 998.2 \text{ kg/m}^3$, $\text{viscosity}'\eta' = 1 \times 10^{-3} \text{ N s/m}^2$) is allowed to flow through the pipe and it develops into a fully laminar flow. The Least-Squares cell based approach was adopted to solve the gradient, and 100 iteration steps were carried out until the solution converges. The velocity contour profile obtained from the CFD analysis is presented in Figure ?. The velocity profile at the cross-section 'L/4' is plotted in Figure ?. The velocity profile matches the analytical Poiseuille's solution and the LBM simulation of laminar flow. Having validated the CFD analysis against Poiseuille's equation, the CFD analysis can now be used to validate the LBM simulation of fluid flow around a rectangular obstacle, to study the capability of the LBM technique to simulate vortex effects in fluid flows around obstacles.

The efficiency of Lattice Boltzmann method to include solid walls/particles is evaluated by placing a solid wall of Length 'D/2' at length 'L/4' in the pipe. The velocity profile obtained after 50,000 iterations is presented in Figure ?. The formation of eddies at the corners of the wall along the flow direction is captured by the LB method. The horizontal velocity profile obtained at 'L/4' is presented in Figure ?. The occurrence of vortex effect in the flow around the edges of the obstacle can be visualized in Figure ?. The velocity profile obtained from the LBM simulation of the fluid flow around a rectangular body compares qualitatively with the FE analysis performed by ?. The CFD analysis of the same problem was performed using ANSYS FLUENT. The control volume is discretized into 10,000 cells and a constant velocity is applied in the inlet. The velocity contour and vector profile

obtained from the CFD analysis are presented in figures ?? and ?. Similar maximum horizontal velocities are observed in both the LBM and the CFD analyses. The velocity vectors obtained from the CFD analysis (see Figure ??) proves the occurrence of eddies in the LBM analysis of flow around an obstacle. The CFD analysis is found to over predict slightly the maximum horizontal velocity in comparison with the LBM simulation. The discrepancy in the horizontal velocity profile (Figure ??) can be attributed to the relaxation parameter used in the LBM, which is obtained by trial and error procedure. Thus, it can be concluded that the Lattice Boltzmann method is a suitable form of numerical representation of Navier-Stokes equation to model fluid flows.

figures/lbm/Velocity.png

Fig. 1.5 Velocity profile for a laminar flow through a pipe

figures/lbm/Mesh.png

Fig. 1.6 Finite Volume Mesh used in the CFD analysis of laminar flow through a pipe

figures/lbm/LaminarFlow.png

Fig. 1.7 Velocity contour obtained from the CFD analysis of laminar flow through a pipe

1.3 Coupled Lattice Boltzmann - Molecular Dynamics for fluid-particle interactions

In principle, the conventional Finite Element and Finite Volume based approaches for the solving the Navier-Stokes equations with moving boundaries and/or structural interaction (?) can be applied to particle fluid interaction problems. The common feature of these approaches is to model the interaction between the fluid and the particles to a high degree of accuracy, but the main computational challenge is the need to continuously generate new geometrically adapted meshes to circumvent severe mesh distortion, which is computationally very intensive (?). The Lattice Boltzmann approach has the advantage of accommodating large particle sizes and the interaction between the fluid and the moving particles can be modelled through relatively simple fluid - particle interface treatments. Further, employing the Discrete Element Method (DE) to account for the particle/particle interaction naturally leads to a combined LB - DEM solution procedure. The Eulerian nature of the Lattice Boltzmann formulation, together with the common explicit time step scheme of both the Lattice Boltzmann and the Molecular Dynamics makes this coupling strategy an efficient numerical procedure for the simulation of particle-fluid systems. LBM-MD technique is a powerful predictive tool for gaining insights into many the fundamental physical phenomena in the fluid-solid interaction domains. Such a coupled methodology was first proposed by (?) for simulating particle-fluid systems dominated by particle-fluid and particle-particle interactions. To capture the actual physical behaviour of the fluid-particle system, it is essential to model the boundary condition between the fluid and the particle as a non-slip boundary condition, i.e. the fluid near the particle should have

figures/lbm/LBMCFD.pdf

Fig. 1.8 LBM and CFD simulation of laminar flow through a circular pipe with and without an obstacle

figures/lbm/ObstacleLBM.png

Fig. 1.9 LBM simulation of velocity profile for a laminar flow through a pipe with obstacle in the middle

similar velocity as the particle boundary. The solid particles inside the fluid are represented by lattice nodes. The discrete nature of lattice will result in stepwise representation of the surfaces, which are circular, this is neither accurate nor smooth, unless sufficiently small lattice spacing is adopted.

Modified bounce back rule

To accommodate the movement of solid particles in the commonly adopted bounce-back rule (see section ??). ? modified the ‘no-slip’ rule for a given boundary link i to be:

$$f_i(\mathbf{x}, t + \Delta t) = f_i(\mathbf{x}, t_+) - \alpha_i \mathbf{e}_i \cdot \mathbf{v}_b \quad (\alpha_i = 6w_i\rho/C_s^2) \quad (1.22)$$

where $f_i(\mathbf{x}, t_+)$ is the post collision distribution at the fluid or solid boundary node \mathbf{x} and \mathbf{v}_b is the velocity at the nominal boundary point at the middle of the boundary link i :

$$\mathbf{v}_b = \mathbf{v}_c + \boldsymbol{\omega} \times (\mathbf{x} + \mathbf{e}_i \Delta t/2 - \mathbf{x}_c) \quad (1.23)$$

in which \mathbf{v}_c and $\boldsymbol{\omega}$ are the translational and angular velocities at the mass centre of the solid particle, respectively. \mathbf{x}_c and $\mathbf{x} + \mathbf{e}_i \Delta t/2$ are the coordinates of centre and the nominal boundary point, respectively. The impact force on the solid particle from the link is defined as:

$$\mathbf{F}_i = 2[f_i(\mathbf{x}, t_+) - \alpha_i \mathbf{e}_i \cdot \mathbf{v}_b] / \Delta t \quad (1.24)$$

The corresponding torque \mathbf{T}_i , produced by the force with respect to the centre of the particle is computed as:

$$\mathbf{T}_i = \mathbf{r}_c \times \mathbf{F}_i (\mathbf{r}_c = \mathbf{x} + \mathbf{e}_i \Delta t/2 - \mathbf{x}_c) \quad (1.25)$$

Then the total hydrodynamic force and torque exerted on the particle can be calculated by summing up the forces and torques from all the related boundary links:

$$\mathbf{F} = \sum_i \mathbf{F}_i; \quad \mathbf{T} = \sum_i \mathbf{T}_i \quad (1.26)$$

figures/lbm/ObstacleCFD.png

Fig. 1.10 CFD simulation of velocity contour for a laminar flow through a pipe with obstacle in the middle

figures/lbm/OVV.png

Fig. 1.11 CFD simulation of velocity vector profile for a laminar flow through a pipe with obstacle in the middle

? described a methodology that minimises the oscillations resulting from particles crossing lattice at very large speed. The fluid/particle force interaction method with momentum exchange method is coupled with the treatment of moving curved boundaries scheme (??). The simulation of the moving curved particle surfaces results in the intersection of links between two nodes at arbitrary distances (?). These distance values are called as delta values:

$$\delta = \frac{\text{Distance between fluid nod and particle surface}}{\text{Distance between fluid node and particle node}} \in [0, 1] \quad (1.27)$$

For each pair of neighbouring fluid and particle nodes, a delta value has to be calculated. Delta values of zero are not possible as nodes on the surface are considered as the particle node. The algorithm for computation of the δ value is presented in ?. Figure ?? shows the three possible situations for delta values in the interval of $[0,1]$. Since the fluid particles in the LBM are always considered to move at the rate of a lattice per time step ($\delta \mathbf{x} / \delta t$), for delta values smaller than 0.5. For δ values larger than 0.5, the fluid particles would come to rest at an intermediate node \mathbf{x}_i . In order to calculate the reflected distribution function in node \mathbf{x}_f , an interpolation scheme has to be applied. The linear interpolation scheme of ?? is used in the present study, which uses a single equation, irrespective of the value of δ being less or larger than 0.5, to the reflected distribution function, which is computed as:

$$f_{\bar{\alpha}}(\mathbf{x}_f, t + \delta t) = \frac{1}{1 + \delta} \cdot [(1 - \delta) \cdot f_{\alpha}(\mathbf{x}_f, t + \delta t) + \delta \cdot f_{\alpha}(\mathbf{x}_b, t + \delta t) + \delta \cdot f_{\bar{\alpha}}(\mathbf{x}_{f2}, t + \delta t) - 2w_a \rho_w \frac{3}{c^2} \mathbf{e}_a \cdot \mathbf{u}_w] \quad (1.28)$$

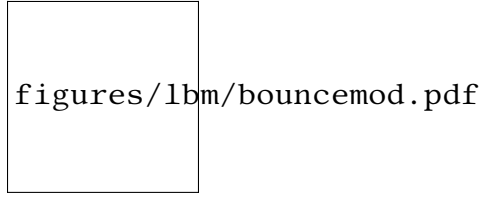
where w_{α} is the weighting factor, ρ_w is the fluid density in node \mathbf{x}_f , and \mathbf{u}_w is the velocity at the bounce-back wall. In order to couple the fluid-particle interaction, the LBM approach is extended by adopting a force integration scheme to calculate the fluid force acting on the particle surface, and the momentum exchanged method described earlier. The physical force acting on particle agglomerate is calculated as the sum over all fluid/particle node pairs, resulting in:

$$F = \sum_{\mathbf{x}_b} \sum_{\alpha=1}^{19} \mathbf{e}_{\alpha} [f_{\alpha}(\mathbf{x}_b, t) + f_{\bar{\alpha}}(\mathbf{x}_f, t)] \delta \mathbf{x} / \delta t \quad (1.29)$$

After the force calculations, the coupled rigid body physics can be simulated in order to move the particle agglomerates according to the applied forces. The total hydrodynamic forces and torque exerted on a particle can be computed as (??):

$$\mathbf{F}_f = Ch \left[\sum_n (\beta_n \sum_i f_i^m \mathbf{e}_i) \right] \quad (1.30)$$

$$\mathbf{T}_f = Ch \left[\sum_n (\mathbf{x}_n - \mathbf{x}_c) \times (\beta_n \sum_i f_i^m \mathbf{e}_i) \right] \quad (1.31)$$

Fig. 1.12 Bounce back boundaries for different values of δ

The summation is over all lattice nodes covered by the particle and \mathbf{x}_n represents the coordinate of the lattice node n . When particles are not in direct contact among themselves, but are driven by the fluid flow and body force, i.e. gravity, their motion can be determined by Newton's equation of motion:

$$m \mathbf{a} = \mathbf{F}_f + m \mathbf{g} \quad (1.32)$$

$$\mathcal{J} \ddot{\theta} = \mathbf{T}_f \quad (1.33)$$

where m and \mathcal{J} are respectively the mass and the moment of inertia of a particle; and $\ddot{\theta}$ is the angular acceleration; \mathbf{g} is the gravitational acceleration; \mathbf{F}_f and \mathbf{T}_f are respectively the hydrodynamic forces and torque. The equation can be solved numerically by an explicit numerical integration, such as central difference scheme. The interaction between solid particles and the solid particles with the walls are dealt with Molecular Dynamics technique. To solve the coupled MD-LBM, the hydrodynamic force exerted and the static buoyancy force are considered by reducing the gravitational acceleration to $(1 - \rho/\rho_s)\mathbf{g}$, where ρ_s is the density of the particles. When taking into account all forces acting on an element, the dynamic equations of Molecular Dynamics can be expressed as:

$$m \mathbf{a} + c \mathbf{v} = \mathbf{F}_c + \mathbf{F}_f + m \mathbf{g} \quad (1.34)$$

where \mathbf{F}_c denotes the total contact forces from other elements and/or the walls; c is a damping coefficient; and the term $c\mathbf{v}$ represents a viscous force that accounts for the effect of all possible dissipation forces in the system including energy lost during the collision between particles. Considering a linear contact model:

$$\mathbf{F}_c = k_n \delta \quad (1.35)$$

where k_n is the normal stiffness and δ is the overlap, the critical time step associated with the explicit integration is determined as (?):

$$\Delta t_{cr} = 2(\sqrt{1 + \xi^2} - \xi)/\omega \quad (1.36)$$

where $\omega = \sqrt{k_n/m}$ is the local contact natural frequency and $\xi = c/2m\omega$ is the critical damping ratio. the actual time step used for the integration of the Molecular Dynamics equations is:

$$\Delta t_D = \lambda \Delta t_{cr} \quad (1.37)$$

The time step factor λ is chosen to be around 0.1 to ensure both stability and accuracy (?). When combining the Molecular Dynamics modelling of the particle interaction with the LB formulation, a minor issue arises. There are now two time steps: Δt for the fluid flow and Δt_D for the particles.

Table 1.2 Micromechanical parameter used to simulate soil grains

Parameter	Value
Normal and shear stiffness of disks in column	$4 \times 10^6 \text{ N/m}$
Normal and shear stiffness of walls	$4 \times 10^8 \text{ N/m}$
Number of disks	184
Radius of disks, R	0.8 to 1.19 mm
Particle friction coefficient, μ	0.577
Coefficient of restitution, Γ	0.2
Density of spheres, ρ	2650 kg/m^3
Size reduction to consider fluid flow	0.2R

Since Δt_D is normally smaller than Δt , Δt_D is slightly reduced to a new value Δt_s so that Δt and Δt_s have an integer ration n_s :

$$\Delta t_s = \frac{\Delta t}{n_s} \quad (n_s = [\Delta t / \Delta t_D] + 1) \quad (1.38)$$

This basically results in a sub-cycling time integration for the Molecular Dynamics part. At every step of the fluid computation, n_s sub-steps of integration are performed for the Molecular Dynamics Eq. ?? using the time step Δt_s . The hydrodynamic force \mathbf{F}_f is unchanged during the sub-cycling.

A granular column collapse problem in fluid is simulated using the developed LBM-MD approach. The parameters used for simulating the fluid is the same as shown in Table. ?. 180 soil grains of density $\rho = 2600 \text{ kg/m}^3$, are packed into a rectangular column and are allowed to collapse under gravity. The micromechanical parameters used for simulating the solid grains are presented in Table. ?. The initial packing of disks in the rectangular column is presented in Figure ?. For every single step of LBM simulation 100 sub-steps of Molecular Dynamics is solved. Figure ?? shows the collapse of granular column in fluid in progress. The formation of vortex behind the grains is captured efficiently by the coupled MD-LBM technique, indicating that the approach is capable of successfully modelling the fluid-particle coupled systems.

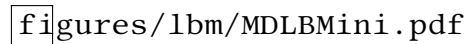
figures/lbm/MDLBMini.pdf

Fig. 1.13 Initial sample of granular column in fluid, showing the variation in the grain size

figures/lbm/MDLBM.png

Fig. 1.14 Collapse of a granular column in fluid in progress, showing the variation in the magnitude of horizontal velocity