

Implementation of Individual Based (IB) Models within LAMMPS to model biofilm dynamics in a waste water treatment plant

Prashant Gupta, Curtis Madsen, Jayathilake Pahala Gedara

July 22, 2015

Contents

1	Introduction	1
1.1	Activated Sludge process	1
1.2	Floc formation	1
2	Individual based modelling (IBm)	1
2.1	Introduction	3
2.2	Purpose of the model	3
2.3	Model variables and principles	4
2.4	IBm submodels	4
2.4.1	Cell Growth	5
2.4.2	Cell Division	6
2.4.3	Cell Death	6
2.4.4	EPS production and excretion	7
2.4.5	EPS-Adhesion forces	8
2.5	Physical models	8
2.5.1	Fluid equations	9
2.5.2	Discrete element method	10
2.5.3	Neighbour lists	11
2.5.4	Collision model	11
2.5.5	Interphase momentum transfer	13
2.5.6	Drag model closures	14
2.5.7	Drag models	14
2.5.8	Algorithm for inter-phase momentum exchange	15
2.5.9	Numerical methods	16
2.6	Stochastic processes: Scope of improvement	16

3	LAMMPS	17
3.1	Introduction to LAMMPS	17
3.2	LAMMPS working methodology	17
3.3	Operating systems	18
3.4	Pre-compilation instructions	18
4	LAMMPS-NCL and getting started	18
4.1	Downloading LAMMPS-NCL	19
4.2	LAMMPS extension	19
4.3	Atom type	19
5	Input script generated for the NUFEB-Bio simulations	20
5.1	Cell growth	21
5.2	Cell Division	22
5.3	Cell Death	23
5.4	EPS extract	24
6	Shoving algorithms	24
7	Graphical User interface (GUI)	26
8	Visualization and post-processing	27

List of Tables

List of Figures

1	Activated sludge process description for a waste water treatment plant (WWTP)	2
2	Schematic of typical length scales for multi-scale modelling of a activated sludge process based Waste water treatment plant (WWTP)	2
3	Growth and decay rates for AOB, NOB, HET and EPS particles.	5
4	Division of daughter cells from the mother cell, randomly oriented around the centre of mother cell	6
5	EPS bound to the the HET particle, extracted to an EPS particle according to the ratio between the EPS shell and the HET particle radius.	8
6	Schematic of two particles i and j in contact and position vectors \mathbf{r}_i , \mathbf{r}_j , respectively, with overlap δ_{ij} . Resultant normal and tangential forces acting on each particle are shown here.	12
7	Input script for NUFEB simulation of biofilm dynamics.	21
8	Command line for the growth fix in LAMMPS	21

9	Command line for the division fix in LAMMPS	22
10	Command line for the death fix in LAMMPS	23
11	Command line for the EPS extract fix in LAMMPS	24
12	Graphical User Interface (GUI) for NUFEB model	27

1 Introduction

The portable water shortage problem is widely discussed and demands a judicious usage of available resources. However, several countries (mostly under-developed and developing) suffer from extreme shortage of the water resources. Waste water treatment plants have served the societies for several decades now. Activated sludge process treatment plants are the most common types as it provides an advantage of better efficiency and economics.

1.1 Activated Sludge process

Activated sludge is defined as a microbial mass cultivated in order to break down the biomass into different components such as carbon dioxide, water and other nitrogen or phosphorous based compounds. Activated sludge process involves 3 different component mechanisms:

- A reactor in which the microbes are kept in suspension, aerated, and in contact with the waste they are treating.
- Settlement of bulkier solid (activated sludge).
- Recycling system for returning activated sludge back.

In present waste water industries, many variants of activated sludge processes, primarily variations on how the activated sludge is recycled. Present study attempts to model following activated sludge process as described in the figure 1.

1.2 Floc formation

A floc is described as an aggregate of microbes bonded together by adhesive material (EPS) secreted by them. These clusters formations, often called as activated sludge, play a key role in the functioning of any waste water treatment plant (WWTP). Efficiency or the energy requirement of either of the primary and secondary tanks in an activated sludge process, depends upon the floc size, settlement or motility. Furthermore, the floc formation is governed by the microbial components adherence and cluster breakages.

2 Individual based modelling (IBm)

Pilot scale plants and laboratory scale experiments of WWTP are expensive, cumbersome, non-invasive and often can not provide information at the micro-scale, which is required for operational optimization of WWTP. Modelling of WWTP is challenging due to wide separation of temporal and

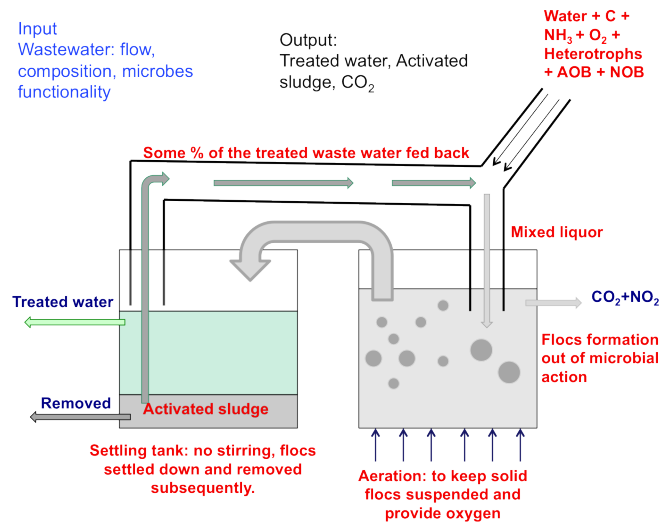


Figure 1: Activated sludge process description for a waste water treatment plant (WWTP)

spatial scales at which biological and physical processes key roles. Henceforth, a multi-scale modelling strategy is required to relate the microscopic microbial actions (often less than a micrometer size) to the macroscopic bulk WWTP operation (meter size scale). Identification of different length scales typical to a WWTP is presented can be shown in the figure 2.

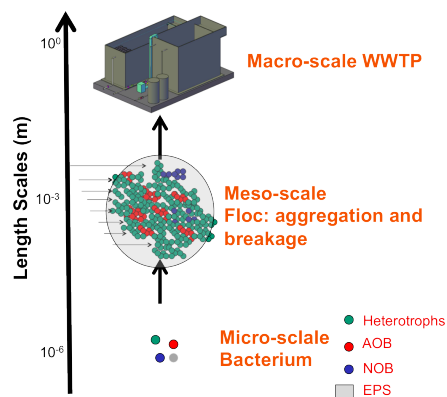


Figure 2: Schematic of typical length scales for multi-scale modelling of a activated sludge process based Waste water treatment plant (WWTP)

2.1 Introduction

Discrete unit models such as the cellular automaton [Picioreanu *et al.*, 1998a] or the individual-based models [Kreft *et al.*, 1998], have been first developed and are now becoming widely applied to study effects of spatially multidimensional gradients in biofilms. Models which use individuals as a basic unit have occasionally been used in ecology since 1970s, but only since the visionary review of Huston *et al.* [1988] has individual-base modelling been an explicitly delineated approach of ecological modelling. Individual-based modelling refers in the following to simulation models that treat individuals as unique and discrete entities which have at least one property in addition to age that changes during the life cycle, e.g. weight, rank in a social hierarchy, etc. For biofilm modelling, Kreft *et al.* [2001]; Picioreanu *et al.* [1998a] initially proposed the Individual-based Modelling (IbM) as a proper tool compared with existing Cellular Automaton (CA) method used for biofilm modelling. In CA models, the biofilm is represented by collection of bricks, in which each brick is represented by a Cartesian grid ([Eberl *et al.*, 2000; Picioreanu *et al.*, 2000, 1998a,b]). In IbM, bacterial cells are represented as hard spheres, with each cell having, besides a variable volume and mass, a set of variable growth parameters, position, velocity, genotype etc. The IbM model consists of two parts: one deals with the growth and behaviour of individual bacteria as autonomous agents (i.e., biological processes); the other deals with the substrate and product diffusion and reaction and fluid flow (i.e., physical processes). Each cell grows by consuming the substrate and divides when a certain volume or mass is reached. The pressure build-up due to the growth of biomass is released by maintenance of a minimum distance between the neighbouring cells. In the traditional IbM approach, for each cell, the vector sum of all positive overlap radii with the neighbouring cells, which is called the shoving vector, is calculated and then the position of the cell is shifted in the direction opposite to this vector 3 sum. The substrate concentration is governed by a convection-diffusion-reaction equation and this transport equation is solved in a fixed Cartesian grid. Figure 1 shows the typical computation domain associated with IbM of biofilms. Basically it has three sub-domains each for biofilm, mass transfer boundary layer, and bulk fluid.

2.2 Purpose of the model

The model treats each microbes as individual entities interacting with each other and the surrounding environmental conditions. Various sub-models are employed for describing biology and physics at different length and temporal scales. The multi-physics/biological model developed can be used to investigate following key questions at different scales:

- Linking the microscopic (bacterium) to the mesoscopic (floc) to the

macroscopic bulk operational parameters.

- Floc formation mechanism: How the microscopic phenomenon such as microbial growth kinetics, division, death, nutrient availability lead to microbial floc formation or breakage?
- Effects of floc morphology and motility on the overall plant efficiency?
- How the hydrodynamics of the system can influence energy requirements of the WWTP?
- Effect of the nutrient availability spatially and temporally?

2.3 Model variables and principles

The solver treats the microbes as spherical particles with EPS shells bounded on them. Fluid is treated as a continuum. Nutrients are treated as concentrations in dissolved and convected through the flow fields. Given by their representations each components are solved accordingly.

- Microbes: solved in a Lagrangian framework. Each microbes is represented as a finite size sphere and has certain functionality. They are tracked throughout the simulations. The interactions between the microbes and with their ambient environment are complex and are described through different biological and physical sub-models described later. Following functional groups are considered: Heterotrophs (HET), Ammonia oxidizing bacteria (AOB), Nitrogen oxidizing bacteria (NOB), EPS particles and dead inert particles.
- Fluid: Fluid or the water phase is treated as a continuum media with microbes mostly suspended (neutrally buoyant). The fluid motion is solved in an Eulerian grid and takes into the account of the microbial suspension. Fluid and particle motion are tightly coupled by the hydrodynamics solver.
- Nutrients: The nutrients are treated as concentrations of the food supply in the water phase. The fluid solver supplies these nutrients to the microbes and hence contributes to their metabolism. Following nutrients relevant to the waste water treatment plant are taken into the account: Carbon substrates (S_s), Oxygen supply (O_2), Nitrates (NO_2), Nitrites (NO_3) and Ammonia (NH_4).

2.4 IBm submodels

Various cell level processes constitutes microbial working at cell level. Each of these processes can be termed as components of IBm model presented

here. Modelling and fundamentals of each of these processes are presented here:

2.4.1 Cell Growth

Commonly found microbes in a WWTP can be grouped into functional group: Heterotrophs (HET), Ammonia oxidizing bacteria (AOB) and Nitrogen oxidizing bacteria (NOB). Each of the functional groups consume different nutrients and compete with each other for survival. Present tool allows a functional group and its function to be defined accordingly. Following equations give the reactions and their description is presented in a matrix form (3):

$$\frac{dm_i}{dt} = Gm_i \quad (1)$$

Each species has its own growth rate (G) and can be described according to the equation 2. $\gamma_{HET} = 1$ and rest are zero, when solving for the HET equations. Same can be extended to AOB, NOB and EPS particles. R_j (j=1-9) are given in the table 3.

$$G = \gamma_{HET} (R_1 + R_4 + R_5) + \gamma_{AOB} (R_2) + \gamma_{NOB} (R_3) - \gamma_{EPS} (R_9) \quad (2)$$

HET bound EPS growth rate is given by:

$$G_{HET-EPS} = \frac{Y_{EPS}}{Y_{HET}} Gm_i \quad (3)$$

Reaction equations are primarily mass balance solved on the mesh discretization defined earlier.

Process	Rate (d ⁻¹)	
Aerobic growth of HET	$\mu_{m,HET} \frac{S_S}{K_{S,HET} + S_S} \frac{S_{O_2}}{K_{O_2,HET} + S_{O_2}}$	R ₁
Aerobic growth of AOB	$\mu_{m,AOB} \frac{S_{NH_4}}{K_{NH_4,AOB} + S_{NH_4}} \frac{S_{O_2}}{K_{O_2,AOB} + S_{O_2}}$	R ₂
Aerobic growth of NOB	$\mu_{m,NOB} \frac{S_{NO_2}}{K_{NO_2,NOB} + S_{NO_2}} \frac{S_{O_2}}{K_{O_2,NOB} + S_{O_2}}$	R ₃
Anoxic growth of HET on NO ₃	$\eta_H \mu_{m,HET} \frac{S_S}{K_{S,HET} + S_S} \frac{S_{NO_3}}{K_{NO_3,HET} + S_{NO_3}} \cdot \frac{K_{O_2,HET}}{K_{O_2,HET} + S_{O_2}}$	R ₄
Anoxic growth of HET on NO ₂	$\eta_H \mu_{m,HET} \frac{S_S}{K_{S,HET} + S_S} \frac{S_{NO_2}}{K_{NO_2,HET} + S_{NO_2}} \cdot \frac{K_{O_2,HET}}{K_{O_2,HET} + S_{O_2}}$	R ₅
Decay of HET	b_{HET}	R ₆
Decay of AOB	b_{AOB}	R ₇
Decay of NOB	b_{NOB}	R ₈
Decay of EPS	b_{EPS}	R ₉

Figure 3: Growth and decay rates for AOB, NOB, HET and EPS particles.

Each of the components of uptake or consumption is calculated separately and added together following superposition principle. The rate is defined according to Monod-kinetics where the kinetics factors are product together. Also, the rate of the growth/decay is proportional to the instantaneous mass of the substrate. The local nutrient availability for the growth of each cell is based upon the ambient conditions solved through the passive scalar transport of the nutrients. Fluid-biofilm coupling solver ensures that each cell is aware of its ambient nutrient conditions. Each of the cell has nutrient availability concentration attributes as: Soluble substrate (S_s), Oxygen (O_2), NO_2 , NO_3 , NH_4 .

2.4.2 Cell Division

Microbes can not grow forever (even with infinite food supply), division occurs due to sustenance of their sizes. Present model employs cell division rather empirically, by fixing up a maximum cell diameter, after which it divides. The model is implemented in following way: If the mass of a bacterial cell becomes greater than twice the mass of an inoculated individual bacterium, it divides into two daughter cells each. During the division process, the cell mass is split in a ratio randomly selected between 0.4-0.6. This generated two daughter cells from a parent cell. These daughter cells are oriented randomly around the centre of the parent cell. Figure 4 shows the division of daughter cells from the mother cell, randomly oriented around the centre of mother cell. This generation of newer particles occur at start of every time step and the forces equilibrated according to the overlaps, ensuring mechanical stability. The diameters of daughter cells are calculated using the cell mass and biomass density. Biomass density is constant for each species.

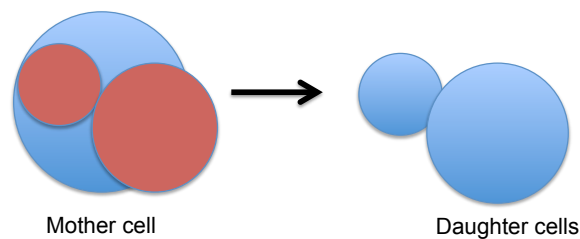


Figure 4: Division of daughter cells from the mother cell, randomly oriented around the centre of mother cell

2.4.3 Cell Death

Microbial death is modelled as a random stochastic process. Every so often, a few microbes becomes biologically inactive i.e. they would not consume

nutrient, grow or divide. These cells are treated as inactive particles by the solver and only participate in any physical/mechanical interactions with their neighbours. Depending on the total decay mass of HET, AOB, and NOB, the appropriate numbers of HET, AOB, and NOB cells are calculated and died. The mass of the dying cell shares between dead cell and carbon source. The mass of decaying EPS cells totally converts to carbon. Completely dissolved EPS particles are removed from the calculation. In the model this implemented by the total virtual dead mass of each species (HET or AOB or NOB) and is calculated from equations as below:

$$D = \gamma_{HET}R_6 + \gamma_{AOB}R_7 + \gamma_{NOB}R_8 \quad (4)$$

$$M_{virtual} = \sum Dm \quad (5)$$

where, D is the decay rate. At each time step (dt), number of cells equivalent to $M_{virtual}$ is set as dead (i.e. ID is changed to 5). These particles are randomly selected. When a cell is died, Y fraction of its mass remains in the dead cell and the other fraction (1-Y) is spontaneously converted to carbon substrate (S_s) and distributed to the Cartesian grid cell centre in which the particle resides.

2.4.4 EPS production and excretion

Microbes secrete extracellular polymeric substances (EPS) every so often as a waste product of their metabolic activities. EPS is secreted into their neighbouring environment and have known to lend structural integrity to the biofilms. Previous solvers have represented EPS in continuum and discrete manner. The solver works on the common knowledge that HETs excrete EPS, while AOB and NOB microbes do not. The present solver follows the iDynnomics approach of EPS treatment. Initially, EPS is accumulated as a extra shell beyond the HET particle (figure 5). It should be noted that the EPS density is much lower than the HET density. When the relative thickness of the EPS shell bound to HET particle exceeds a certain threshold value, almost half (random ratio between 0.4-0.6) of the EPS mass excretes as a separate EPS particle and positions next to the HET cell.

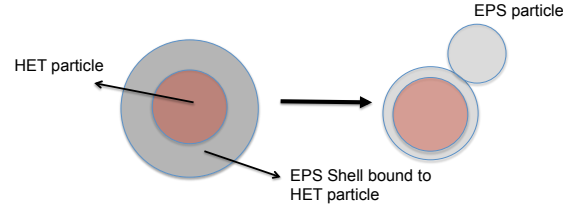


Figure 5: EPS bound to the the HET particle, extracted to an EPS particle according to the ratio between the EPS shell and the HET particle radius.

2.4.5 EPS-Adhesion forces

The excreted EPS mass from the HET particles can be employed as a parameter of adhesion force models between the particles. The EPS link between the particles are treated as much more stiffer springs, but only employing the attractive forces. Total effective EPS mass is calculated between the microbes (M_{eps_e}) and a spring stiffness is defined per unit mass (k_e). Effective overlap between the EPS links are calculated and the forces calculated according to the effective spring stiffness ($M_{eps_e} * k_e$) multiplied by the overlap. This approach was first applied by Head et al. to reproduce a mechanically stable biofilm.

2.5 Physical models

Present code employs physical modelling of 3 key aspects and coupling inbetween them: Microbes, fluid and nutrients. Microbes are treated as spherical motile living organisms interacting with themselves and with the ambient fluid. Physics of the microbial rheology is solved by using discrete element modelling (DEM). Fluid momentum and continuity equations are coupled with the particle solvers and solved together. Passive scalar transport equations of nutrient transport is solved along with the fluid equations. The field operational library OpenFOAM is used to construct upon a fluid solver, which is based on an existing TFM multiphase model there within [Rusche, 2003]. The Molecular Dynamics (MD) code LAMMPS is used to solve the DEM equations [Plimpton, 1995]. Some of the key features of the hybrid DEM-CFD code include:

- Open-source C++ based libraries providing an open architecture are coupled together using MPI libraries.
- BubbleFOAM, a TFM model based in OpenFOAM library [Rusche, 2003], is modified to couple with the DEM equations. The solver employs the finite volume method for discretization. The momentum transfer terms are treated implicitly leading to an improved numerical stability.

- Passive scalar transport is solved with the fluid solver using Scalar-TransportFOAM.
- Unstructured meshes and complex geometries are within the capabilities of the OpenFOAM environment but not so straightforward in LAMMPS.
- A soft sphere approach employing various visco-elastic contact models is used to resolve particle collisions with a highly efficient DEM solver.

2.5.1 Fluid equations

The fluid velocity at each point in space is replaced by its average, taken over a spatial domain large enough to contain many particles but still small compared to the whole region occupied by the flowing mixture. The locally averaged incompressible continuity and momentum equations for the gas phase are given by [Anderson and Jackson, 1967; Jackson, 1997]:

$$\frac{\partial \epsilon}{\partial t} + \nabla \cdot (\epsilon \mathbf{u}_f) = 0, \quad (6)$$

and

$$\rho_f \frac{\partial \epsilon \mathbf{u}_f}{\partial t} + \rho_f \nabla \cdot (\epsilon \mathbf{u}_f \mathbf{u}_f) = -\nabla P + \nabla \cdot \tau_f + \epsilon \rho_f \mathbf{g} - \mathbf{I}_f, \quad (7)$$

where ϵ is porosity, \mathbf{u}_f , P , ρ_f and τ_f are the fluid velocity, pressure, density and viscous stress tensor, respectively; \mathbf{g} is the gravitational acceleration; and \mathbf{I}_f is the inter-phase momentum transfer term arising due to fluid-particle interactions.

The deviatoric stress tensor τ_f include the viscous stresses also. The porosity term ϵ is calculated from the microbial positions supplied by the DEM solver. An exact procedure for the porosity calculations will be presented at a later section. The fluid momentum equations are discretized and solved on an Eulerian grid by a finite volume method. The PISO solution procedure is followed to solve the fluid momentum equation and can be summarized as follows:

- Momentum prediction: solve the equation using the pressure field and the external forcing (like the interphase momentum term), available from the previous time step
- Pressure solution: solve the pressure equation, and update the pressure field.
- Velocity correction: correct the velocity field using the updated pressure field.
- Repeat step 2 and 3 until convergence.

The inter-phase momentum equation terms are treated semi-implicitly to improve numerical stability (details can be found at Xiao and Sun [2011]). The convection and diffusion terms are discretized with a blend of central difference (with second-order accuracy) and upwind difference (with first order accuracy), respectively. An implicit first-order Euler scheme is used for the time integration of the momentum equation.

2.5.2 Discrete element method

The Newtonian equations of motion are solved for each particle in a Lagrangian framework [Cundall and Strack, 1979]. The equations for the translational and rotational movement given by:

$$m_i \frac{d}{dt} \mathbf{v}_i = \mathbf{f}_{ci} + \mathbf{f}_{fpi} + m_i \mathbf{g} + \mathbf{f}_{vdwi} \quad (8)$$

The total force acting on a microbe is calculated as the sum of total contact, adhesion, gravitational and fluid interaction forces. The contact force (\mathbf{f}_c) is calculated as a sum of all the forces due to interaction with neighbouring microbes. The adhesive force (\mathbf{f}_{adh}) is calculated as a sum of all pair-wise adhesive interaction due to the EPS. The fluid-particle interaction force (\mathbf{f}_{fp}) is obtained from a drag model, which is based on coarse grained fluid and particle variables.

Following steps are enlisted to solve the dynamics of microbial motion by the DEM equations:

- Neighbour list building: Identify the neighbouring microbes that are in actual contact or in the vicinity of the EPS sphere of influence, using the microbial positions information.
- Contact Model: Calculate the forces due to the inter-microbial collisions and the energy dissipation due to the frictional and viscous damping.
- Adhesion Model: Calculate the adhesive forces resulting from EPS interactions.
- Time-step calculations: Estimate the microbial interaction time-step based on the contact model employed and its parameters.
- Drag Model: Calculate the fluid-particle interaction force from the coarse-grained quantities. These calculations are done at the fluid time step which is much greater than the microbial interaction time step.
- Time integration: Evolve the microbial positions and velocities to the next time-step using the forces calculated from above steps.

These steps would be briefly discussed in following subsections. A detailed explanation and implementation of each step can be found at Plimpton [2005].

2.5.3 Neighbour lists

DEM time-step is typically of the order of 10^{-7} s. In order to calculate the forces by contact models on each microbes, one has to loop over each particle twice per time step. This makes the DEM computations very expensive. The computational load can be reduced by maintaining neighbour lists to keep a track of only the neighbouring particles of each particles in the system. Efforts are done to minimize the neighbour list size without omitting any force pair: contact or cohesion. In literature, three such algorithms are employed to the build neighbour lists:

- Verlet Neighbour List (VNL): A maximum search radius is supplied as an input parameter, which is more than the cohesion influence of sphere. This radius is decided by: (1) the dynamics of the system (2) the particle time-step. A lower search radius can decrease computational times drastically, but there is a risk of omitting some force pairs. A conservative search radius would be around 2-2.5 times diameter of the average particle size in the domain. It should be noted that the neighbour list is updated only so often and not at every particle time step.
- Link Cells: The simulation domain is divided into a number of sub-domains with a bin size around 2-3 diameter of particle. A list of all the particles is maintained for every sub-domain and the contacts are only checked within the sub-domain. Computational costs of this method is slightly more than the VNL method due to extra costs associated with keeping a track of all the contacts across the boundaries. However, this algorithm can be employed more efficiently if the code is parallelized. LAMMPS employs a link cell algorithm for contact detection.
- Lattices: Lattice points are defined within the domain. Each particle is indexed in relation to a lattice point and a neighbour list is created for all cells within a particle diameter d .

2.5.4 Collision model

Contact models are used to determine forces on the particles based on the particle's colliding velocities and material properties. Consider two contacting particles (i, j), with radii (a_i, a_j), at positions ($\mathbf{r}_i, \mathbf{r}_j$), with velocities ($\mathbf{v}_i, \mathbf{v}_j$) and angular velocities ω_i, ω_j .

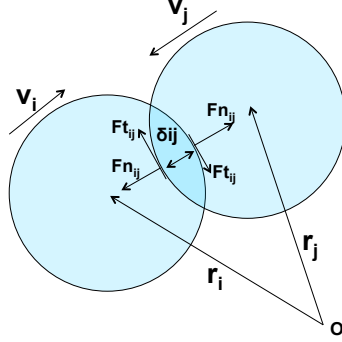


Figure 6: Schematic of two particles i and j in contact and position vectors \mathbf{r}_i , \mathbf{r}_j , respectively, with overlap δ_{ij} . Resultant normal and tangential forces acting on each particle are shown here.

The normal compression δ_{ij} , relative normal velocity $\mathbf{v}_{n_{ij}}$, and relative tangential velocity $\mathbf{v}_{t_{ij}}$ can be given as follows:

$$\delta_{ij} = d - r_{ij}, \quad (9)$$

$$\mathbf{v}_{n_{ij}} = (\mathbf{v}_{ij} \cdot \mathbf{n}_{ij})\mathbf{n}_{ij}, \quad (10)$$

$$\mathbf{v}_{t_{ij}} = \mathbf{v}_{ij} - \mathbf{v}_{n_{ij}} - (a_i\omega_i + a_j\omega_j) \times \mathbf{n}_{ij}, \quad (11)$$

where $d = a_i + a_j$, $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $\mathbf{n}_{ij} = \mathbf{r}_{ij}/r_{ij}$, with $r_{ij} = |\mathbf{r}_{ij}|$ and $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$. The rate of change of the elastic tangential displacement $\mathbf{u}_{t_{ij}}$, set to zero at the initiation of a contact is:

$$\frac{d\mathbf{u}_{t_{ij}}}{dt} = \mathbf{v}_{t_{ij}} - \frac{(\mathbf{u}_{t_{ij}} \cdot \mathbf{v}_{ij})\mathbf{r}_{ij}}{r_{ij}^2}. \quad (12)$$

The last term in equation (12) arise from the rigid body rotation around the contact point and ensures that $\mathbf{u}_{t_{ij}}$ always lies in the local tangent plane of contact. Popular contact models that are employed in the literature are a combination of spring, slider and a dash-pot. This assembly is used to estimate the normal and tangential contact forces [Cundall and Strack, 1979]. The capability of the model to capture collisional behaviour depends upon the force-displacement $(\mathbf{F}_{n_{ij}} - \delta_{ij})$ relationships. A brief listing of different contact models is provided here. For a more comprehensive details on each model, readers are referred to LAMMPS user manual [Plimpton, 2005].

- Cohesion-less normal contact models: Common examples of these are linear spring dash-pot (first used by Cundall and Strack [1979]). Hertz-midillin contact model [Tsuji *et al.*, 1992], hysteretic spring models that dissipates energy [Walton and Braun, 1986] and models accounting for plastic deformations [Thornton, 1997].

- Elastic adhesive contact models: These model include effect of adhesive forces to simulate fine powders. Commonly used models are JKR [Johnson *et al.*, 1971], DMT [Derjaguin *et al.*, 1980] and models accounting for both adhesion and plasticity [Thornton and Ning, 1998],[Luding, 2008],[Thakur *et al.*, 2014].

In the present work, a linear spring-dashpot model is employed for the contact force model with the static friction between the microbes, modelled according to the Coulomb's law (given below).

$$\mathbf{F}_{n_{ij}} = f(\delta_{ij}/d)(k_n \delta_{ij} \mathbf{n}_{ij} - \gamma_n m_{\text{eff}} \mathbf{v}_{n_{ij}}), \quad (13)$$

$$\mathbf{F}_{t_{ij}} = f(\delta_{ij}/d)(-k_t \mathbf{u}_{t_{ij}} - \gamma_t m_{\text{eff}} \mathbf{v}_{t_{ij}}), \quad (14)$$

where $k_{n,t}$ and $\gamma_{n,t}$ are the spring stiffness and viscoelastic constants, respectively, and $m_{\text{eff}} = m_i m_j / (m_i + m_j)$ is the effective mass of spheres with masses m_i and m_j . The corresponding contact force on particle j is simply given by Newton's third law, i.e., $\mathbf{F}_{ji} = -\mathbf{F}_{ij}$. The function $f(\delta_{ij}/d) = 1$ is for the linear spring-dashpot model, and $f(\delta_{ij}/d) = \sqrt{\delta_{ij}/d}$ is for Hertzian contacts with viscoelastic damping between spheres.

2.5.5 Interphase momentum transfer

The coupling between the gas phase and particle motion is implemented through a volume averaged fluid-particle interaction term (\mathbf{I}_f) in the gas momentum equation. According to the Newton's third law, negative of a similar term is added to the particle equation of motion (\mathbf{f}_{fpi}):

$$\mathbf{f}_{fpi} = -V_{pi} \nabla p + \frac{\beta_i V_{pi}}{\phi} (\mathbf{u}_{fi} - \mathbf{v}_i), \quad (15)$$

Here V_{pi} is the volume of particle i ; $\phi = (1 - \epsilon)$ is the solid volume fraction; \mathbf{u}_{fi} is the fluid velocity extrapolated to the particle i position and β_i is the particle based drag coefficient. The total interaction \mathbf{I}_f in a fluid cell is calculated by adding all particle-fluid interaction forces in the cell as

$$\mathbf{I}_f = \frac{1}{V_{\text{cell}}} \sum_{i=1}^n \mathbf{f}_{fpi} W_i, \quad (16)$$

where V_{cell} is the volume of the fluid cell and W is the weighting function accounting for the contribution of a particle. The weights are calculated according to a Box-car or a Gaussian function [Xiao and Sun, 2011]. The drag force is first calculated on each individual particle and then volume averaged to give it back to the fluid cell. In order to calculate β_i , many different correlations derived in limited conditions are employed in ad-hoc ways, to dynamic conditions present in a fluidized bed.

2.5.6 Drag model closures

The drag coefficient used in the Eq. 15 can be written as

$$\beta = 18\mu\phi(1 - \phi)\frac{F}{d^2}, \quad (17)$$

where F is the drag force, non-dimensionalized by the Stokes–Einstein drag force ($3\pi\mu d(\mathbf{u}_{fi} - \mathbf{v}_i)(1 - \phi)$); and μ is the fluid viscosity. This dimensionless drag force can be expressed as a function of solid fraction (ϕ) and the particle Reynolds number ($Re = \epsilon\rho_f d_i |\mathbf{u}_{fi} - \mathbf{v}_i| / \mu$), for a particle of diameter d .

2.5.7 Drag models

Drag models correlations were traditionally deduced from the experiments [Di Felice, 1994; Ergun, 1952; Wen and Yu, 1966]:

- Measuring pressure drop across a fixed bed [Ergun, 1952], fluidized with different inlet fluid velocities. The pressure drop was related to the particle and fluid properties along with the operational conditions.
- Sedimentation of small particles in a very dilute regime [Richardson and Zaki, 1954] and correlating the drag forces to the terminal velocity and voidage function (form ϵ^n).
- Wen and Yu [1966] used experimental data of Richardson and Zaki [1954] and correlated drag forces by a different function with a slightly better fit.
- Sedimentation of mono-disperse particles in a dense phase regime [Di Felice, 1994], and introducing an explicit function of voidage to take into account of restricted flow around many particles.

For a list of more drag models, readers are referred to the fluidization hand book by Yang [2003]. A combinations of these drag models also been used to cover a wide range of solid fractions and flow conditions found in a fluidized bed. The most popular of these models are by Gidaspow [1994] which combines the Ergun [1952] and the Wen and Yu [1966] drag models, given by equation 18.

$$F_{EWY}(\phi, Re) = \begin{cases} (1 + 0.15Re^{0.687})(1 - \phi)^{-3.65} & \phi \leq 0.2 \\ \frac{150}{18} \frac{\phi}{(1-\phi)^2} + \frac{1.75}{18} \frac{Re}{(1-\phi)^2} & \phi > 0.2 \end{cases}, \quad (18)$$

Modifications to the Richardson and Zaki [1954] correlations were done by Syamlal and O'Brien [1987] to get a better drag model applicable at high

solid fractions and extended to cover particle Re numbers upto 10. To account for non-spherical particles, drag coefficient expression was developed by Haider and Levenspiel Haider and Levenspiel [1989], based on the Di Felice [1994] equations:

$$F_{HL}(\phi, Re) = \left[(1 + A(Re)^B) + \frac{C}{1 + \frac{D}{Re}} \right] (1 - \phi)^{-\chi}, \quad (19)$$

where

$$\chi = 4.7 - 0.65 \exp(-(1.5 - \log_{10} Re)^2 / 2),$$

$$A = \exp(2.3288 - 6.4581\alpha + 2.4486\alpha^2),$$

$$B = 0.0964 + 0.5565\alpha,$$

$$C = \exp(4.905 - 13.8944\alpha + 18.4222\alpha^2 - 10.2599\alpha^3),$$

$$D = \exp(1.4681 + 12.2584\alpha - 20.7322\alpha^2 + 15.8855\alpha^3),$$

α is the particle sphericity which is defined as the ratio of the surface areas between a spherical particle with an equivalent volume and the irregular particle.

2.5.8 Algorithm for inter-phase momentum exchange

The overall algorithm for fluid-particle interaction used in the open-source DEM-CFD code can be summed up in the following way manner:

1. Calculate the inter-phase momentum term from the previous time step velocity fields (u_f^{n-1}, P_f^{n-1}) .
2. Substitute the drag terms in the momentum equation and solve iteratively the momentum equation using PISO algorithm described earlier. Update the pressure and velocities field to current time step: (u_f^{n-1}, P_f^{n-1}) to (u_f^n, P_f^n) .
3. Calculate the inter-phase momentum term based on (u_f^n, P_f^n) .
4. Compute the fluid particle drag on each particle based on (u_f^n, P_f^n) and evolve particle position using DEM simulator.
5. Obtain the voidage (ϵ) and the coarse grained particle velocity fields via averaging from DEM particle positions and velocity.
6. Give these field back to the fluid momentum equation as an input for (n+1) time step.

2.5.9 Numerical methods

DEM-CFD code works on two time scales: (1) inter-particle (2) fluid-particle interaction. Inter-particle contact time for the collisions is estimated from the contact force model employed. For a linear spring dash-pot model, a maximum contact time can be calculated in terms of spring coefficient, effective mass of particles, and the damping coefficient for collision between the particles m and n (equation 20). The time step for DEM integration is typically kept fairly low, around 2% of the contact time. For fluid-particle interaction, the particle relaxation times τ_p is used to quantify the time scales of the interactions. For a stationary particle released in a fluid, relaxation times can be calculated with a constant slip velocity assumption (defined as $u_{ri} = u_{pi} - u_{fi}$) by the equation 21.

$$t_{mn}^{col} = \pi \left(\frac{k_{mn}}{m_e} - \frac{\eta_{mn}^2}{4m_e} \right)^{-1/2} \quad (20)$$

$$\tau_p = \frac{U_e m_i}{\epsilon_f f_{di}} \quad (21)$$

where ϵ_f is the void fraction; f_{di} is the drag force acting on particle; U_e is the slip velocity and m_i is the mass of the particle i . Another important consideration for the fluid time step is the Courant-Friedrichs-Lewy (CFL) conditions based upon the mesh size employed [Xiao and Sun, 2011]. Particle time steps are decided by the minimum of the two time scales, whereas fluid time step is decided by the CFL conditions. Furthermore, turbulent length scales can be important especially in case of high Re number and lean phase cases such as pneumatic conveying Sommerfeld and Kussin [2003]. τ_p and t_{mn}^{col} governs the fluid and particle time steps. For gas-solid interactions, the particle time steps are 2-3 order of magnitude more than the fluid time steps.

2.6 Stochastic processes: Scope of improvement

Different processes are modelled by the sub-models as stochastic processes. This adds variability and randomness to the biological process, typically evident in the nature. Following processes are random:

- Initialization of the system by placing the microbes with random physical and biological attributes are placed at random locations within the domain. This introduces the randomness of the initial configuration.
- Cell division criterion is based on a critical random maximum ratio, which can be selected by the user. Solver further divides the daughter cell masses randomly in a ratio between 0.4-0.6 placing daughter cells in a random orientation.

- During the death sub-model, microbes equivalent to the dead-mass are randomly chosen by the solver.
- EPS excretion occurs after the shell bound EPS exceeds a randomly selected value by the user. This excreted EPS is oriented and placed randomly around the mother cell.

As recommended by the iDynamics solver, these processes can be tackled from a microscopic point of view with fundamental chemical and biological processes driving these mechanisms.

3 LAMMPS

3.1 Introduction to LAMMPS

LAMMPS is a classical molecular dynamics code developed at Sandia labs and primarily built to solve the particle physics including wide range of inter-particle interactions and potentials. The code treats each particle as an individual discrete unit, much similar to the popular IB approach. Sandia Labs distributes LAMMPS under the terms of the GNU Public License (<http://lammps.sandia.gov/>). The current version of the code is written in C++ with an open architecture and provides an opportunity to couple with other open-source codes. LAMMPS can run efficiently in both serial and parallel versions depending upon the computational facilities available to the users. The LAMMPS code is designed to modify and extend it with newer capabilities as desired by the user. While only 25% of the 140K line code in LAMMPS forms the core of the solver, rest of the code is contributed by a large user database across the globe in order to extend its capabilities. An overview can of current LAMMPS capabilities can be found at LAMMPS-feature.

3.2 LAMMPS working methodology

LAMMPS solves the motion of every single particle by simply integrating Newton's equations of motion in response to sum of the forces (short or long range based on their interaction with neighbours). At a particular time instance, motion of each particle is collectively solved when subjected to initial or boundary conditions. In order to maintain computational tractability while calculating the interaction forces, LAMMPS maintains a neighbourhood list for each particle which gets updated every so often. These lists are optimized so that local densities and particle overlaps never becomes non-physical. For parallel simulations, LAMMPS spatially partition the domain into smaller sub-domains assigned to each processors. Interprocessor communications are maintained by storing ghost atom interactions with the

sub-domain boundaries. LAMMPS development can be helped by two user manuals: User manual and developer manual. The following links will be helpful for the users to get started on LAMMPS:

1. User manual: <http://lammps.sandia.gov/doc/Manual.pdf>
2. Developers guide: <http://lammps.sandia.gov/doc/Developer.pdf>
3. Tutorials: <http://lammps.sandia.gov/tutorials.html>
4. Commands: http://lammps.sandia.gov/doc/Section_commands.html
5. Features: <http://lammps.sandia.gov/features.html>

In the present study, LAMMPS-Feb14 version is developed and newer IB features and capabilities added, this version will be now on referred as LAMMPS-NCL.

3.3 Operating systems

In general, LAMMPS can be run on Windows, Linux, Mac OS using pre-built executables. LAMMPS-NCL can be compiled with almost any linux or Mac OS (instructions in the user manual). It is emphasized that present NCL version has been rigorously tested on Ubuntu-14.10 and Fedora-22. In near future, prebuilt executables, binaries or RPMS will be provided to be used on any OS.

3.4 Pre-compilation instructions

Before compiling LAMMPS, please make sure you are installed with these packages depending upon the operating system used:

- fftw (<http://www.fftw.org/doc/Installation-on-Unix.html>)
- openmpi (<http://www.open-mpi.org/>)
- libjpeg (<http://libjpeg-turbo.virtualgl.org/>)
- gcc/g++ (<https://help.ubuntu.com/community/InstallingCompilers>)

4 LAMMPS-NCL and getting started

For the pre-existing LAMMPS commands, features and documentation, please refer to the LAMMPS user manual, listed above. The manual covers extensive instructions on compiling LAMMPS and how to get started. LAMMPS-NCL is compiled the same way as you would compile LAMMPS and there is

no change in those instructions. The newer capabilities and commands will be highlighted and emphasized in the next sections and the sample input script.

4.1 Downloading LAMMPS-NCL

In general, different versions of LAMMPS can be downloaded from the download section of the LAMMPS website. For the present study, LAMMPS version Feb 2014 was developed and enhanced with biological IB model capabilities. It is advised to use the version developed at NCL. It can be downloaded using following set of instructions from the NUFEB Github repository.

4.2 LAMMPS extension

In the present study, capabilities of LAMMPS have been extended to include cell level biological processes such as cell growth, division, maintenance, death etc. These extended capabilities involve addition of newer particles at each time-steps e.g. during a cell division: a single parent cell divide into two daughter cells. A summary of list of newer capabilities added to the LAMMPS and its implementation is presented here:

4.3 Atom type

Classical LAMMPS provide different atom types that could be used by user. These are specified in the input script by the command: `atom_style`. `atom - style` command must be used before a simulation is setup via a `read-data`, `read-restart` or `create-box` command. A newer `atom_style` is added (named "bio") to increase the number of attributes such as nutrients available to each microbe type. The new `atom_style` is inherited from already existing `atom - style` sphere. Though LAMMPS provides another route for adding newer attributes through `fix property/atom`. However, since the newer attributes would be used to couple with the fluid solver, they should be more tightly linked to the atom style.

The new atom type "Bio" have following attributes in conjunction to the sphere type: ID, type, radius, outer-radius, mass, outer-mass,x,y,z, S_s , O_2 , NO_3 , NO_2 , NH_4 . As a general rules following types are assigned to the functional groups.

- 1-HET
- 2-AOB
- 3-NOB

- 4-EPS
- 5-Inert

Each microbe has unique moniker ID but share its type according to the functional group. Physical parameters such as initial radius, outer radius, density and outer density can be mentioned. Radius and density are the radius and density of the microbe. Outer radius and densities are the initial EPS-shell bound radius and densities. Density is constant throughout the simulation and time invariant. However radius and mass (calculated from density and radius) can change. EPS is only bound to the HET particles (type 1), hence outer radius should be specified as radius for other particle types initially. LAMMPS will give an error message if this is not followed. For LAMMPS book-keeping, inner radius is always same as outer-radius for other types than 1 throughout the simulation. Initial configuration can be either generated randomly through LAMMPS or by the user. Caution must be taken to avoid overlaps when defined by the user. A typical user input script is included in the folder (IC5nut.in).

Different sub-models are coded within LAMMPS in a modular way. Each of these models, implementations and command lines are explained in the following sub-sections.

5 Input script generated for the NUFEB-Bio simulations

In order to execute LAMMPS commands, an input script is usually prepared with certain sub-commands and parameters list. Figure 7 gives the input script for the NUFEB project. This script can be generated by GUI and will be explained later. New additions to the LAMMPS commands are pointed in the boxes as shown in the figure.

```

atom_style      bio
atom_modify     map array sort 10000 1e-5
boundary        pp pp pp
neutron         off
communicate     single yes
read_data       local in
group HET type 1 group AOB type 2 group NOB type 3 group EPS type 4 group inert type 5
neigh_modify     1.0e-5 bin neigh_modify delay 0

pair_style      gran/hooke/history 200000000 NULL 15000000 NULL 0.5 1
pair_coeff       * *
timestep        1
velocity        all set 0.0 0.0 0.0 units box
fix             1 all mve/sphere
fix             2 all gravity 9.8 vector 0 -1 0 # spherical 90.0 -100.0
fix             3 all fdnrg

variable KshET equal 0.01 variable Ko2HET equal 0.01 variable Knc2HET equal 0.0003 variable Knc3HET equal 0.0003 variable Knh4AOB equal 0.001 variable Ko2AOB equal 0.0005
variable Knc2NOB equal 0.0013 variable Ko2NOB equal 0.00068
variable MshET equal 0.0000644444 variable MshAOB equal 0.0000347222 variable MshNOB equal 0.0000347222 variable etshET equal 0.6
variable bshET equal 0.00000462962 variable bshAOB equal 0.0000017314 variable bshNOB equal 0.0000017314 variable bEPS equal 0.00000196759
variable YEPS equal 0.10 variable YHET equal 0.61 variable EPSdens equal 30 variable EPSratio equal 1.25 variable factor equal 2.0

fix g1 all nugrowth 1 v_KshET v_Ko2HET v_Knc2HET v_Knc3HET v_Knh4AOB v_Ko2AOB v_Knc2NOB v_Ko2NOB v_MshET v_MshAOB v_MshNOB v_etshET v_bEPS v_YEPS v_YHET v_EPSdens
fix dt1 HET death 1 v_bHET v_factor 1562467
fix dt2 AOB death 1 v_bAOB v_factor 1234312
fix dt3 NOB death 1 v_bNOB v_factor 1335362
fix dt4 all divide 1 v_EPSdens 2.0 1242242
fix et HET eps_extract 1 v_EPSratio v_EPSdens 1242242

dump            id1 HET custom 2000 HET.bubblelend radius outerradius
#dump           id2 HET custom 173000 HETdeath.bubblelend type radius outerradius mass outermass
#dump           id3 EPS custom 1000 snapshot2.bubblelend id type diameter mass v_x v_y v_z
dump            id all custom 2000 snapshot.bubblelend id type radius vx vy vz x y z outerradius outermass

thermo_style     custom step atoms ke vol
#thermo_style    granular # not work. syntax change?
#thermo_style    one # granular does not work
thermo           1
thermo_modify     lost error
#restart          1000000 restart.*.bubblelend

run 172800

```

Figure 7: Input script for NUFEB simulation of biofilm dynamics.

5.1 Cell growth

Cell growth model described in the section 2.4.1 is implemented as a fix type "growth" in the LAMMPS, the source code can be found in the src folder of the LAMMPS directory. In the present implementation, GUI will help to generate LAMMPS command line for the fix growth and user just have to make sure to put in correct parameters. GUI is described later. Figure 8 gives the LAMMPS command line for the growth model.

```

fix g1 all nugrowth 1 v_KshET v_Ko2HET v_Knc2HET v_Knc3HET v_Knh4AOB v_Ko2AOB v_Knc2NOB v_Ko2NOB v_MshET v_MshAOB v_MshNOB v_etshET v_bEPS v_YEPS v_YHET v_EPSdens

```

Figure 8: Command line for the growth fix in LAMMPS

In total, 20 parameters are given for the growth fix command. Specifically, "fix g1 all nugrowth 1" means invoke a fix nugrowth instance named g1 and apply it at every timestep (1 = frequency of call) to "all" the particle types in the assembly. The second parameter "all" indicates that all the microbes can grow. Any other microbial type could also be mentioned, which would indicate that growth only occurs only for that specific type. All of the parameters has to be in SI units. A table of the typical parameters can be seen at the end of the documentation. The parameter description:

- fixID: unique ID of the fix (not used in the script before)

- group: Functional groups on which fix will be invoked, it can take: HET, AOB, NOB, all.
- fix-name: nugrowth (fix name). It can not take another value.
- freq: frequency at which the fix is invoked (typically at every timestep). Values has to be integers.
- KsHET Ko2HET Kno2HET Kno3HET Knh4AOB Ko2AOB Kno2NOB Ko2NOB: Affinity of all the functional groups to the substrate.
- MumHET MumAOB MumNOB: Monod kinetic parameters for the HET, AOB and NOB.
- etaHET: Reduction factor in anoxic conditions for HET.
- bEPS: Decay rate of EPS.
- YEPS, YHET: Yield coefficient for EPS and HET particles.
- EPS-density: Density of the EPS, required for the divisions of HET particles. Values can be floating type number.

5.2 Cell Division

Cell division model described in the section 2.4.2 is implemented as a fix type "division" in the LAMMPS, the source code can be found in the src folder of the LAMMPS directory. The command line is given in the figure 9:

```
fix d1 all divide 1 v_EPSdens 2.0 1242242
```

Figure 9: Command line for the division fix in LAMMPS

The command line follows the template: "fix fixID group fix-name freq EPS-density Ratio seed". The parameter description:

- fixID: unique ID of the fix (not used in the script before)
- group: Functional groups on which fix will be invoked, it can take: HET, AOB, NOB, all.
- fix-name: divide (fix name). It can not take another value.
- freq: frequency at which the fix is invoked (typically at every timestep). Values has to be integers.

- EPS-density: Density of the EPS, required for the divisions of HET particles. Values can be floating type number.
- Ratio: User defined diameter ratio at which the division takes place. The ratio is calculated as the instantaneous diameter divided by the average diameter of the microbes. The average diameter is initially defined and hard coded. The values have been indicated in the earlier sections. Values typically floating type number greater than 1.0.
- Seed: Seed for the orientation and introduce the randomness, very large integer values. Same seed will result in the same orientation for every simulation run.

5.3 Cell Death

Cell death model described in the section 2.4.3 is implemented as a fix type "death" in the LAMMPS, the source code can be found in the src folder of the LAMMPS directory. Figure 10 gives the specific command lines.

```
fix dt1 HET death 1 v_bHET v_factor 1952467
fix dt2 AOB death 1 v_bAOB v_factor 1234312
fix dt3 NOB death 1 v_bNOB v_factor 1325352
```

Figure 10: Command line for the death fix in LAMMPS

The command line follows the template: "fix fix-ID group fix-name freq decay-rate ratio seed". The parameter description:

- fixID: unique ID of the fix (not used in the script before)
- group: Functional groups on which fix will be invoked, it can take: HET, AOB, NOB, all.
- fix-name: death (fix name). It can not take another value.
- freq: frequency at which the fix is invoked (typically at every timestep). Values has to be integers.
- decay-rate: Rate of decay, according to the type of microbe.
- Ratio: Death ratio, described earlier in the model, typically floating number greater than 1.0.
- Seed: Seed for randomly choosing particles to kill.

5.4 EPS extract

EPS extraction model described in the section 2.4.4 implemented as a fix type "EPS extract" in the LAMMPS, the source code can be found in the src folder of the LAMMPS directory. Figure 11 gives the specific command lines.

```
| fix e1 HET eps_extract 1 v_EPSratio v_EPSdens 1242242
```

Figure 11: Command line for the EPS extract fix in LAMMPS

The command line follows the template: "fix fix-ID group epsextract freq EPSratio EPSdens seed". The parameter description:

- fixID: unique ID of the fix (not used in the script before)
- group: Functional groups on which fix will be invoked, it can take: HET, AOB, NOB, all.
- fix-name: death (fix name). It can not take another value.
- freq: frequency at which the fix is invoked (typically at every timestep). Values has to be integers.
- EPS ratio: Ratio between outer-radius and inner-radius of the microbe. Value is typically floating number greater than 1.0.
- EPS-density: Density of the EPS, required for the divisions of HET particles. Values can be floating type number.
- Seed: Seed for the orientation and introduce the randomness, very large integer values. Same seed will result in the same orientation for every simulation run.

6 Shoving algorithms

Implementation of IB submodels often result in overlaps between the microbes. Resolution of these overlaps is necessary because of 2 reasons 1) Physical sub-models dealing with particle level physics are often based on the soft particle dynamics with minimum overlaps. 2) In natural conditions existing in microbial colonies, microbes would not overlap with each other. Previously shoving algorithms have been proposed in literature, iDynamics employ a relaxation algorithm that determines the steady-state microbial locations with the minimal number of overlaps. Pair-wise overlap is resolved by moving each agent by half the overlap distance. This approach can be computationally expensive and might require several iterations to resolve the overlaps. Present solver attempts to solve physics at the particle level,

which require forces to be calculated based on instantaneous particle overlaps. Force equation is solved on each particle and assembly is evolved using Newton's laws of motion. LAMMPS solves the force equations and method as earlier discussed is called Discrete Element Method (DEM). The method works with soft particle force models (system modelled as particles linked with spring-dashpot system) where particles are allowed small overlaps (typically less than 1% of their sizes). A pair-wise repulsive force is exerted on the particles as a function of the overlap and the relative velocities, resolving particle collisions. This dynamics can be controlled by using smaller particle time-steps if un-physically high overlaps are seen in the system. Introduction of IB sub-models can result in increasingly high particle overlaps, hence needing an algorithm to resolve the high overlaps without giving system un-physically high forces and velocities.

1. Microbial growth: Microbes are represented as spherical masses with finite diameter and volume. As the microbial mass is added, volume and diameter of the particle increases. Overlap between the particles is calculated as the difference between the sum of radii and distance between their centres. Since the radii are increasing, an extra compensation has to be made to the particle centre coordinates in order to avoid high overlaps. Particle growth rate is usually much smaller than the collision times, hence overlaps due to particle growth models are expected to resolve by themselves.
2. Microbial division: As the mother cells divides into 2 cells, orientation and the placement of these daughter cells can result into sudden and large overlaps. This might result in bad dynamics in the system. One of the ways is to check for the overlaps before inserting the cell into the position. However, re-building neighbour lists before inserting each particle can be computationally very expensive.
3. EPS extraction: Similar to problem with the microbial division, placing extracted EPS particle at an appropriate location or resolving resulting large overlaps can be problematic.

Apart from reducing timesteps, LAMMPS provides several options to resolve these overlaps. Any of these commands or LAMMPS functionality can be used. At present, test runs are being conducted to check the parameters and the optimization of the run time. These LAMMPS options are listed here:

- Add a viscous damping force to atoms in the group that is proportional to the velocity of the atom. The added force can be thought of as a frictional interaction with implicit solvent, i.e. the no-slip Stokes drag on a spherical particle. This can be useful for draining the kinetic energy

from the system in a controlled fashion. If used without additional thermostating (to add kinetic energy to the system), it has the effect of slowly (or rapidly) freezing the system; hence it can also be used as a simple energy minimization technique. The damping force F is given by $F = -\gamma * \text{velocity}$. The larger the coefficient, the faster the kinetic energy is reduced. A full description can be found [here](#).

- Perform a constant NVE updates of position and velocity for atoms in the group each timestep. A limit is imposed on the maximum distance an atom can move per timestep. Normally large overlaps would generate huge forces which would blow atoms out of the simulation box, causing LAMMPS to stop with an error. This can be done by using limited time integrator, full description can be found [here](#).
- Using a pair-wise repulsive forces that would not blow up if particle overlaps. Description and implementation can be found [here](#).
- Energy minimization: Perform an energy minimization of the system, by iteratively adjusting atom coordinates. Iterations are terminated when one of the stopping criteria is satisfied. At that point the configuration will hopefully be in local potential energy minimum. More precisely, the configuration should approximate a critical point for the objective function, which may or may not be a local minimum. This method allows to adjust the particle coordinates to minimize the energy, instead of completely resolving the overlaps as is the case with iDynamics. Description and implementation can be found [here](#).

7 Graphical User interface (GUI)

Graphical user interface can be used to run the test case in place of the LAMMPS input script. However, it is strongly recommended to go through LAMMPS tutorials and user manuals before using the GUI. GUI summarizes the user input commands that are mostly required by the user. Some of the LAMMPS commands which are essential to run every case are not a part of GUI. GUI generates a LAMMPS input script based on user set options and this script can be checked by the user and run accordingly. GUI source code (written using wxwidgets) and compilation instructions can be found at [Git-hub repository](#). A snapshot of the GUI is shown in the [figure 12](#).



Figure 12: Graphical User Interface (GUI) for NUFEB model

8 Visualization and post-processing

LAMMPS can output snapshot file which gives information on the particle ids, positions, diameter etc at each timestep frequency as required by the user. Visualization tools often used in conjunction with LAMMPS are can be found here. POVray is more preferred visualization tool by the NUFEB project, the tool and compilation instructions can be found here. Post processing tools to convert the LAMMPS format to POVray format can be found in the pizza python toolkits found here. Scripts for post-processing of the LAMMPS data are under preparation.

References

Anderson, T.B. and Jackson, R. (1967). Fluid mechanical description of fluidized beds. Equations of motion. *Industrial & Engineering Chemistry Fundamentals*,

- 6(4): pp. 527–539.
- Cundall, P.A. and Strack, O.D.L. (1979). A discrete numerical model for granular assemblies. *Geotechnique*, **29**(1): pp. 47–65.
- Derjaguin, B., Muller, V., and Toporov, Y. (1980). On different approaches to the contact mechanics. *Journal of Colloid and Interface Science*, **73**(1): pp. 293–294.
- Di Felice, R. (1994). The voidage function for fluid-particle interaction systems. *International Journal of Multiphase Flow*, **20**(1): pp. 153–159.
- Eberl, H., Picioreanu, C., Heijnen, J., and Van Loosdrecht, M. (2000). A three-dimensional numerical study on the correlation of spatial structure, hydrodynamic conditions, and mass transfer and conversion in biofilms. *Chemical Engineering Science*, **55**(24): pp. 6209–6222.
- Ergun, S. (1952). Fluid flow through packed columns. *Chem. Eng. Prog.*, **48**.
- Gidaspow, D. (1994). *Multiphase flow and fluidization: continuum and kinetic theory descriptions*. Academic press.
- Haider, A. and Levenspiel, O. (1989). Drag coefficient and terminal velocity of spherical and nonspherical particles. *Powder technology*, **58**(1): pp. 63–70.
- Huston, M., DeAngelis, D., and Post, W. (1988). New computer models unify ecological theory. *BioScience*: pp. 682–691.
- Jackson, R. (1997). Locally averaged equations of motion for a mixture of identical spherical particles and a Newtonian fluid. *Chemical Engineering Science*, **52**(15): pp. 2457–2469.
- Johnson, K., Kendall, K., and Roberts, A. (1971). Surface energy and the contact of elastic solids. *Proceedings of the royal society of London. A. mathematical and physical sciences*, **324**(1558): pp. 301–313.
- Kreft, J.U., Booth, G., and Wimpenny, J.W. (1998). Bacsim, a simulator for individual-based modelling of bacterial colony growth. *Microbiology*, **144**(12): pp. 3275–3287.
- Kreft, J.U., Picioreanu, C., Wimpenny, J.W., and van Loosdrecht, M.C. (2001). Individual-based modelling of biofilms. *Microbiology*, **147**(11): pp. 2897–2912.
- Luding, S. (2008). Cohesive, frictional powders: contact models for tension. *Granular matter*, **10**(4): pp. 235–246.
- Picioreanu, C., van Loosdrecht, M.C., and Heijnen, J.J. (2000). A theoretical study on the effect of surface roughness on mass transport and transformation in biofilms. *Biotechnology and bioengineering*, **68**(4): pp. 355–369.
- Picioreanu, C., Van Loosdrecht, M.C., Heijnen, J.J., *et al.* (1998a). Mathematical modeling of biofilm structure with a hybrid differential-discrete cellular automaton approach. *Biotechnology and bioengineering*, **58**(1): pp. 101–116.

- Piciooreanu, C., van Loosdrecht, M.C., Heijnen, J.J., *et al.* (1998b). A new combined differential-discrete cellular automaton approach for biofilm modeling: application for growth in gel beads. *Biotechnology and bioengineering*, **57**(6): pp. 718–731.
- Plimpton, S. (1995). Fast parallel algorithms for short-range molecular dynamics. *Journal of Computational Physics*, **117**(1): pp. 1–19.
- Plimpton, S. (2005). LAMMPS user's manual. *Sandia National Laboratory*.
- Richardson, J. and Zaki, W. (1954). The sedimentation of a suspension of uniform spheres under conditions of viscous flow. *Chemical Engineering Science*, **3**(2): pp. 65–73.
- Rusche, H. (2003). *Computational fluid dynamics of dispersed two-phase flows at high phase fractions*. Ph.D. thesis, Imperial College London (University of London).
- Sommerfeld, M. and Kussin, J. (2003). Analysis of collision effects for turbulent gas-particle flow in a horizontal channel. Part II. Integral properties and validation. *International journal of multiphase flow*, **29**(4): pp. 701–718.
- Syamlal, M. and O'Brien, T.J. (1987). A generalized drag correlation for multiparticle systems. *Unpublished report*.
- Thakur, S.C., Morrissey, J.P., Sun, J., Chen, J., and Ooi, J.Y. (2014). Micromechanical analysis of cohesive granular materials using the discrete element method with an adhesive elasto-plastic contact model. *Granular Matter*, **16**(3): pp. 383–400.
- Thornton, C. (1997). Coefficient of restitution for collinear collisions of elastic-perfectly plastic spheres. *Journal of Applied Mechanics*, **64**(2): pp. 383–386.
- Thornton, C. and Ning, Z. (1998). A theoretical model for the stick/bounce behaviour of adhesive, elastic-plastic spheres. *Powder technology*, **99**(2): pp. 154–162.
- Tsuji, Y., Tanaka, T., and Ishida, T. (1992). Lagrangian numerical simulation of plug flow of cohesionless particles in a horizontal pipe. *Powder technology*, **71**(3): pp. 239–250.
- Walton, O.R. and Braun, R.L. (1986). Viscosity, granular-temperature, and stress calculations for shearing assemblies of inelastic, frictional disks. *Journal of Rheology (1978-present)*, **30**(5): pp. 949–980.
- Wen, C.Y. and Yu, Y.H. (1966). A generalized method for predicting the minimum fluidization velocity. *AIChE Journal*, **12**(3): pp. 610–612.
- Xiao, H. and Sun, J. (2011). Algorithms in a Robust Hybrid CFD-DEM Solver for Particle-Laden Flows. *Communications in Computational Physics*, **9**(2): p. 297.
- Yang, W.c. (2003). *Handbook of fluidization and fluid-particle systems*. CRC Press.