Powder Technology 235 (2013) 33–41

Contents lists available at SciVerse ScienceDirect

Powder Technology

journal homepage: www.elsevier.com/locate/powtec

Development of parallel DEM for the open source code MFIX

Pradeep Gopalakrishnan [a,b,*], Danesh Tafti [b]

[a] *National Energy Technology Laboratory, Morgantown WV, United States*
[b] *Viginia Tech, Blacksburg, VA, United States*

ARTICLE INFO

Article history:
Received 29 December 2011
Received in revised form 1 June 2012
Accepted 7 September 2012
Available online 15 September 2012

Keywords:
MFIX
Discrete Element Method
Parallelization
Fluidized beds

ABSTRACT

The paper presents the development of a parallel Discrete Element Method (DEM) solver for the open source code, Multiphase Flow with Interphase eXchange (MFIX) based on the domain decomposition method. The performance of the code was evaluated by simulating a bubbling fluidized bed with 2.5 million particles. The DEM solver shows strong scalability up to 256 processors with an efficiency of 81%. Further, to analyze weak scaling, the static height of the fluidized bed was increased to hold 5 and 10 million particles. The results show that global communication cost increases with problem size while the computational cost remains constant. Further, the effects of static bed height on the bubble hydrodynamics and mixing characteristics are analyzed.

1. Introduction

Coupled Eulerian–Lagrangian simulation of multiphase flow has been recognized as a useful technique to elucidate the fundamentals of dense particulate flows that occur in various industrial applications. In this approach, the fluid is treated as a continuum while the solid phase is modeled using the Discrete Element Method (DEM). In the DEM, particle–particle and the particle–wall interactions are resolved and the time integration is carried out using Newton's second law of motion. The computational expense of the DEM is very high owing to the expensive contact detection algorithm, and solid time step limitations to resolve particle interactions via collisions. Hence most of the previous DEM studies have been limited to small scale systems with few thousand particles and mostly to two dimensional simulations(Tsuji et al. [1], Muller et al. [2,3], Zhang et al. [4]). However, real applications at the lab as well as industrial scale are rarely 2D and the extrapolation of results from 2D to 3D introduces additional uncertainty [5]. Thus the development of an efficient parallel simulation tool which can handle large particle systems is necessary to allow routine calculations of 3D systems.

Most previous CFD/DEM parallelization efforts available in the open literature employ domain decomposition for the fluid solver, but differ on the parallelization of the discrete phase. There are three major categories available for parallelizing the DEM, namely; mirror domain method [6,7], particle-subset method [8,9], and domain-decomposition method [9,10]. The main computational cost for the DEM arises from the particle neighbor search for the collision operator, the particle contact force computation, and the communication overheads. In the case of coupled simulation, additional cost arises from the drag force computation and the communication required between the flow solver and the DEM solver.

In the mirror domain technique, each processor holds the information of all the particles in the system but carries out computations only on a subset. Synchronization of the dataset is carried out via interchange between the processors after every particle time step. Washington and Meegoda [7] divided the inter particle force computation among the processors but stored all particle information on each processors. They reported a speedup of 8.7 for 1672 particles with 512 processors. Darmana et al. [6] employed the mirror domain method and obtained speedup of 20 for the simulation of buoyancy driven flow in a bubble column with 105 bubbles using 32 processors. The disadvantages of this method are large memory requirements for holding all particle information and the high communication cost associated with the synchronization of data. In addition, for coupled simulations, the fluid variables required for the drag computation also needs to be mirrored. Due to these drawbacks, the mirror domain method could result in poor scalability for large systems.

Kaufi et al. [8] employed particle subset method where the particles are divided among the processors based on a graph partitioning algorithm. In this algorithm, a graph of particles connected by contacts is developed and the particles in close proximity are assigned to a single processor, thereby reducing the amount of particles in contact with the particles in the neighbor processor. They estimated the performance for the bubbling bed with 50,000 particles and reported a speedup of 35 for 64 processors. The main advantage of the particle-subset system is that the load balancing is ensured for both dense and dilute systems. However since particles are divided, the fluid variables required for the drag computation have to be mirrored and global summations are needed for the particle–fluid interaction

[*] Corresponding author at: Virginia Tech, Blacksburg, VA, United States.
E-mail address: pradeepg@vt.edu (P. Gopalakrishnan).

0032-5910/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.powtec.2012.09.006

terms, which could affect the scalability for large systems with large processor counts [9].

Kachunikas et al. [10] employed the standard domain decomposition for DEM and analyzed mono and polydispersed DEM. Their analysis for tri-axial compaction of materials with around 100,000 particles showed a scale up of 8.8 for 10 processors for both mono and polydisperse simulation. They reported that the computational cost associated with the particle neighbor search increases for a polydisperse system due to limitation of cell size based on large particle diameter. However the effect of polydispersity on the parallelization is not conclusive. Tsuji et al. [11] employed the domain decomposition for three dimensional analysis of shallow bubbling fluidized bed to study the effect of bed dimension on the bubble characteristics. They simulated 4.5 million particles with 16 processors. They did not provide any scaling results for their simulation.

Plimpton [9] employed three different types of parallelization; atom decomposition (fixed subset of atoms), force-decomposition (fixed subset of inter atomic forces) and spatial decomposition for molecular dynamics simulation. They concluded that the atom decomposition and the force-decomposition are not optimal for large simulations due to high communication costs. The spatial decomposition algorithm has strong scaling for large problems but suffers easily from load imbalance.

The open source code, Multiphase Flow with Interphase eXchange (MFIX) [12] developed at NETL has been widely used to simulate hydrodynamics, heat transfer and chemical reactions occurring in bubbling and circulating fluidized beds. MFIX supports both Two-Fluid Method (TFM) [13] and CFD/DEM [14] approaches. In the CFD/DEM approach, the flow is solved as a continuum similar to that of the TFM while the dispersed solids are modeled using the soft-sphere model developed by Cundall and Strack [15]. In the current released version of the MFIX, the TFM is parallelized using domain decomposition technique, while the CFD/DEM supports only serial execution. In this study, an MPI parallel implementation of the DEM is developed in compliance with the existing framework of MFIX. The study details the implementation algorithm and the results of a strong and a weak scaling analysis. Profiling is carried out with Tuning and Analysis Utilities (TAU) [16] along with Program Database Toolkit (PDT) [17] automated instrumentation.

2. Governing equations

In the coupled CFD/DEM simulation, the solid phase is represented by a finite number of spherical particles with predefined properties. The linear velocity, $\boldsymbol{v_p}$ and the angular velocity, ω_p of these discrete particles are advanced based on the forces acting on them.

$$m_p \frac{d\boldsymbol{v_p}}{dt} = \boldsymbol{f_c} + \boldsymbol{f_f} + m_p \boldsymbol{g} \tag{1}$$

$$I_p \frac{d\boldsymbol{\omega_p}}{dt} = \boldsymbol{T_c}. \tag{2}$$

Where $\boldsymbol{f_c}$ is the contact force due to particle–particle and particle–wall contacts, $\boldsymbol{f_f}$ is the particle–fluid interaction force, $\boldsymbol{g}$ is the gravity force, $\boldsymbol{T_c}$ is the torque due to tangential component of the contact force, and m_p, I_p are mass and moment of inertia, respectively. The particle–particle and particle–wall interactions are resolved using the soft sphere approach [15]. In this approach the particles are allowed to overlap and the forces are computed based on the linear spring-dashpot model. There are two linear-spring dashpot models used; one for the normal direction and another for the tangential direction. The spring force is computed based on the overlap and the spring constant, while the damping force is computed based on the relative velocity of the particles and the damping coefficient. More detailed description of the contact force computation is provided in previous MFIX-DEM studies [14,18,19]. The fluid particle interaction force is given by

$$f_f = -V_p \nabla p_g + \frac{V_p \beta}{(1-\varepsilon)} \left(\boldsymbol{v_g} - \boldsymbol{v_p} \right) \tag{3}$$

where V_p is the volume of the particle, p_g is the gas pressure, β is the friction coefficient, v_g is the velocity of gas, and ε is the void fraction. In this study, the frictional coefficient β is computed based on Gidaspow [20] which uses the Ergun [21] and Wen-Yu [22] correlation. In order to avoid discontinuity a blending factor is used [23], as follows

$$\begin{aligned}
&\beta = W\beta_{\mathrm{Ergun}} + (1-W)\beta_{\mathrm{Wen-Yu}} \\
&\beta_{\mathrm{Ergun}} = 150 \frac{(1-\varepsilon)^2 \mu_g}{\varepsilon d_p^2} + 1.75 \frac{\left| \boldsymbol{v_g} - \boldsymbol{v_p} \right| \rho_g (1-\varepsilon)}{d_p} \\
&\beta_{\mathrm{Wen-Yu}} = 0.75 C_{\mathrm{D}} \frac{\left| \boldsymbol{v_g} - \boldsymbol{v_p} \right| \varepsilon \rho_g (1-\varepsilon)}{d_p} \varepsilon^{-2.65} \\
&W = \frac{\tan^{-1}[150(\varepsilon - 0.8)]}{\pi} + 0.5
\end{aligned} \tag{4}$$

where d_p is the diameter of the particle, ρ_g is density of the gas, μ_g is the absolute viscosity of the gas and C_D is the drag coefficient, which is given by

$$C_{\mathrm{D}} = \begin{cases} \frac{24}{\mathrm{Re}} \left[1 + 0.15 \mathrm{Re}^{0.687} \right] & \mathrm{Re} < 1000 \\ 0.4 & \mathrm{Re} \geq 1000 \end{cases}. \tag{5}$$

The flow is solved using volume averaged continuity and Navier–Stokes Equation,

$$\frac{\partial}{\partial t} \left(\varepsilon \rho_g \right) + \nabla . \left(\varepsilon \rho_g \boldsymbol{v_g} \right) = 0 \tag{6}$$

$$\frac{\partial}{\partial t} \left(\varepsilon \rho_g \boldsymbol{v_g} \right) + \nabla . \left(\varepsilon \rho_g \boldsymbol{v_g} \boldsymbol{v_g} \right) = -\varepsilon \nabla p_g + \nabla . \left(\varepsilon \overline{\overline{\boldsymbol{\tau}}}_{\boldsymbol{g}} \right) + \varepsilon \rho_g \boldsymbol{g} - \boldsymbol{I_{gs}} \tag{7}$$

where $\boldsymbol{I_{gs}}$ is the interaction term representing the interphase momentum transfer between the gas and the solid system, which is given by

$$\boldsymbol{I_{gs}} = \frac{1}{V^k} \sum\nolimits_p \frac{\phi_p^k \beta V_p}{(1-\varepsilon)} \left(\boldsymbol{v_p} - \boldsymbol{v_g} \right) \tag{8}$$

where V^k is the volume of the cell k. The summation is carried over all the particles in and around the cell k. The interpolation factor ϕ determines the contribution of each particle drag to the cell, which is inversely proportional to the distance between particles and the cell center. For a particle p, the interpolation factor satisfies

$$\sum\nolimits_k \phi_p^k = 1. \tag{9}$$

The flow solver is based on the SIMPLE algorithm and more detailed information about the solver is available in the report by Syamlal et al. [13,24].

3. Algorithm

3.1. Serial algorithm

Before discussing the parallel implementation, critical steps involved in the serial CFD/DEM computation are shown in the flow chart, Fig. 1a. The flow solver is first invoked and the solution is advanced to time step, t^{n+1}. During the flow advancement, the particle locations are fixed and the interphase momentum source term, $\boldsymbol{I_{gs}}$ is computed based on the values at time level t^n. The flow variables

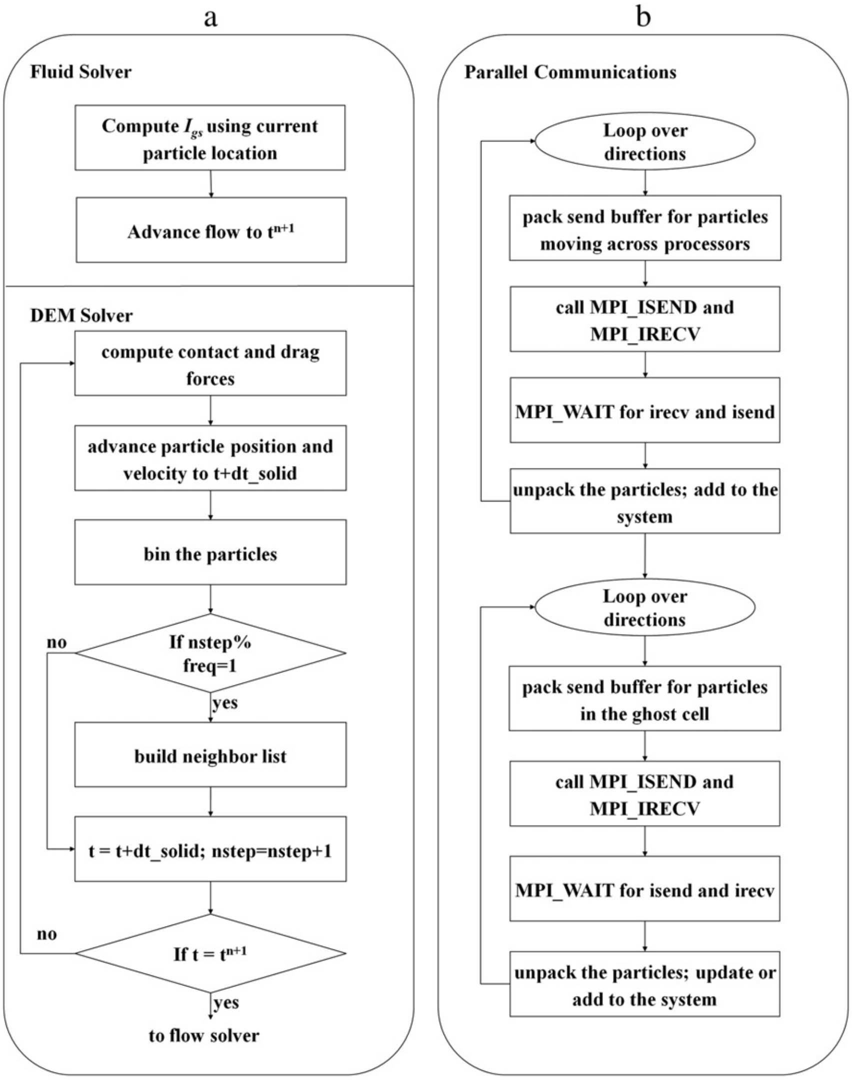
a
Fluid Solver
Compute I_{gs} using current particle location
Advance flow to t^{n+1}
DEM Solver
compute contact and drag forces
advance particle position and velocity to t+dt_solid
bin the particles
no
If nstep% freq=1
yes
build neighbor list
t = t+dt_solid; nstep=nstep+1
no
If t = t^{n+1}
yes
to flow solver
b
Parallel Communications
Loop over directions
pack send buffer for particles moving across processors
call MPI_ISEND and MPI_IRECV
MPI_WAIT for irecv and isend
unpack the particles; add to the system
Loop over directions
pack send buffer for particles in the ghost cell
call MPI_ISEND and MPI_IRECV
MPI_WAIT for isend and irecv
unpack the particles; update or add to the system

Fig. 1. a) Flow chart for a serial run. b) Parallel communication flow chart.

available at t^{n+1} is used for subsequent DEM integration. The time step for the DEM is limited by the collision time, which is given by

$$t_{col} = \pi\sqrt{\frac{k}{m_{eff}} - \frac{\eta^2}{4m_{eff}^2}} \tag{10}$$

where m_{eff} is the effective mass of the particle, η is the damping factor, and k is the spring stiffness. In the current study, a solid time step of 1/10 of t_{col} is used. Since the solid time step is normally an order of magnitude lower than the fluid time step, DEM integration is carried out in sub-steps until the time reaches t^{n+1}.

The first step in DEM involves the computation of forces due to particle–particle contacts. The simplest algorithm for resolving particle contacts is by doing an N^2 search, which checks for contact between any two particles. However, the N^2 algorithm is very expensive and hence an alternate grid-based search algorithm is employed. In this algorithm, the flow domain is divided by a uniform grid, DES-GRID, and the particles are binned on this grid. A neighbor list is created for each particle by searching particles within the neighboring cells. The list will have the particles, which lie within a certain user defined radius. In this study a value 1.2 times the diameter is used. Since the time step for DEM integration is small, the distance traveled by a particle during a single time step will be small. Hence In order to minimize the computational cost the list is only updated at a user defined frequency (after every 10 time steps in the current study). The algorithm has consistency checks and neighbor build will be invoked in extreme situations.

During each solid time step, the distance between the particle and particles in the neighbor list are computed. If there is an overlap, the spring and damping forces are computed. For the particle–wall interaction, if there is an overlap between the particle and wall, a pseudo-particle of equal diameter will be introduced to compute the contact forces. For the particle–fluid interaction, the flow velocity is interpolated to the particle location and the friction coefficient, β is computed based on the relative velocity of the particle and the fluid. In order to avoid the complexity associated with the interpolation of the fluid pressure, the pressure gradient term is computed at the cell center and it is applied to all particles which lie within that cell. The Euler explicit method is employed to advance the particle velocities based on the total force computed (Eqs. (1) and (2)) and the position of the particles are advanced based on the new velocity. After the advancement, the particles are binned to the base CFD grid and the average quantities such as solid fraction and solid velocities are computed.

3.2. Parallel algorithm

The 3D domain decomposition architecture has been used for parallelization of the DEM in compliance with the existing MFIX flow solver. The main advantage of using the domain decomposition method is high scalability even for a large number of processors, and its usability for both shared and distributed architecture machines. In this architecture each processor holds information of the particles belonging to its domain, and also particles which lie in the ghost cells. Ghost cells are the layer of cells which replicate information from the adjoining processor. During initialization, each particle is assigned a unique global ID and they are scattered across the processors based on their location. During each solid time step, there are two critical inter-processer communications required; one for the particles moving across the processor and another for exchanging the information of the particles in the ghost cells. The steps involved in the parallel communication are shown in Fig. 1b.

After time advancement, the particles are binned to the uniformly spaced DES-grid. Then the particles moving across processor boundaries are identified based on their location and they are packed into a buffer. For particles crossing processor boundaries, all information pertaining to the particle including previous velocity, previous position, neighbor list, and contact history required for the computation of the tangential contact force are sent to the neighboring process. Since particles will have different local numbers in the neighboring processor, the local ID is converted to global ID for packing. Since MFIX is based on a Cartesian grid, the communication is carried out along east–west direction, followed by north–south and then high–low with calls to MPI_BARRIER in between them. This reduces the number of MPI calls and also exchanges information of particles crossing into corner cells. For example, a particle crossing from a processor to its northwest neighbor will be first exchanged to the west and then to the north. After exchanging the processor information, the buffers are unpacked and global IDs are used to regenerate the neighbor list and the contact history based on the assigned local particle numbers in the receiving processor.

After completing the exchange of particle crossing the boundary, the particles in ghost cells are exchanged. The communication is similar to that of particle crossing processors, i.e. first along east–west direction followed by north–south and high–low. For packing the buffer, only relevant information such as position, velocity, and properties such as diameter, density are used. After the exchange, the particles are unpacked in the receiving buffer. If the ghost particles information already exists in the receiving processor, its values are updated. Otherwise, the particle will be added to the ghost particle list. Apart from these communications, collective communications are added for writing restart and output files. In addition point to point communications are added for average variables such as voidage fraction, solid velocity and interphase momentum transfer terms.

4. Verification

For verification, a pseudo-2D fluidized bed similar to the experiments of Muller et al. [2,3] is simulated at two superficial velocities 0.6 and 0.9 m/s. The bed dimension and the particle properties used for the simulation are listed in Table 1. The simulation is carried out for a total of 25 s and the results are stored at a frequency of 100 Hz. In order to get time average profiles the last 20 s of the simulation results was used, a total of 2000 frames. The simulation is carried out using the existing serial version of the MFIX-DEM and with the newly developed parallel version. For the parallel simulations, two simulations with 2 and 3 processors are carried out with domain decomposition along the width of the bed. Fig. 2 shows the comparison of average void profiles at two different heights (16.5 and 31.5 mm) for the superficial velocity of 0.9 m/s. The comparison shows no apparent deviation between the serial and the parallel runs and the maximum difference is within 1%. This verifies that the parallel simulation does not alter the accuracy of the solution. Fig. 3 shows the comparison of the parallel run with previous experiments and previous DEM simulation of Muller et al. The average voidage profiles from the DEM simulations qualitatively compare well with the experimental results. For the superficial velocity U = 0.9 m/s, the void fraction near the walls is over predicted by both current and previous [3] DEM simulations. Fig. 4 shows the time averaged vertical velocity of solid at three different axial heights for the superficial velocity of 0.9 m/s. The velocity obtained by the DEM simulation is parabolic whereas the experimental results show a flat velocity profile at y = 15 mm and high centerline velocity at height 35 mm. The DEM simulation by Muller et al. also shows a similar deviation from the experimental results.

Table 1
Simulation parameters for 2D bubbling bed.

Bed	
Width	44 mm
Depth	10 mm
Static bed height	~36 mm
Grid dimension	15 × 40 × 3 (44 × 120 × 10 mm)
Particle properties	
Diameter	1.2 mm
Density	1000 kg/m^3
Total particles	9240
Solid time step	1.5e − 5
Spring constant	200 N/m
Coefficient of friction	0.1,0.1 (particle–particle, particle–wall)
Normal coefficient of restitution	0.9,0.9 (particle–particle, particle–wall)
Tangential coefficient of restitution	
Fluid properties	
Density	1.2 kg/m^3
Viscosity	1.8e − 5
Fluid time step	2.0e − 4

5. Strong scaling analysis

For the scaling analysis, a rectangular bubbling fluidized bed, as shown in Fig. 5, with Geldart type D particles is simulated. The properties of particles, fluid, and dimensions of the system are given in Table 2. The bed width and depth are 160 times the particle diameters and the static bed height is 100 particles diameters with a total of

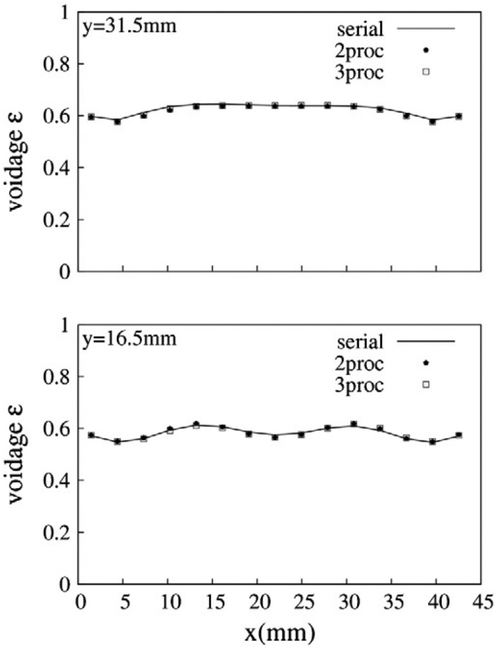

Fig. 2. Comparison of time averaged voidage profiles for serial and parallel versions for superficial velocity 0.9 m/s.

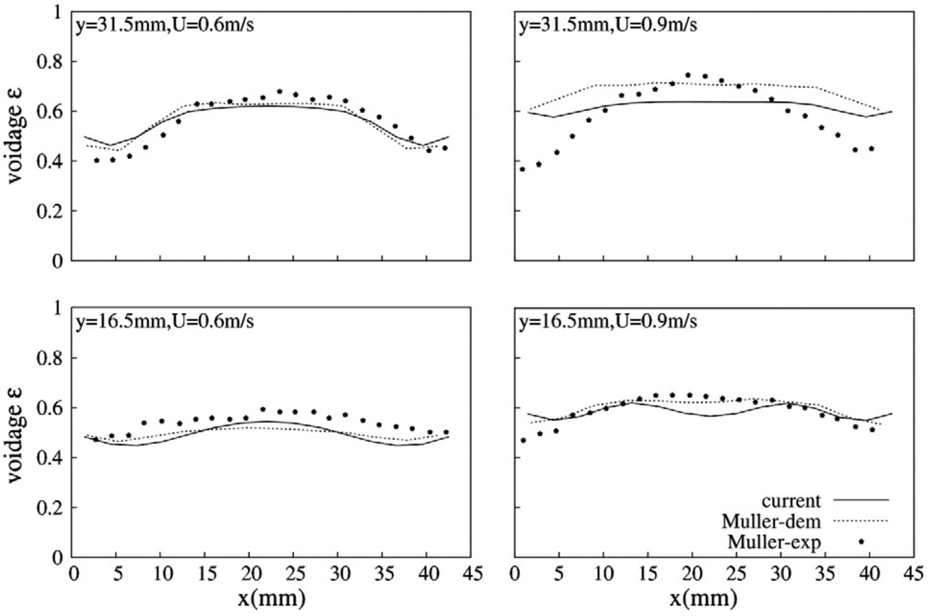

Fig. 3. Comparison of time averaged voidage profiles with the experiments and DEM simulations of Muller et al. [3].

2.56 million particles. The computation domain is resolved by 64×64×200 cells. The minimum fluidization velocity based on Ergun correlation for a void fraction of 0.44 is 1.98 m/s. In the present study, a slightly higher superficial velocity of 2 m/s is used. The bubbling bed is simulated for 5 s in order to avoid the initial transient due to start up and the scaling analysis is carried out for the last 0.1 s. The fluctuation of the inlet pressure is used to analyze the flow behavior. During the initial period, the pressure increases above the bed weight showing the initiation of fluidization followed by uniform fluctuations signifying the movement of bubbles. The visualization of the simulation results also confirms the formation and breakage of bubbles at regular intervals.

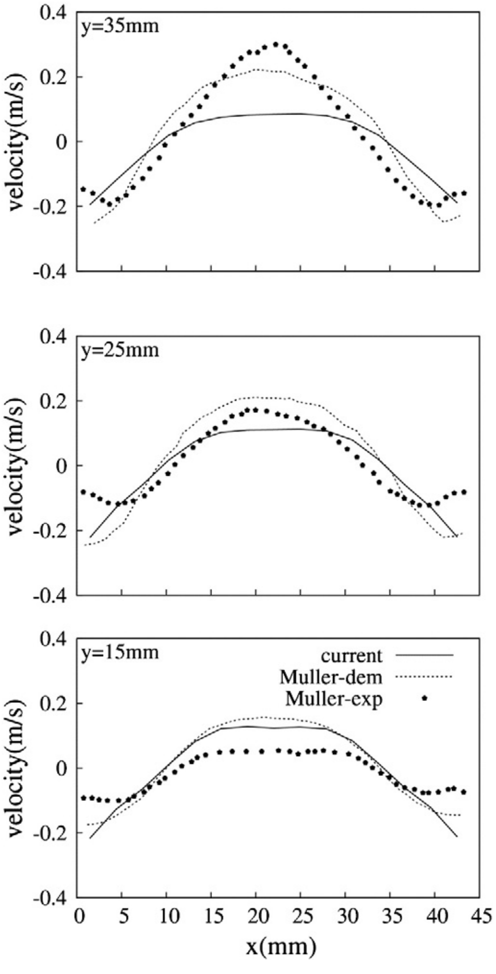

Fig. 4. Comparison of time averaged vertical velocity at different heights for U = 0.9 m/s with Muller et al. [2].

The scaling analysis is carried out using a cluster with 42 computer nodes each having a quad socket with AMD 2.3 GHz Magny Cour 8 core processors (32 processors/node) with infiniband interconnection (40 Gb/s). The TAU profiling along with the PDT is used to analyze the time taken by the individual routines and to compute the communication overheads. Fig. 6 shows the speedup and the efficiency of the DEM solver, the fluid solver, and the full simulation.

$$speedup = \frac{t_{serial}}{t_{parallel}} \tag{11}$$

$$\eta = \frac{t_{serial}}{nt_{parallel}}. \tag{12}$$

The speedup of the DEM solver is close to ideal till 64 processors and for 256 processors it is 208 with an efficiency of 81%. The flow solver shows strong scalability up to 32 processors with an efficiency of around 70%. The efficiency of the flow solver drops with further domain decomposition and a speedup of 54 is obtained at 256 processors. This drop in flow solver performance is mainly due to low number of cells/processor, which results in high communication cost compared to computation cost. For the 256 processor run, the number of cells/processor is only 3200. Fig. 7 shows the comparison of communication overheads in terms of percentage of the total computational time. The communication costs including the point-to-point (P2P) communication and the global communication such as reduction, scatter, and gather increases with the number of processors. The DEM P2P communication, which includes the exchange of particle properties crossing the processor boundary and exchanging information of the particles in ghost cells, takes 8% of the total time for the 256 processor case. The fluid P2P communication and the reduction communication take around 60% of total

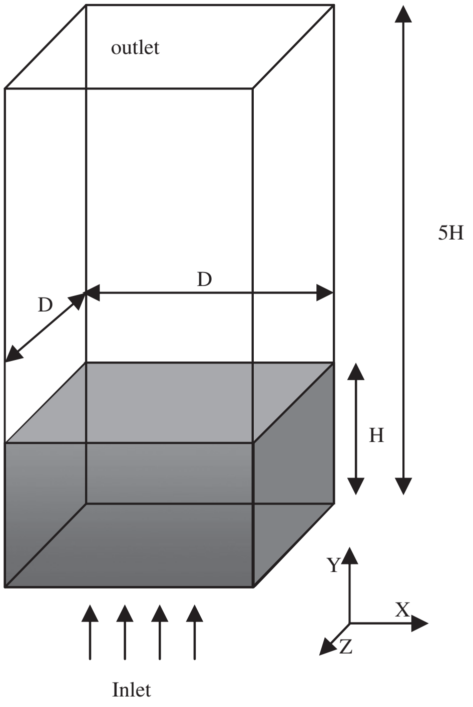

Fig. 5. Fluidized bed simulation.

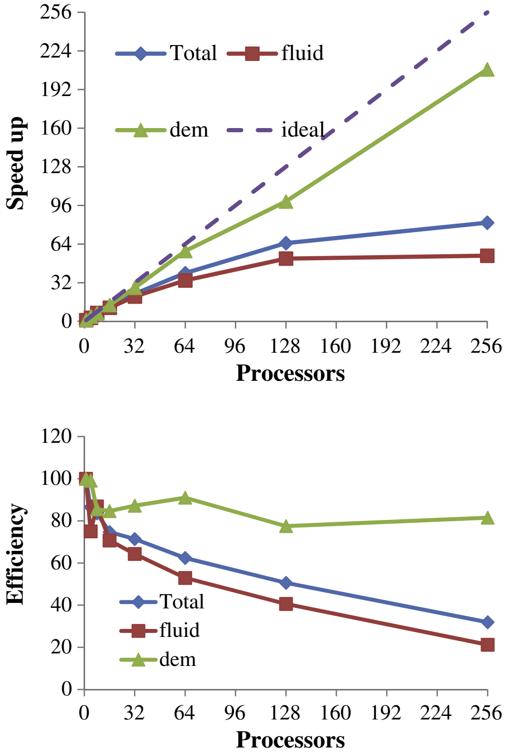

Fig. 6. Speedup and efficiency of strong scaling.

time for the 256 processors, which results in poor scaling of the fluid solver.

Fig. 8 shows the time taken by the critical routines of the DEM solver for two cases with 16 and 256 processors. The major contribution arises from the drag computation, which includes the computation of the drag force for particles and the computation of the interphase momentum transfer term, $\boldsymbol{I_{gs}}$. The other critical routines are contact force computation and the generation of the neighbor list. The time taken by the P2P communication is 15% for 16 processors and it rises to 35% for 256 processor case. In the current implementation, there is no overlap of computation with communication. Since P2P communication takes around 35% for the 256 processor case, it would be beneficial to overlap the communication with the computation of the drag force, which does not require information pertaining to the ghost particles.

Table 2
Details of 3D bubbling bed used for strong scaling analysis.

Bed	
Width	640 mm
Depth	640 mm
Static bed height	400 mm
Grid dimension	64 × 64 × 200
Particle properties	
Diameter	4 mm
Density	2700 kg/m^3
Total particles	2.56 Million
Solid time step	4e − 5
Spring constant	2400 N/m
Coefficient of friction	0.3, 0.3 (particle–particle, particle–wall)
Normal coefficient of restitution	0.95, 0.95 (particle–particle, particle–wall)
Tangential coefficient of restitution	
Fluid properties	
Density	1.2 kg/m^3
Viscosity	1.8e − 5
Fluid time step	5.0e − 4

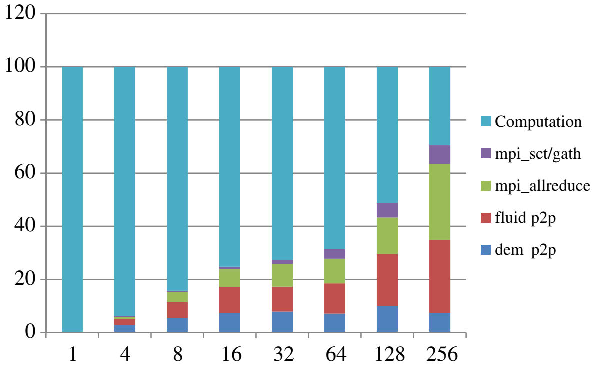

Fig. 7. Comparison of communication overheads in percentage of total computational time.

6. Weak scaling analysis

In order to analyze the performance for large scale systems, two more fluidized beds were simulated by increasing the static height

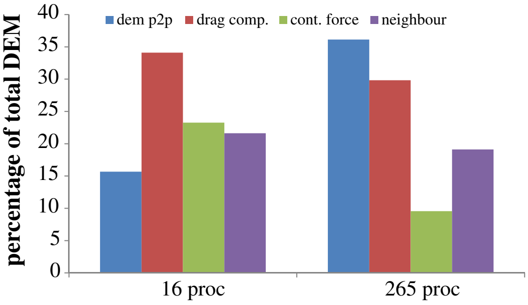

Fig. 8. Time taken by critical DEM subroutines normalized based on DEM solver time.

Table 3
Bed dimensions and simulation parameters used for weak scaling analysis.

Bed	H/D = 0.625	H/D = 1.25	H/D = 2.5
Static height	100 d_p(400 mm)	200 d_p(800 mm)	400 d_p (1600 mm)
Grid dimension	64 × 64 × 200 (640 × 640 × 2000 mm)	64 × 64 × 400 (640 × 640 × 4000 mm)	64 × 64 × 800 (640 × 640 × 8000 mm)
Total particles (million)	2.56	5.12	10.24
Max ghost cells	3200	4800	6400
Number of processors	64	128	256

by a factor two and four. The dimensions of the fluidized beds and the run details for the weak scaling analysis are provided in Table 3. The cross sectional areas of all three fluidized beds are kept constant. The computational domain along the y-direction is also increased resulting in an increase in the computational cells. The properties of the particles and inlet velocity are kept constant as was done for the strong scaling analysis (Table 2). The particles for the three fluidized beds are 2.56, 5.12 and 10.24 million in number. The processors used are 64, 128 and 256 for beds with height to depth ratio of 0.625, 1.25 and 2.5, respectively. The domain is decomposed along the depth and width maintaining approximately the equal number of particles per processor and equal number of computational cells per processors for all three systems. However, the inter-processor interface areas are not constant for the three simulations, which increase with the problem size. Similar to the strong scaling analysis, the simulations are carried out for 5 s to prevent any initial transients from affecting the scaling analysis which is carried out for the last 0.1 s.

Fig. 9 shows the actual time taken by the three fluidized bed simulations. The total time of the simulation increases with the static bed height. It is critical to note that the interface area is not constant for these three simulations and hence the time taken for P2P communications will be different. Further the problem size also affects the load imbalance, which also leads to difference in communication cost. Fig. 10 gives a measure of the interface communication as the number of processors increase. The maximum number of ghost cells that have to be exchanged for the fluid variables increase with the problem size. In addition, the maximum number of ghost cell particles to be exchanged also increase with the static height. Hence in order to delineate the effect of change in the interface area, the P2P communication costs are subtracted from the total time (Fig. 9). The total time minus P2P communication also increases by a factor of 2 for H/D = 2.5 case when compared with H/D = 0.625 case. The cost associated with the global reduction operation and collective communication is further subtracted (Fig. 9) and the results show that the actual computation time remains nearly constant for all three simulations. Fig. 11 shows the cost associated with the communications and computations normalized based on the total computational time. The total cost of the fluid solver increases slightly with the problem size as both P2P communication cost and the global reduction cost associated with the fluid solver increase. The cost increases mainly due to collective communication required for IO operation, which increases from 1% for H/D = 0.625 case to 10% for the H/D = 2.5 case. Hence employing distributed IO could reduce the total computational time for large systems.

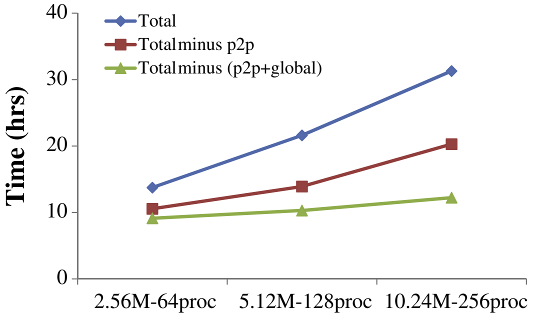

Fig. 9. Time taken by fluidized beds with different static heights.

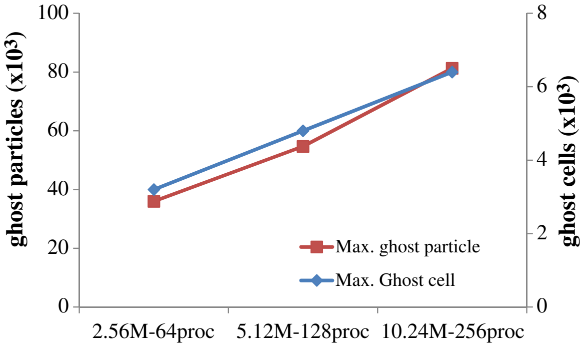

Fig. 10. Variation of interface communication for three fluidized beds.

7. Effect of bed height on bubbling hydrodynamics

The effectiveness of fluidized beds depends critically on the bubble hydrodynamics and homogenous mixing. The effect of the bed height on the hydrodynamics is analyzed by extending the simulations of the fluidized beds with different bed heights (Table 3). The simulations were carried out for a total of 12 s. Fig. 12 shows the inlet pressure fluctuations for the three beds normalized based on their weight. The inlet pressure variation shows that the fluidization reaches a steady state in less than 0.5 s. In the case of the shallow bed, the pressure fluctuations are more uniform signifying the regular formation and collapse of bubbles. In the case of a deeper bed, the pressure fluctuations are less uniform signifying a different bubbling pattern. Fig. 13 shows the instantaneous voidage fraction at time 3 s. In the shallow bed, the bubble forms from the bottom and near the walls and moves toward the center while rising to the top. For H/D = 1.25, the bubbles coalesce while rising to form larger bubbles at the center of the bed. For the deeper bed, the bubble grows to the entire bed width and collapses in the middle of the bed causing a slug flow. The collapse of the bubbles causes the non-uniform pressure fluctuation for the deeper bed case.

Time averaged voidage profiles are used to analyze the mixing and process efficiency. For time averaging, the results from 2 to 12 s at a frequency of 100 Hz for a total of 1000 frames, are used. Fig. 14 shows the cross sectional views of the voidage profiles for three bed

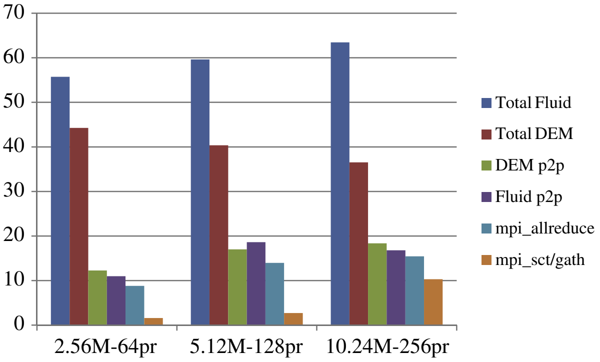

Fig. 11. Solver and communication cost for weak scaling analysis normalized by the total computational time.

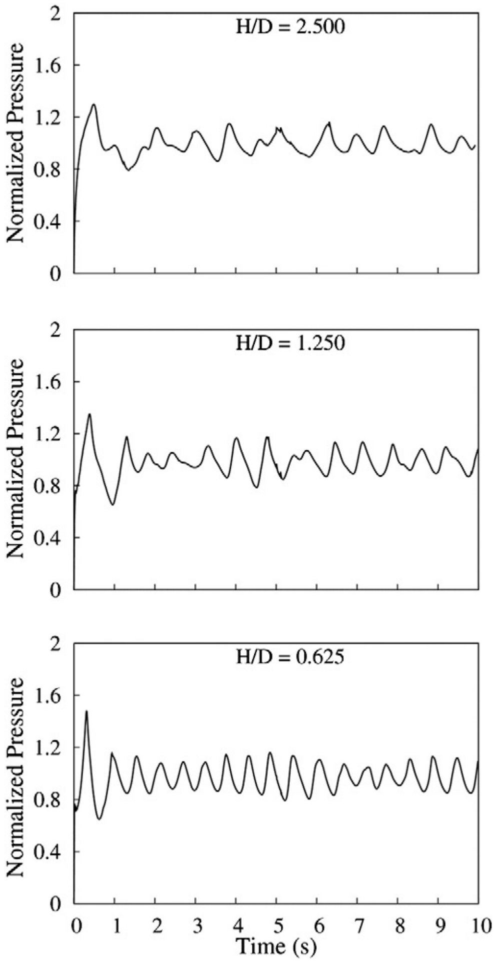

Fig. 12. Normalized inlet pressure fluctuation for fluidized beds with different static bed heights.

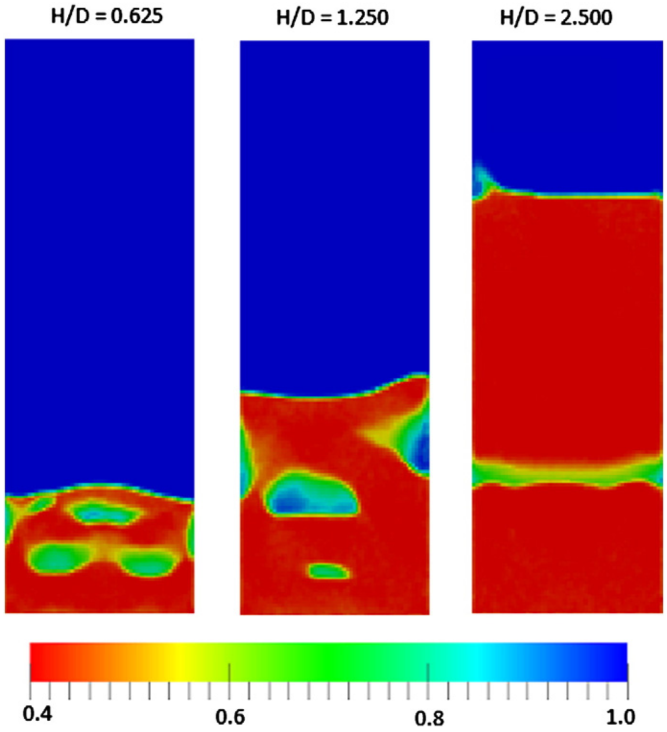

Fig. 13. Instantaneous voidage plot for three different bed heights.

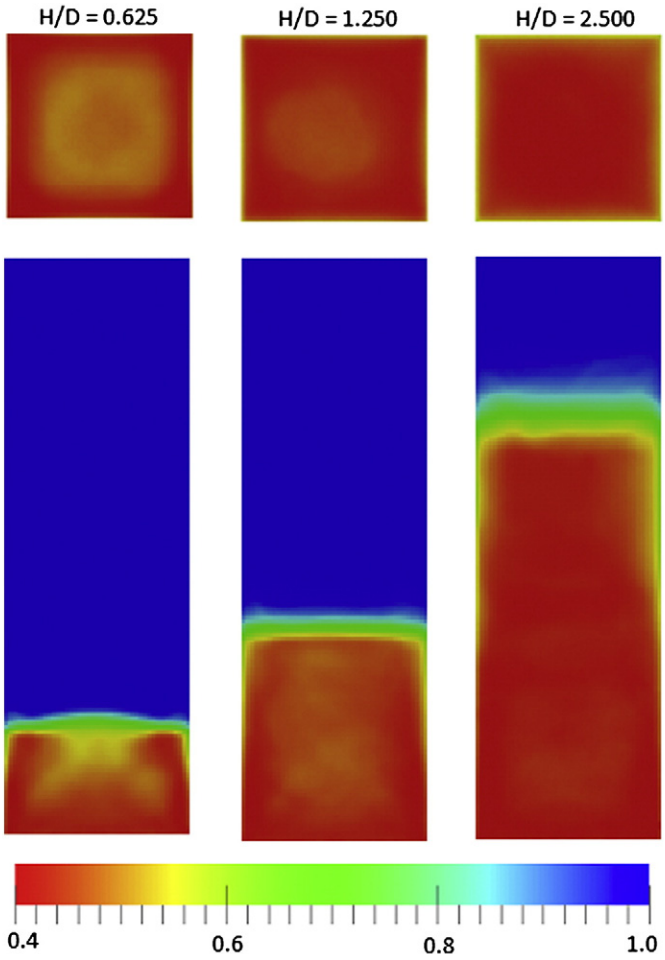

Fig. 14. Time averaged voidage showing mixing pattern for different bed heights.

heights. The y cross sections are at the middle of the bed (y = 0.5 H). The contours show high values of void fraction in the regions of bubble movement and collapse. The voidage profiles near the wall are identical showing a high value near the bottom and low value near the top. The void fraction in the core region differs significantly between the three beds. Similar observations were made by previous experimental studies [25,26]. For the shallow bed, the axial voidage profile is symmetric whereas for the H/D = 1.25 case the void fraction shows asymmetric profiles due to spiraling movement of the bubbles. In the case of H/D = 2.5, the core region has low void fraction, which is mainly due to the slug flow formation. Fig. 15 shows the space averaged voidage fraction along the height of the bed. For the shallow bed, the voidage is high and it increases sharply along the height signifying high expansion and strong fluidization characteristics. In the case of H/D = 1.25, the void fraction increases almost linearly along the bed height. For H/D = 2.5 case, the void fraction is lower and it is almost constant near the bottom.

8. Conclusion

A parallel DEM solver based on the domain decomposition method was developed for the open source code MFIX. The scaling analysis was carried out using the TAU and the PDT auto instrumentation. The results from the strong scaling analysis show good scalability for the DEM solver up to 256 processors (10,000 particles/processors) with an efficiency of 81%. The fluid solver also shows strong scalability up to 32 processor (25,000 cells/processor) with an efficiency of 70%. Detailed investigation reveals that the global communication cost and the P2P communication cost associated with the fluid solver increase

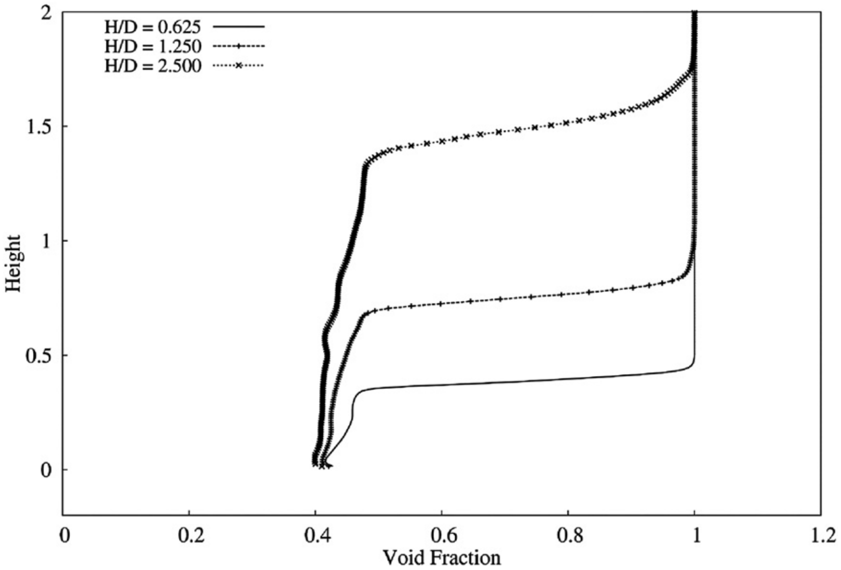

Fig. 15. Variation of voidage fraction along y direction.

significantly due to low computational cells/processor beyond 32 processors. The drag force computation, the contact force computation, and the neighbor list generation are three critical routines in the DEM solver. The cost of drag computation is around 30% of the total DEM time for the 256 processor case. Hence overlapping the P2P communication with drag computation could further improve the efficiency of the DEM solver. The weak scaling analysis was carried out by increasing the bed height of the fluidized bed. The analysis results show that the cost of global communication associated with IO increases significantly while the actual computational time remains constant. The analyses of different bed heights were extended to study the effects of the static bed height on the bubble hydrodynamics. For the superficial velocity simulated, the bubbles form and rise to the surface at regular intervals for the shallow bed. As the bed height increases the bubbles grow to the entire width and collapses in the middle of the bed causing slug flow. This change in bubble behavior results in significant difference in the time averaged voidage profiles in the core of the fluidized bed.

Acknowledgment

This technical effort was performed in support of the National Energy Technology Laboratory's ongoing research in advanced numerical simulation of multiphase flow under the RES contract DE-FE0004000.

References

[1] Y. Tsuji, T. Kawaguchi, T. Tanaka, Discrete particle simulation of two-dimensional fluidized bed, Powder Technology 77 (1) (1993) 79–87.
[2] C.R. Müller, D.J. Holland, A.J. Sederman, S.A. Scott, J.S. Dennis, L.F. Gladden, Granular temperature: comparison of magnetic resonance measurements with discrete element model simulations, Powder Technology 184 (2) (2008) 241–253.
[3] C.R. Müller, S.A. Scott, D.J. Holland, B.C. Clarke, A.J. Sederman, J.S. Dennis, et al., Validation of a discrete element model using magnetic resonance measurements, Particuology 7 (4) (2009) 297–306.
[4] M.H. Zhang, K.W. Chu, F. Wei, A.B. Yu, A CFD-DEM study of the cluster behavior in riser and downer reactors, Powder Technology 184 (2) (2008) 151–165.
[5] D. Geldart, The size and frequency of bubbles in two- and three-dimensional gas-fluidised beds, Powder Technology 4 (1) (1970) 41–55.
[6] D. Darmana, N.G. Deen, J.A.M. Kuipers, Parallelization of an Euler–Lagrange model using mixed domain decomposition and a mirror domain technique: application to dispersed gas–liquid two-phase flow, Journal of Computational Physics 220 (1) (2006) 216–248.
[7] D.W. Washington, J.N. Meegoda, Micro-mechanical simulation of geotechnical problems using massively parallel computers, International Journal for Numerical and Analytical Methods in Geomechanics 27 (14) (2003) 1227–1234.
[8] D.K. Kafui, S. Johnson, C. Thornton, J.P.K. Seville, Parallelization of a Lagrangian–Eulerian DEM/CFD code for application to fluidized beds, Powder Technology 207 (1–3) (2011) 270–278.
[9] S. Plimpton, Fast parallel algorithms for short-range molecular dynamics, Journal of Computational Physics 117 (1995) 1–19.
[10] R. Kacianauskas, A. Maknickas, A. Kaceniauskas, D. Markauskas, R. Balevicius, Parallel discrete element simulation of poly-dispersed granular material, Advances in Engineering Software 41 (1) (2010) 52–63.
[11] T. Tsuji, K. Yabumoto, T. Tanaka, Spontaneous structures in three-dimensional bubbling gas-fluidized bed by parallel DEM-CFD coupling simulation, Powder Technology 184 (2) (2008) 132–140.
[12] www.mfix.org.
[13] M. Syamlal, W. Rogers, T.J. O'Brien, MFIX documentation theory guide. US Dept of Energy, Technical Note DOE/METC-94/1004. (1993).
[14] R. Garg, J. Galvin, T. Li, S. Pannala, Documentation of open-source MFIX-DEM software for gas–solids flows, 2010.
[15] P.A. Cundall, O.D.L. Strack, A discrete numerical model for granular assemblies, Geotechnique 29 (1) (1979) 47–65.
[16] S.S. Shende, A.D. Malony, The TAU parallel performance system, International Journal of High Performance Computing Applications 20 (2) (Summer 2006) 287–311.
[17] K.A. Lindlan, J. Cuny, A.D. Malony, S. Shende, F. Juelich, R. Rivenburgh, et al., A tool framework for static and dynamic analysis of object-oriented software with templates, in: Proceedings of the 2000 ACM/IEEE Conference on Supercomputing (CDROM), IEEE Computer Society, Dallas, Texas, United States, 2000, p. 49.
[18] T. Li, R. Garg, J. Galvin, S. Pannala, Open-source MFIX-DEM software for gas–solids flows: part I—verification studies, Powder Technology 220 (2012) 122–137.
[19] T. Li, R. Garg, J. Galvin, S. Pannala, Open-source MFIX-DEM software for gas-solids flows: part II-validation studies, Powder Technology 220 (2012) 138–150.
[20] D. Gidaspow, Multiphase Flow and Fluidization, Academic Press, 1994.
[21] S. Ergun, Fluid flow through packed columns, Chemical Engineering Progress 48 (2) (1952) 89–94.
[22] C.Y. Wen, Y.H. Yu, Mechanics of fluidization, Chemical Engineering Progress Symposium Series (1966) 100–111.
[23] D. Lathouwers, J. Bellan, Modeling of dense gas–solid reactive mixtures applied to biomass pyrolysis in a fluidized bed, 2000.
[24] M. Syamlal, MFIX Documentation Numerical Technique. US Dept of Energy, Technical Note DOE/MC31346-5824. (1998).
[25] D. Escudero, T.J. Heindel, Bed height and material density effects on fluidized bed hydrodynamics, Chemical Engineering Science 66 (16) (2011) 3648–3655.
[26] H. Zhu, J. Zhu, G. Li, F. Li, Detailed measurements of flow structure inside a dense gas–solids fluidized bed, Powder Technology 180 (3) (2008) 339–349.