Discrete Particle Simulation of Gas-Solid Two-Phase Flows with Multi-scale CPU-GPU Hybrid Computation

Ming Xu, Feiguo Chen, Xinhua Liu, Wei Ge, Jinghai Li

PII: S1385-8947(12)00940-0
DOI: http://dx.doi.org/10.1016/j.cej.2012.07.049
Reference: CEJ 9564

To appear in: Chemical Engineering Journal

Please cite this article as: M. Xu, F. Chen, X. Liu, W. Ge, J. Li, Discrete Particle Simulation of Gas-Solid Two-Phase Flows with Multi-scale CPU-GPU Hybrid Computation, Chemical Engineering Journal (2012), doi: http://dx.doi.org/10.1016/j.cej.2012.07.049

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.
Discrete Particle Simulation of Gas-Solid Two-Phase Flows
with Multi-scale CPU-GPU Hybrid Computation

Ming Xu, Feiguo Chen, Xinhua Liu, Wei Ge*, Jinghai Li

State Key Laboratory of Multiphase Complex Systems, Institute of Process Engineering,
Chinese Academy of Sciences, Beijing 100190, China
*Author to whom correspondence should be addressed.
Tel: +86-10-8254 4943; Fax: +86-10-6255 8065; E-mail: wge@home.ipe.ac.cn

Abstract Though discrete particle simulation (DPS) has been widely used for investigating
gas-solid flows from a more detailed level as compared to traditional two-fluid models (TFM), it
is still seriously limited by the computational cost when large scale systems are simulated.
GPUs (graphic processing units), with their massive parallel architecture and high floating
point performance, provide new possibilities for large-scale DPS. In this paper, a multi-scale
CPU (central processing unit)-GPU hybrid computation mode is developed, in which the fluid
flow is computed by CPU(s) while the particle motion is computed by GPU(s). To explore its
feasibility, this mode is adopted to simulate the flow structures in the fluidization of Geldart D
and A particles, respectively. Further coupled with an EMMS (energy minimization multi-scale)
based meso-scale model, the flow behavior in an industrial fluidized bed is finally simulated,
shedding light on the engineering applications of DPS, as an alternative to TFM.

Keywords Gas-solid two-phase flow; Discrete particle simulation; Multi-scale; CPU-GPU
hybrid computation; Fluidization

1. Introduction

Gas-solid two-phase flows are very common in industrial processes, displaying remarkable
spatio-temporal multi-scale heterogeneities. The evolution of such structures is very
complicated due to the interactions among the individual particles, the gas phase and the
boundaries, which have to be considered in detail for reasonable prediction of the flow
behavior.
Discrete particle simulation (DPS) tracks down the motion of each particle by solving Newton’s equations of motion while the continuous gas phase is described by Navier-Stokes equations and usually solved by grid-based numerical methods. DPS seems to be capable of providing the micro-scale information of the complex interactions and reproducing the macro-scale dynamic flow behavior, and is, therefore, widely used in investigations on gas-solid flows [1-12]. However, its application is seriously limited by the huge computational cost when simulating large scale systems and most simulations still focus on experimental systems with Geldart [13] D particles or small bubbling systems with Geldart A/B particles.

Discrete element method (DEM) proposed by Cundall and Strack [14] is the most popular method in DPS for the computation of the particle phase. Taking advantage of the massive parallel architecture of GPUs and the development of its programming tools, such as CUDA [15], DEM simulation of $10^8$–$10^9$ particles is becoming realistic, which is already of practical interest to engineering applications. Radeke et al. [16] studied the size effects in granular flows with DEM using GPUs. Sakai et al. [17] conducted a parallel DEM simulation on GPUs. Xu et al. [18] achieved quasi-real-time simulation of an industrial rotating drum in which about $9.6\times10^6$ particles are treated with 270 GPUs. As the computation cost of DPS is mostly from the particle phase, GPU computing seems to provide a feasible way for the engineering application of DPS also.

In this paper, a multi-scale CPU-GPU hybrid computation mode is developed, in which the fluid flow is computed by CPU(s) while the particle motion is calculated by GPU(s). This assignment of the workload is consistent to the characteristic of gas-solid systems. As the source term exerted by the solid phase through the drag force is strong and highly irregular, the Navier-Stokes equation for the gas-flow is solved with the Semi-implicit Method for Pressure-linked Equation (SIMPLE) algorithm [19], so as to ensure the robustness and accuracy of the numerical scheme. However, there are many logical operations as well as global data dependence and iterations in this algorithm, which is more suitable for CPUs in terms of relative efficiency and the total computational cost. On the other hand, the particle motion is solved by Newtonian mechanics with DEM in explicit scheme, which is most suitable for GPUs, as demonstrated in previous simulations [16-18].
To explore its feasibility, this hybrid computation mode is first adopted to simulate the fluidization of Geldart D and Geldart A particles. Some interesting details of the particle behavior in experimental-scale systems are revealed and the speed-up in computation time is demonstrated. Finally, we will report some preliminary work to couple an EMMS-based meso-scale model with this computation mode, in searching for a feasible approach to the industrial applications of DPS.

2. Model

As usually adopted in DPS, the fluid flow is described by the Navier-Stokes equations. The continuum equation and the momentum equation are

\[
\frac{\partial (\varepsilon \rho g)}{\partial t} + (\nabla \cdot \varepsilon \rho g \mathbf{u}) = 0,
\]

\[
\frac{\partial (\varepsilon \rho g \mathbf{u})}{\partial t} + (\nabla \cdot \varepsilon \rho g \mathbf{u} \mathbf{u}) = -\varepsilon \nabla p - S_p - (\nabla \cdot \varepsilon \tau_g) + \varepsilon \rho g g,
\]

where \( \varepsilon \) is the voidage, \( \rho_g \) is the gas density, \( \mathbf{u} \) is the velocity vector, \( p \) is the gas pressure, \( S_p \) is the momentum exchange source term, \( \tau_g \) is the viscous stress tensor, and \( g \) is the gravity acceleration. According to Newton's third law of motion, \( S_p \) can be obtained by summing the drag force exerting on all the particles in the cell, i.e.,

\[
S_p = \frac{1}{V_{cell}} \sum F_d.
\]

where \( V_{cell} \) is the volume of the cell, and \( F_d \) is the drag force on an individual particle. Accordingly, the motion of individual particle is solved by Newton's second law considering the external forces acting on each particle

\[
m_p \frac{d\mathbf{v}}{dt} = m_p g + F_d + F_c
\]

where \( m_p \) is the mass of the particle, \( \mathbf{v} \) is the velocity vector, and \( F_c \) is the collision force between the particle and its neighbors as well as the wall. The drag force exerting on \( F_d \) is determined by
\[ F_d = \frac{V_p \beta}{(1 - \epsilon)} (u - v) \] (5)

where \( V_p \) is the particle volume. \( \beta \) is the inter-phase momentum exchange coefficient which is usually obtained from the Ergun equation [20] and the correlation proposed by Wen and Yu [21] according to different voidages,

\[ \beta = \begin{cases} 150 \frac{(1 - \epsilon)^2}{\epsilon} \frac{\mu_g}{d_p^2} + 1.75(1 - \epsilon) \frac{\rho_g}{d_p} |u - v| & (\epsilon < 0.8) \\ \frac{3}{4} C_d \frac{\epsilon(1 - \epsilon)}{d_p} \rho_g |u - v| e^{-2.65} & (\epsilon \geq 0.8) \end{cases} \] (6)

where \( d_p \) is the particle diameter. The drag coefficient \( C_d \) is given by

\[ C_d = \begin{cases} \frac{24}{\text{Re}_p} (1 + 0.15 \text{Re}_p^{0.687}) & \text{Re}_p < 1000 \\ 0.44 & \text{Re}_p \geq 1000 \end{cases} \] (7)

where \( \text{Re}_p \) is the particle’s Reynolds number defined as

\[ \text{Re}_p = \frac{\epsilon \rho_g |u - v| d_p}{\mu_g} \] (8)

The collision force representing the complex interactions between particle-particle and particle-wall is computed by DEM in which the interactions between individual particles are simplified as spring, dashpot and friction effects. As an exploratory work focusing on the hybrid computation mode, the DEM model is further simplified to facilitate the GPU implementation, that is, only the translational motion is computed without consideration of the friction effect and rotation. Therefore, the following linear spring-dashpot model is adopted in this work,

\[ F_c = -k \delta_{ij} - \eta V_{ij} \] (9)

where \( k \) is the spring stiffness, \( \eta \) is the damping coefficient, \( \delta_{ij} \) and \( V_{ij} \) are the overlap and relative velocity between particles \( i \) and \( j \). The damping coefficient \( \eta \) is calculated according to

\[ \eta = 2 \sqrt{k \frac{m_i m_j}{m_i + m_j} \ln(1/e) \frac{\ln(1/e)}{\sqrt{\pi^2 + [\ln(1/e)]^2}}} \] (10)
where $e$ is the restitution coefficient, $m_i$ and $m_j$ are the mass of particle $i$ and $j$ respectively.

3. The multi-scale CPU-GPU hybrid computation mode

The schematic diagram of the multi-scale CPU-GPU hybrid computation mode is illustrated in Fig. 1. At the macro-scale, the master CPU is responsible for the task partition and general control, and the whole computation region is divided into sub-regions by the master CPU. Then at the meso-scale, each sub-region is computed in a single process running on a CPU plus a GPU. And accordingly, the computation of the fluid phase is implemented on the CPU while that of the particle phase is implemented on the GPU. For the computation on the GPU, the sub-region of the particle phase is further divided into many blocks and the motion of each particle in a block is computed with a thread at the micro-scale. As up to hundreds of thread processors (480 for NVIDIA GTX295 and 448 for NVIDIA C2050) could work simultaneously in one GPU, the acceleration effect is very significant. The master CPU, the slave CPUs, and the blocks and threads in the GPUs are working together to constitute the multi-scale CPU-GPU hybrid computation mode.

The general multi-scale hybrid computation procedure is summarized in Fig. 2, and the GPU-based algorithm for DEM is based on our previous works [18,22]. In particular, the mapping of particles to cells is done by CPUs and copied to GPUs before updating the particle status. And before solving the Navier-Stokes equations, the information of the boundary conditions has been exchanged among neighboring processes. Then the momentum and continuum equations of the gas phase are solved on CPUs. After that, the information of the gas phase, such as voidage and velocities, is copied to GPUs for the computation of particle motion. Subsequently, the information of the boundary particles is exchanged before solving the motion of the particle phase.

For solving the equations of particle motion, the Leap-Frog scheme [23] is adopted. The
position and velocity of each particle are updated by the corresponding thread. And the mapping of particle to cells is also updated along with the moving of the particles. Therefore, each thread calculates the cell indexes of the corresponding particle in parallel. The particles having new cell indexes are then removed from the previous cells and inserted into the current cells. To avoid writing conflict, the operations of removal and insertion are implemented only by one thread in each block. The neighbor search is followed based on the new map of cells. One thread loads the data of the home cell, and then the data of the neighboring cells are loaded into the shared memory to be searched one by one. The shared memory is used here to speed up the reading of the data from GPU global memory. After the data is completely loaded, each thread loads the home particle, and then loops the neighboring particles to compute and accumulate the collision forces. Consequently, the drag force exerting on the particle is calculated according to equations (5)-(8). Finally, the updated information of the particles is written back to the global memory.

The current implementation of the multi-scale computation mode is straightforward and fundamental, more advanced optimizations are still underway which can be expected to improve its performance significantly. For example, the CPU computation should make use of the multi-core architecture of modern CPUs in the shared-memory manner. Meanwhile, more sophisticated particle indexing algorithms, such as the combination of the current cell-list method with the neighbor-list method [24], could be adopted. The communication overhead between CPUs and between CPU-GPUs can also be concealed by overlapping with computation. This overlapping is critical when more complicated particle interactions are considered, such as friction and rotation. However, considerable speed-up is already seen with the current implementation, as will be demonstrated in the follows.

4. Results and discussion

The multi-scale CPU-GPU hybrid computation mode is verified with simulations on the fluidization of Geldart D and A particles, respectively, using single and multi process(es). The simulated fluidization behavior and the computational performance is reported in this section.

4.1. Simulation of fluidization behavior of Geldart D particles
Three 2D gas-solid fluidization systems, with the same mean voidage and inlet gas velocity but different geometry of the fluidized bed, are simulated with the computation mode described above, using an Intel Xeon 5430 CPU and a NVIDIA GTX295 GPU. For the gas phase, the velocity at the inlet is uniform, the no-slip boundary condition is applied at the side walls, and zero normal gradient is assumed at the top exit. The particles are initially distributed randomly in the bed. The number of particles moving out of the ceiling is counted in each step, and the same number of particles is inserted at the bottom with an average velocity of their neighbors. The simulation conditions and parameters are listed in Table 1.

The simulation reveals the evolution of flow structures in different cases, as illustrated in Fig. 3-5. Under the same operation conditions, it shows consistent changes in their characteristics in response to bed size. Fig. 3 is for a bed with small diameter and typical slug fluidization is observed. Initially dispersed particles aggregate to form plugs, and then particles fall apart from the rear of the bulk continuously, leaving a homogeneous tail behind. These two types of structure are found to coexist in the steady state. As the bed diameter increases, flow patterns similar to fast fluidization are observed, as shown in Fig. 4. The particle clusters are found to be in continuous formation, deformation and dissolution. At the bottom of the bed, the particles are always pushed to the wall and carried back to the center at the top by the gas phase. With further increase in bed diameter, as shown in Fig.5, the heterogeneity is apparently more distinguished in that the dense region with large swarms of particles at the bottom and the dilute region with dispersed particles at the top coexist. In the dense region, the particles are gathered to form clusters and accelerated upward until the gas phase pushes them aside to the walls. And then the particles slip down along the wall and aggregate into new clusters. This flow behavior is typical for the so-called core-annulus structure.

<Fig. 3>

<Fig. 4>

<Fig. 5>
The qualitative aspect of the simulation results is consistent to general observations on fluidization and with CPU-based simulations previously. The speed ratios (CPU+GPU vs CPU) are 1.05(case a), 1.44(case b), 2.37(case c), respectively. The speed-up in these cases is not attractive, due to overall scale of the problem and the limited number of particles, which determines the ratio of computational costs for the gas and solid phases. However, it suggests that higher speed-up ratio could be achieved with larger systems, which are of primary interest to this computation mode and are simulated next.

4.2. Clustering behavior of Geldart A particles in fast fluidization

Previously, most DPS attempts in literatures have used Geldart D particles, as the computational cost for large particles is relatively low, due to the use of less particles and larger timestep. However, Geldart A particles, which are more common in engineering fluidization systems, were seldom addressed so far. With this hybrid computation mode, a gas-solid fluidization system consisting of $10^6$ Geldart A particles could be investigated with only one Intel Xeon 5520 CPU and a NVIDIA C2050 GPU board.

The layout and boundary conditions are the same as that in Sec. 4.1, and the main simulation parameters are listed in Table 2. Although it is a two-dimensional simulation, the computational cost involved is still enormous considering the combination of particle number, timestep and physical time. The typical computational time in this case is 0.24s per day, or $5.56 \times 10^6$ particle updates per second (PUPS), and the speed-up ratio is about 10. In comparison, the speed-up in pure DEM simulations is typically 20 folds as in the case of Xu et al. [18]. Load balance and communicational costs due to the two-phase coupling and the dynamical heterogeneity of particle distribution are the main reasons for this difference. However, the speed-up is still significant in the sense that, the dominant computational cost in DPS now shifts to the gas phase and no longer proposes a disadvantage when compared with fully continuum-based two-fluid models [25,26]. This change opens up the possibility for its industrial application.

<Table 2>
With this simulation, the evolution of the heterogeneous structures in fast fluidization is revealed at both large scale and high resolution. As shown in Fig. 6, starting from a random and homogeneous distribution, many small wire-like clusters come into being shortly afterwards. Then, particles are more likely to be carried upwards in the center and fall down near the wall. Meanwhile small clusters continue to aggregate into bigger strip-like clusters, and the relative velocities between the clusters and dispersed particles are increasing. Clusters capture dispersed particles or other clusters to aggregate into swarms, or break into dispersed particles. And behind the clusters, there are always dilute regions.

<Fig. 6>

The characteristics of the heterogeneous structures in the flow also vary at different locations in the fluidized bed. A dense regime at the bottom, a dilute regime at the top and a transition regime in the middle are found coexistent in Fig. 7A. Particles move upwards in the center and downward near the wall, and the radial core-annular structure is observed clearly. There have been many investigations on the structures and behavior of clusters in the literatures [27-32], and different structures have been observed, such as the V- or U-shaped and their inverted counterparts, as well as the horseshoe-shaped with thin downward tails. All these shapes can be found in our simulations. As show in Fig. 7B, clusters like thin wires can be bended to horseshoe in the top region. As the height decreases, more clusters are connected with each other, and the U- or inverse U-shaped are seen in Fig. 7C. In the middle region, clusters moving upwards collide with descending clusters and aggregate to strip-like larger clusters. The clusters in V- or inverse V-shape or even more complicated shapes can be found in Fig. 7D. In the upper part of the dense regime, some clusters capture dispersed particles with higher velocity below them and falling particles or clusters above them, and are then both enlarged and thickened. Some other clusters break up into smaller clusters and dispersed particles, which leads to very different cluster structures are seen in Fig. 7E. In the bottom region, the particles fall down along the wall and aggregate into swarms moving to the center, and then the swarms are pushed away by the gas. Dispersed particles in the center have very high velocity upwards and swarms near the wall move downwards. A clear interface between them can be found.
The simulation results also illustrate the details of cluster deformation, dissolution and regeneration, as shown in Fig. 8. The cluster in the red dashed frame first twists slightly like a strap fluttering in breeze as shown in Fig. 8A, then its left part moves upward while the right part moves downward, leading to an inclined S-shape as shown in Fig. 8B. It then deforms to a N-shape while the left part of the cluster in the blue ellipse begins to break up as shown in Fig. 8C. Subsequently, the peak of the left part breaks up into two thin wire-like clusters and between them are dispersed particles. As the cluster changing its structure, small clusters below it continues to aggregate and a new cluster is formed as shown in the brown ellipse in Fig. 8D. Then, the left part of the cluster breaks up as shown in Fig. 8E. The new cluster behind then changes to V-shape again and undergoes a new round of breakage as shown in Fig. 8F. From these details, the characteristic dimensions of the clusters can be measured and analyzed, as discussed in more rigorous direct numerical simulations (DNS) [33,34]. However, the scale of DNS is still very limited and DPS seems to be the only feasible approach to understand clustering behavior across its full scale spectrum.

4.3. Comparison to experiments of a quasi-3D fluidized bed

While the results described above seem reasonable and have revealed some interesting details in the development of multi-scale structures in fluidization which are not well-known in experiments previously, direct comparison to experimental results are not possible since only 2D simulations are carried out. Going to fully 3D simulations is, however, very costly at the moment even with our GPU-accelerated mode. Therefore, to further validate the simulation method and explore more realistic systems, a quasi-3D experimental fluidization system is established as shown in Fig. 9. The system is then simulated with one Intel E5520 CPU plus 10 NVIDIA GTX295 GPU boards. Some preliminary results have appeared in [35] already.
The simulation conditions and parameters are listed in Table 3. For simplicity, only the fluidized bed is simulated and it is operated under the turbulent fluidization regime so that the circulating of solids, although not absolutely absent, can be safely overlooked in the simulation. Due to the large aspect ratio of the rectangular cross-section in quasi-3D bed, the gas phase is still computed in 2D, that is, only one cell along the depth side, but a parabolic velocity profile is assumed in this direction, with its mean value obtained from the 2D solver. Although it is a very inaccurate guess, it is a cheap trick to preserve the 3D characteristic of the flow qualitatively. Meanwhile, totally 1.6 million solid particles with 1mm diameter are simulated in 3D, which corresponds closely to the particle number in the fluidized bed.

Typical snapshots from the simulations and the corresponding experimental results are shown in Fig. 10, with apparently similar flow behavior and heterogeneous structures. Irregular bubbles form as shown in Fig. 10-1 and -4, grow up or deform (Fig. 10-2 or -5) and then break up (Fig. 10-3 and -6). As the bubble structure changes continuously, particles are carried upwards in the center and fall down along the walls. A steady state with fluctuating bed surface is maintained hereafter, as shown in Fig. 10-5 and -6. The bed expansion rates are also similar, although only visual comparison is made so far. More comprehensive and quantified comparison is still underway and will be reported in due course. However, it can be said that the results obtained in this multi-scale computing mode is reasonable and encouraging. The simulations proceed at a speed of 4 seconds per day, and the performance is $7.4 \times 10^7$ PUPS using 10 GPUs.

5. Exploratory industrial-scale DPM simulation with an EMMS-based meso-scale model

To explore the capability of this multi-scale hybrid mode, a case consisting of $10^7$ Geldart A particles in a circulating fluidized bed (CFD) riser is simulated with 20 NVIDIA C2050 GPU boards. This riser is part of the Virtual Process Engineering Platform [35,36] at IPE, as shown
in the left of Fig.11. The simulation conditions and parameters are listed in Tab. 5.

<Table 5>

<Fig. 11>

Snapshots from the simulation are shown in Fig.11. In general, the core-annulus structure is reproduced globally and various clustering scenario, as discussed in the previous cases, are also found at different locations. The computation evolves at 4.63×10^7 PUPS, and the efficiency not as high as expected, due to load balance issues and more iterations in the gas phase solver at large scales which take more than 90% of the total time. No restrict experimental comparison has been made, as for simplicity the simulation still employs same inlet-outlet conditions used the previous cases, which is different from the real conditions of the rise in the full CFB loop. Full loop simulation with complicated reactor geometry is still underway.

Although the foregoing discussion have demonstrated that with the acceleration of this multi-scale mode, DPS may become an alternative to two-fluid models for industrial-scale simulation of gas-solid flow, one critical issue is yet to be solved. That is, for two-fluid models, coarse grids can be used when simulating very large systems, and with the help of meso-scale models, such as the EMMS model [37-40] for describing the sub-grid structure, reasonable accuracy can be maintained [41]. In this way, the computational cost can be reduced even more significantly and most industrial problems can be tackled. For DPS, however, only elementary industrial scale systems are possible if each physical particle should be considered. Methods for coarse-graining of the actual solid particles to a much smaller number of computational elements are yet to be developed. Multiphase particle-in-cell (MPPIC) [42,43] method may present such an attempt, where the motion of the particle parcels, rather than individual particles, are tracked, and direct interactions among the parcels are replaced by the stress computed by returning to a grid-based continuum description.

In this work, we try to take the inherent meso-scale structures in the flow, that is, the clusters, as the natural choice of the computational elements. For this purpose, the
complicated geometry and behavior of the clusters must be simplified drastically to a solvable model. The EMMS model is, therefore, employed here to give a reasonable estimation of the characteristic hydraulic diameter of the clusters, and the clusters are assumed to be spherical as a first approximation. The clusters are treated as porous large particles, and the drag forces on the clusters are summations over their member particles, which can be readily described by the EMMS model. Their deformation and mass exchanges are considered much slower than their momentum exchanges, and hence the clusters still interact as discrete elements, but with a much softer potential and higher tangential friction and inelasticity. The dilute phase in the EMMS model is treated as a fluid phase and its properties, such as density, are also provided by the EMMS model. In this method, the flow structures of the gas-solid system in an industrial reactor have been simulated. The simulation conditions and parameters are listed in Table 4.

The simulation results of the spatial-temporal heterogeneous structures in the reactor are illustrated in Fig. 12, some first results has been reported previously [35]. In this case, the flow structure is similar to that of fast fluidization. There is a core-annulus structure in the wider bottom region, and particles are carried out of the reactor quickly when it has entered the narrower upper region. The results are supported by the limited data from industry, in terms of the overall pressure drop, however, more detailed in validation is still necessary. For the moment, we may say that the integration of EMMS with DPS, together with multi-scale hybrid computation mode, is a promising computational approach towards wider industrial application of DPS. The simulation now proceeds at a speed of 5 seconds per hour with 20 NVIDIA C2050 GPU boards, that is the ratio of the computation time to physical time is 720. It is reasonable to expect that such speed would be reached in 3D simulations in the future, considering the room for further optimization of the method and algorithm, and the use of more CPUs and GPUs.

6. Conclusions

In this paper, a multi-scale CPU-GPU hybrid computation mode is developed for DPS of gas-solid systems. The computation of the fluid phase is implemented on CPUs and that of the particle phase is implemented on GPUs, taking the respective advantages of implicit and
explicit numerical schemes. This mode has been demonstrated to improve the capability and efficiency of DPS. The fluidization of Geldart A particles is becoming much less costly and feasible with this mode. With further integration of a meso-scale description of particle clusters and their interactions with the gas phase based on the EMMS model, industrial application of DPS, or in more general sense, Eulerian-Larangian approaches, is becoming reasonable and attractive.

Acknowledgement

The work is sponsored by Chinese Academy of Sciences (CAS) under the grant No. XDA07080105, Ministry of Science and Technology under the grant No. 2012BAE05B01, National Natural Science Foundation under the grant No. 51106168, and Key Laboratory of Tobacco Processing Technology of China National Tobacco Corporation.

References

[7] M. Ye, M.A. van der Hoef, J.A.M. Kuipers, A numerical study of fluidization behavior of


List of tables

Table 1. Simulation conditions and parameters of Geldart D particles on single process

Table 2. Simulation conditions and parameters of Geldart A particles on single process

Table 3. Simulation conditions and parameters of a quasi-3D fluidization system

Table 4. Simulation conditions and parameters for an industrial-scale reactor
Table 5. Simulation conditions and parameters of Geldart A particles on multi processes

List of Figures

Fig. 1. Schematic diagram of the multi-scale CPU-GPU hybrid computation mode

Fig. 2. Flow-sheet of the multi-scale CPU-GPU hybrid computation mode

Fig. 3. Dynamic evolution of flow structures in slugging fluidization

Fig. 4. Dynamic evolution of flow structures in fast fluidization

Fig. 5. Simulation of core-annulus structure

Fig. 6. Dynamic evolution of heterogeneous flow structures in fluidization of Geldart A particles

Fig. 7. Different heterogeneous flow structures in fluidization of Geldart A particles at t=2.7s

Fig. 8. Temporal development of a cluster in fluidization of Geldart A particles

Fig. 9. Experimental setup of the quasi-3D fluidization system

Fig. 10. Comparison between experimental and simulation results in the quasi-3D fluidization system

Fig. 11. Snapshot of the spatial heterogeneous structures in the riser

Fig. 12. Dynamic evolution of flow structures in the industrial reactor
Table 1. Simulation conditions and parameters for Geldart D particles on single process

<table>
<thead>
<tr>
<th>Particle</th>
<th>case a</th>
<th>case b</th>
<th>case c</th>
<th>Gas</th>
<th>Geometry</th>
</tr>
</thead>
<tbody>
<tr>
<td>density ( \rho_p ) (kg·m(^{-3}))</td>
<td>1170</td>
<td>1170</td>
<td>1170</td>
<td>density ( \rho_g ) (kg·m(^{-3}))</td>
<td>1.29</td>
</tr>
<tr>
<td>number ( N_p )</td>
<td>32000</td>
<td>64000</td>
<td>128000</td>
<td>velocity ( u_g ) (m·s(^{-1}))</td>
<td>7.5</td>
</tr>
<tr>
<td>restitution coefficient</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>viscosity ( \mu_g ) (kg·m(^{-1})·s(^{-1}))</td>
<td>1.8×10(^{-5})</td>
</tr>
<tr>
<td>spring stiffness</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>time step ( \Delta t ) (s)</td>
<td>1×10(^{-3})</td>
</tr>
<tr>
<td>diameter ( d_p ) (mm)</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>time step ( \Delta t ) (s)</td>
<td>1×10(^{-5})</td>
<td>1×10(^{-5})</td>
<td>1×10(^{-5})</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Geometry</th>
<th>case a</th>
<th>case b</th>
<th>case c</th>
</tr>
</thead>
<tbody>
<tr>
<td>height ( H ) (m)</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
</tr>
<tr>
<td>width ( W ) (m)</td>
<td>0.25</td>
<td>0.5</td>
<td>1.0</td>
</tr>
<tr>
<td>grid number</td>
<td>2500</td>
<td>5000</td>
<td>10000</td>
</tr>
</tbody>
</table>

Table 2. Simulation conditions and parameters for Geldart A particles on single process

<table>
<thead>
<tr>
<th>Particle</th>
<th>Gas</th>
<th>Geometry</th>
</tr>
</thead>
<tbody>
<tr>
<td>density ( \rho_p ) (kg·m(^{-3}))</td>
<td>930</td>
<td>density ( \rho_g ) (kg·m(^{-3}))</td>
</tr>
<tr>
<td>number ( N_p )</td>
<td>1×10(^6)</td>
<td>velocity ( u_g ) (m·s(^{-1}))</td>
</tr>
<tr>
<td>restitution coefficient</td>
<td>0.8</td>
<td>viscosity ( \mu_g ) (kg·m(^{-1})·s(^{-1}))</td>
</tr>
<tr>
<td>spring stiffness</td>
<td>800</td>
<td>time step ( \Delta t ) (s)</td>
</tr>
<tr>
<td>diameter ( d_p ) (mm)</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>time step ( \Delta t ) (s)</td>
<td>5×10(^{-7})</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Geometry</th>
<th>case a</th>
<th>case b</th>
<th>case c</th>
</tr>
</thead>
<tbody>
<tr>
<td>height ( H ) (m)</td>
<td></td>
<td>0.5</td>
<td>0.1</td>
</tr>
<tr>
<td>width ( W ) (m)</td>
<td>0.25</td>
<td></td>
<td>0.01</td>
</tr>
<tr>
<td>grid number</td>
<td></td>
<td>1×10(^5)</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Simulation conditions and parameters for a quasi-3D fluidization system

<table>
<thead>
<tr>
<th>Particle</th>
<th>Gas</th>
<th>Geometry</th>
</tr>
</thead>
<tbody>
<tr>
<td>density ( \rho_p ) (kg·m(^{-3}))</td>
<td>2500</td>
<td>density ( \rho_g ) (kg·m(^{-3}))</td>
</tr>
<tr>
<td>number ( N_p )</td>
<td>1.6×10(^6)</td>
<td>velocity ( u_g ) (m·s(^{-1}))</td>
</tr>
<tr>
<td>restitution coefficient</td>
<td>0.9</td>
<td>viscosity ( \mu_g ) (kg·m(^{-1})·s(^{-1}))</td>
</tr>
<tr>
<td>spring stiffness</td>
<td>1000</td>
<td>time step ( \Delta t ) (s)</td>
</tr>
<tr>
<td>diameter ( d_p ) (mm)</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>time step ( \Delta t ) (s)</td>
<td>1×10(^{-5})</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Geometry</th>
<th>case a</th>
<th>case b</th>
<th>case c</th>
</tr>
</thead>
<tbody>
<tr>
<td>height ( H ) (m)</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>width ( W ) (m)</td>
<td>0.25</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>depth ( W ) (m)</td>
<td></td>
<td></td>
<td>0.01</td>
</tr>
<tr>
<td>grid number</td>
<td></td>
<td>1×10(^4)</td>
<td></td>
</tr>
</tbody>
</table>
Table 4. Simulation conditions and parameters for an industrial-scale fluidized bed

<table>
<thead>
<tr>
<th>Particle</th>
<th>Fluid</th>
<th>Geometry</th>
</tr>
</thead>
<tbody>
<tr>
<td>particle density (kg·m⁻³)</td>
<td>density ρₚ (kg·m⁻³)</td>
<td>height H (m)</td>
</tr>
<tr>
<td>1500</td>
<td>4.72</td>
<td>30.5</td>
</tr>
<tr>
<td>cluster density (kg·m⁻³)</td>
<td>velocity uₚ (m·s⁻¹)</td>
<td>width W (m)</td>
</tr>
<tr>
<td>750</td>
<td>4.0</td>
<td>0.9~2.9</td>
</tr>
<tr>
<td>particle diameter (mm)</td>
<td>viscosity μₚ (kg·m⁻¹·s⁻¹)</td>
<td>grid number</td>
</tr>
<tr>
<td>0.0665</td>
<td>1.75×10⁻⁵</td>
<td>35380</td>
</tr>
<tr>
<td>cluster diameter (mm)</td>
<td>time step ∆t (s)</td>
<td></td>
</tr>
<tr>
<td>5.0</td>
<td>1×10⁻⁵</td>
<td></td>
</tr>
<tr>
<td>cluster number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>115607</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cluster restitution coefficient</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cluster spring stiffness</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>time step ∆t (s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1×10⁻⁵</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5. Simulation conditions and parameters of Geldart A particles on multi processes

<table>
<thead>
<tr>
<th>Particle</th>
<th>Gas</th>
<th>Geometry</th>
</tr>
</thead>
<tbody>
<tr>
<td>density ρₚ (kg·m⁻³)</td>
<td>density ρ₉ (kg·m⁻³)</td>
<td>height H (m)</td>
</tr>
<tr>
<td>2450</td>
<td>1.29</td>
<td>5.0</td>
</tr>
<tr>
<td>number Nₚ</td>
<td>velocity u₉ (m·s⁻¹)</td>
<td>width W (m)</td>
</tr>
<tr>
<td>1×10⁷</td>
<td>3.546</td>
<td>0.1</td>
</tr>
<tr>
<td>restitution coefficient</td>
<td>viscosity μ₉ (kg·m⁻¹·s⁻¹)</td>
<td>grid number</td>
</tr>
<tr>
<td>0.8</td>
<td>1.8×10⁻⁵</td>
<td>2×10⁶</td>
</tr>
<tr>
<td>spring stiffness</td>
<td>time step ∆t (s)</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>5×10⁻⁵</td>
<td></td>
</tr>
<tr>
<td>diameter dₚ (mm)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0824</td>
<td></td>
<td></td>
</tr>
<tr>
<td>time step ∆t (s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5×10⁻⁷</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
- A multi-scale CPU-GPU hybrid computation mode for discrete particle simulation.
- Significant time speedup by using graphic processing units (GPUs).
- Fluidization behavior revealed in simulations with unprecedented details.
- Preliminary meso-scale model for industrial-scale discrete particle simulation.