Particuology 7 (2009) 317–323

Contents lists available at ScienceDirect

Particuology

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

Parallelization of pseudo-particle modeling and its application in simulating gas–solid fluidization

Jianxin Lu[a,b], Jiayuan Zhang[b,*], Xiaowei Wang[b], Limin Wang[a,b], Wei Ge[b,*]

[a] Graduate School of Chinese Academy of Sciences, Beijing 100049, China
[b] State Key Laboratory of Multiphase Complex Systems, Institute of Process Engineering, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 14 January 2009
Accepted 22 April 2009

Keywords:
Parallelization
Pseudo-particle modeling
Gas–solid fluidization
Dynamic load balancing

ABSTRACT

Pseudo-Particle Modeling (PPM) is a particle method proposed by Ge and Li in 1996 [Ge, W., & Li, J. (1996). Pseudo-particle approach to hydrodynamics of particle–fluid systems. In M. Kwauk & J. Li (Eds.), *Proceedings of the 5th international conference on circulating fluidized bed* (pp. 260–265). Beijing: Science Press] and has been used to explore the microscopic mechanism in complex particle–fluid systems. But as a particle method, high computational cost remains a main obstacle for its large-scale application; therefore, parallel implementation of this method is highly desirable. Parallelization of two-dimensional PPM was carried out by spatial decomposition in this paper. The time costs of the major functions in the program were analyzed and the program was then optimized for higher efficiency by dynamic load balancing and resetting of particle arrays. Finally, simulation on a gas–solid fluidized bed with 102,400 solid particles and 1.8×10^7 pseudo-particles was performed successfully with this code, indicating its scalability in future applications.

1. Introduction

With the limitation of experimental technology in exploring micro- and meso-scale mechanisms in chemical reactors and the fast development of computer capacity, numerical simulation has attracted more and more attention in the research of process engineering (Li et al., 2005). In recent years, discrete simulation has become a hot point in the research of particle–fluid systems. Pseudo-Particle Modeling (PPM) is a particle method (Ge & Li, 1996) that discretizes the fluid into a large number of smooth and rigid pseudo-particles, which are much smaller than the real particles but larger than molecules or atoms, as shown in Fig. 1. With this approach, the computation of fluid dynamics is transformed into a series of collisions among pseudo-particles (PPs), and thereby facilitating and unifying the interaction algorithms of three kinds of collisions among solid particles (SP) and pseudo-particles, i.e., SP–SP, PP–SP and PP–PP.

In PPM, each pseudo-particle has four properties: mass (m), radius (r), position (P) and velocity (v), among which m and r are constant in the whole process. In the simulation process, all the particles move synchronously in the same time step (dt). In each time step, all particles move independently, possibly under some external forces. At the end of each step, if the distance between the centers of two particles $|\boldsymbol{P}_1 - \boldsymbol{P}_2|$ is less than the sum of their radii $(r_1 + r_2)$, and they are moving towards each other, they will collide as two rigid and smooth particles. The new velocities after collision are as follows:

$$\boldsymbol{v}_1^* = \boldsymbol{v}_1 - \frac{2m_2}{m_1 + m_2} \frac{(\boldsymbol{v}_1 - \boldsymbol{v}_2)(\boldsymbol{P}_2 - \boldsymbol{P}_1)}{|\boldsymbol{P}_2 - \boldsymbol{P}_1|^2} (\boldsymbol{P}_2 - \boldsymbol{P}_1), \quad (1)$$

$$\boldsymbol{v}_2^* = \boldsymbol{v}_2 + \frac{2m_1}{m_1 + m_2} \frac{(\boldsymbol{v}_1 - \boldsymbol{v}_2)(\boldsymbol{P}_2 - \boldsymbol{P}_1)}{|\boldsymbol{P}_2 - \boldsymbol{P}_1|^2} (\boldsymbol{P}_2 - \boldsymbol{P}_1), \quad (2)$$

where the indexes 1 and 2 mean the two different particles and v^* means velocity after collisions.

Many works have been done since PPM was proposed. For instance, PPM has successfully reproduced the "long time tail" phenomena at high gas concentrations (Ge & Li, 2003); it has also been used to simulate some classical flow such as plane Poiseuille flow and flow around static solid particles, with reasonable results (Ge, 1998; Ge & Li, 2003), indicating its feasibility and accuracy in discrete simulation of single-phase flows.

As for the application in particle–fluid systems, Ge and Li (2003) also simulated typical phenomena in fluidization, including clustering, slugging and bubbling, where periodic boundary condition was used due to limitations on computation capacity and hence on simulation scale in earlier times (Ge, 1998).

* Corresponding authors. Tel.: +86 10 82616050; fax: +86 10 62558065.
E-mail addresses: jyzhang@home.ipe.ac.cn (J. Zhang), wge@home.ipe.ac.cn (W. Ge).

1674-2001/$ – see front matter © 2009 Chinese Society of Particuology and Institute of Process Engineering, Chinese Academy of Sciences. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.partic.2009.04.003

Nomenclature

d_p	pseudo-particle diameter, m
d_s	solid particle diameter, m
e_g	pseudo-particle restitution coefficient
e_s	solid particle restitution coefficient
e_w	boundary restitution coefficient
g	gravity acceleration, m/s^2
H	flow field height, m
n_p	pseudo-particle number
n_s	solid particle number
Re	Reynolds number
U_g	gas flow velocity, m/s
U_s	solid flow velocity, m/s
v_0	thermal velocity, m/s
W	flow field width, m
ε	voidage

Zhang et al. upgraded the algorithms of PPM and used it to simulate more realistic gas–solid systems (Ge, Zhang, Li, & Li, 2003; Zhang, Ge, & Li, 2004). Considering that a real system was connected to the outside, the periodic boundary condition was replaced by an open boundary condition, where the pressure and temperature of inlet and outlet can be controlled separately. Simulation of a fluidized bed with 2500 solid particles and 4.5×10^5 pseudo-particles successfully demonstrated the evolution process of heterogeneous structures (Zhang, 2004). Another more significant application of PPM is the verification of the stability criterion, $N_{st} = \min$, i.e., the stability condition of the energy-minimization multi-scale (EMMS) model (Zhang, Ge, & Li, 2005). Before 2004, all simulations with PPM were executed by serial computing.

Since the simulation capacity is much limited in serial computing with single CPU and the scale is still much smaller than a real system, it is but natural to develop a parallel algorithm of this method. The first attempt was made in 2003 by Zhang and Wang (Wang et al., 2005; Zhang, 2004) with 1-D spatial decomposition and dynamic load balancing. Due to the sequential and non-additive nature of the hard-sphere model, parallelization is more complicated as compared to traditional soft-sphere models. In this paper, a 2-D spatial decomposition algorithm of PPM for parallel simulation of gas–solid fluidization was carried out based on Zhang and Wang's work, and then computation efficiency was optimized by analysis of the time consumption of the main operation functions before a final demonstration on a gas–solid system with 102,400 solid particles and 1.8×10^7 pseudo-particles.

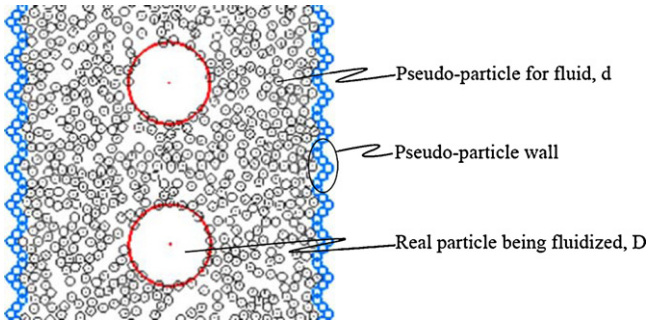

Fig. 1. Schematic diagram of a continuous fluid that is discretized into pseudo-particles (Zhang et al., 2005).

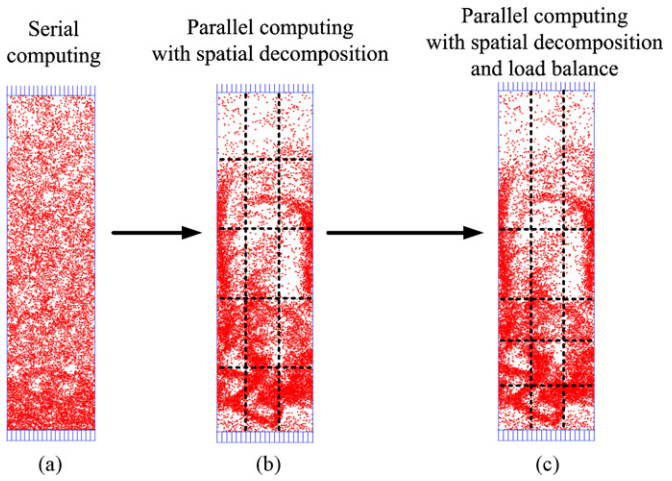

Fig. 2. Spatial decomposition and dynamic load balancing.

2. Parallelization of hard-sphere PPM

2.1. Spatial decomposition

In parallel computing, the key problem is to divide a large computation task into a certain number of parts which can be executed on independent CPUs simultaneously with some necessary communications at some intervals. In this way, the simulation scale can be enlarged while the increment of time consumption is endurable. Generally, there are three decomposition algorithms: Atom Decomposition, Force Decomposition and Spatial Decomposition (Plimpton, 1995). As the forces in PPM are short-range and our application objective is mainly fluidization system with a rather regular geometry, spatial decomposition was adopted in this work.

Since the fluidized bed is simplified as a rectangle in our 2-D simulation, and the particles only have collisions with adjacent particles due to the hard-sphere algorithm, the flow field is partitioned into sub-spaces along width and height (as shown in Fig. 2) and each sub-region is assigned to a unique processor, which deals with the calculation task of the particles belonging to its respective domain. Certainly, the processors must communicate but only with the four adjacent processors, i.e., the upper and the lower processors, and the left and right processors.

Compared to serial computation, a task assigning process is needed before parallel computation. That is, all particles (including SPs and PPs) are designated to a processor beforehand according to their initial positions. When a particle moves out of its sub-domain into a neighboring sub-domain, deleting and adding operation would be necessarily implemented in the corresponding processors.

2.2. Parallelization algorithm

Fig. 3 shows the basic flowchart of parallel algorithm for gas–solid fluidized bed together with the major functions (or steps) of the program. Different from serial computing, special treatments (e.g. those highlighted by red italic in Fig. 3) need to be done for parallel computing, and even more, for parallel algorithm in hard-sphere model.

In the initialization progress, the uniform distributions and random velocities are given to both PPs and SPs. For parallel computing, spatial decomposition also needs to be done here and both SPs and PPs are stored in their linked lists in each processor.

During the computing period, in each step the PPs and SPs first move under the external forces and their displacements are calcu-