Powder Technology 333 (2018) 219–228

Contents lists available at ScienceDirect

Powder Technology

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

A GPU-based DEM for modelling large scale powder compaction with wide size distributions

Y. He [a], T.J. Evans [b], A.B. Yu [c], R.Y. Yang [a,*]

[a] School of Materials Science and Engineering, University of New South Wales, Sydney 2052, Australia
[b] Rio Tinto Iron Ore Group, Australia
[c] Department of Chemical Engineering, Monash University, Clayton 3900, Australia

ARTICLE INFO

Article history:
Received 16 November 2017
Received in revised form 7 April 2018
Accepted 15 April 2018
Available online 17 April 2018

Keywords:
Discrete element method
GPU
CUDA
Parallel computing
Powder compaction

ABSTRACT

In the present study, we developed a GPU-based discrete element method (DEM) to tackle the challenges associated with modelling powder compaction, in particular large scale systems with wide size distributions. In the model, a multi-grid searching method specifically designed within the GPU architecture was proposed for particle neighbour searching. A memory layout was designed to ensure coalesced memory access for neighbour list and associated contact history. The proposed GPU implementation was able to achieve a three-level parallelism, from single GPU to GPUs within a computing node and to GPUs across nodes. The model was applied to powder compaction and the simulation results showed significant gain in computational efficiency and reliable prediction of the compaction behaviour.

1. Introduction

The discrete element method (DEM) is an effective tool to study particle behaviour [1]. It can generate abundant information at particle scale, such as the trajectories of and forces acting on individual particles, which is critical to understanding the underlying mechanisms of various applications [2]. Recently, we have conducted a DEM study of the mechanisms of densification and consolidation in compaction and the mechanical properties of formed compacts [3,4].

There are, however, a few challenges in simulating large scale compaction due to high computational cost of the DEM [5,6]. Firstly, a compaction process includes a number of stages including mixing, die filling, die compression and unloading [7,8]. A small strain rate is required to keep a quasi-static state, leading to very long simulation time [9]; Secondly, the complex contact behaviour between particles, such as elastic-plastic deformation, adhesive force and formation of inter-particle bonding, requires intensive computation with large amount of memory [10,11]. Other factors, such as wide size distributions and large number of particles, make modelling real systems even more challenging [12,13]. While various methods have been developed to accelerate DEM simulation, such as reducing particle stiffness [14,15], increasing particle size [5,16] (coarse grain model), introducing cut-off distance for long-ranged forces [17] and density scaling [9,18], each method has its own limitations. Therefore, developing a general DEM model capable of accelerating simulation while handling various aspects specific to powder compaction is of great importance for both fundamental research and industrial practice.

Recently, parallelization based on the Graphic Processing Unit (GPU) is gaining popularity due to its significant performance on floating point operations. Applications of the GPU techniques to the various particle-based numerical methods have been reported, such as Molecular Dynamics [19], Smoothed Particle Hydrodynamics [20,21] and Lattice Boltzmann Method [22]. The DEM scheme is particularly well suited for using the GPU parallelisation due to its locality of particle interactions [23]. In recent years, GPU-based DEM has been increasingly applied to various industrial fields, such as milling, mixing and fluidization [23–28]. To the best of our knowledge, simulating powder compaction on GPU has yet to be reported.

For particle-based methods, neighbour searching is often the most time-demanding aspect in calculation. Developing efficient searching algorithm is thus crucial to the overall performance, especially for a polydispersed particle system. Commonly used neighbour searching methods include spatial sorting and spatial subdivision. The spatial sorting method requires an efficient sorting algorithm to maintain particles according to their coordinates, thus not efficient for large scale simulations while the spatial subdivision method partitions the computational domain into orthogonal cells, such as linked-cell method [29]. However, the linked-cell method suffers from the difficulties of handling polydispersed system due to the fact that the cell size of the searching grid need to conform to the largest particle size. This problem can be effectively addressed by multigrid searching

* Corresponding author.
E-mail address: r.yang@unsw.edu.au. (R.Y. Yang).

https://doi.org/10.1016/j.powtec.2018.04.034
0032-5910/© 2017 Elsevier B.V. All rights reserved.

methods. The basic idea of multigrid methods is to map particles onto different level of grids according to particle size and then conduct cross-level neighbour searching. Different multigrid methods have been proposed, which differ in the way of conducting neighbour searching between different grid levels, either in the coarse grid [30–32] or in the fine grid [29,33]. Multigrid searching methods is robust and can be easily paralleled on the distributed memory systems. For example, He et al. [30] proposed a multigrid method which is shown to be insensitive to the distribution of particle size and later addressed the problem of load balancing among processes by estimating the amount of calculation [34]. Ogarko et al. [33,35] discussed the choice of optimal cell sizes and number of grid levels to achieve best performance. On the GPU platform, to address diversity of particle size, Zheng et al. [24] adopted a uniform grid in which the cell size can be smaller than the particles. For a given cell, all particles that overlap with it has to be identified and the total number of particles is determined using atomic operation. To date, however, implementation of multigrid searching method on GPU has not been reported, especially no discussion on memory management for systems with large size difference.

This work will address the limitations of system scales and large size ratios using the GPU techniques. An efficient multi-grid searching, specially designed within the GPU architecture, will be developed to handle the neighbour searching for particles with large size difference. The performance on single GPU and multi-GPUs will be evaluated by modelling the compaction of powders.

2. Model description and implementation

2.1. Governing equations

In DEM, the translational and rotational motions of a particle of mass m_i and moment of inertia $\mathbf{I}_\mathrm{i}$ are governed by Newton's second law of motion, given by,

$$\mathrm{m_i}\frac{d\mathbf{v}_\mathrm{i}}{dt} = \sum_\mathrm{j} \mathbf{F}_\mathrm{ij} + \mathrm{m_i}\mathbf{g} \tag{1}$$

$$\mathbf{I}_\mathrm{i}\frac{d\boldsymbol{\omega}_\mathrm{i}}{dt} = \sum_\mathrm{j} \mathbf{M}_\mathrm{ij} \tag{2}$$

where $\mathbf{v}_\mathrm{i}$ and $\boldsymbol{\omega}_\mathrm{i}$ are, respectively, the particle translational and rotational velocities. $\mathbf{g}$ is the gravitational acceleration. The inter-particle forces $\mathbf{F}_\mathrm{ij}$ include the normal contact force $\mathbf{F}_\mathrm{ij}^\mathrm{n}$, tangential contact force $\mathbf{F}_\mathrm{ij}^\mathrm{t}$, capillary force $\mathbf{F}_\mathrm{ij}^\mathrm{cap}$ for wet particles and bonding forces $\mathbf{F}_\mathrm{ij}^\mathrm{b}$ if present. $\mathbf{M}_\mathrm{ij}$ is the corresponding moment, including the moment $\mathbf{M}_\mathrm{ij}^\mathrm{t}$ caused by tangential force, $\mathbf{M}_\mathrm{ij}^\mathrm{r}$ the moment due to particle rolling friction and the moment $\mathbf{M}_\mathrm{ij}^\mathrm{b}$ induced by the tangential bonding force and bond bending/torsion. Fig. 1 shows the schematic illustration of forces between elemental spheres. In order to predict the compact strength, a bonded particle model is introduced to account for the effect of inter-particle bonding [3]. Table 1 lists the equations of the forces. More details can be found in our previous study [3].

2.2. GPU implementation

The GPU code was developed based on the NVIDIA CUDA platform which provides an efficient low level of hardware access. It was formulated according to a paradigm of single-program, multiple-data (SPMD), in which same program operates on different data sets simultaneously. Due to the discrete nature of particles, each particle was assigned with one CUDA thread so that neighbour searching and interaction with neighbours can be carried out independently for each particle. The DEM calculation was fully implemented on the GPU except the data were intermittently retrieved out of the GPU memory for data

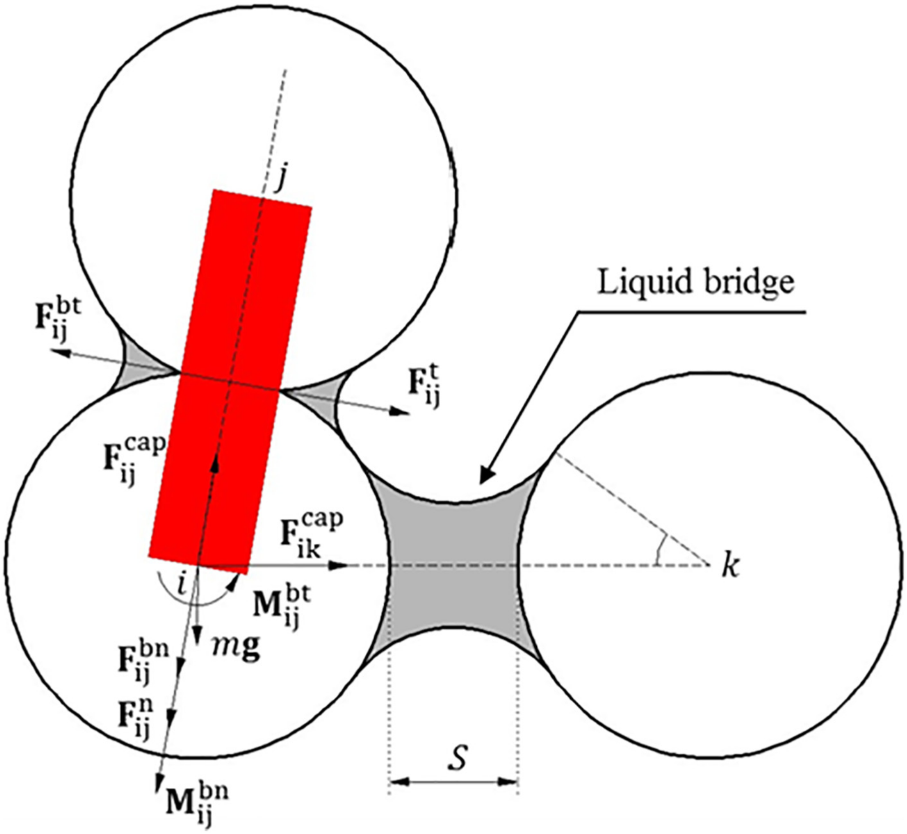
j
Liquid bridge
$\mathbf{F}_\mathrm{ij}^\mathrm{bt}$
$\mathbf{F}_\mathrm{ij}^\mathrm{t}$
$\mathbf{F}_\mathrm{ij}^\mathrm{cap}$
$\mathbf{F}_\mathrm{ik}^\mathrm{cap}$
k
i
$\mathbf{M}_\mathrm{ij}^\mathrm{bt}$
mg
$\mathbf{F}_\mathrm{ij}^\mathrm{bn}$
$\mathbf{F}_\mathrm{ij}^\mathrm{n}$
S
$\mathbf{M}_\mathrm{ij}^\mathrm{bn}$

Fig. 1. Schematic of the forces acting on sphere *i* from contacting sphere *j* and pendular liquid-bridge linked sphere *k*.