IV International Conference on Particle-based Methods – Fundamentals and Applications
PARTICLES 2015
E. Oñate, M. Bischoff, D.R.J. Owen, P. Wriggers & T. Zohdi (Eds)

VALIDATION OF THE GPU BASED BLAZE-DEM FRAMEWORK FOR HOPPER DISCHARGE.

Nicolin Govender[1], Patrick Pizette[2] AND Daniel N Wilke[3] AND Nor-edine Abriak[2]

[1] Advanced Mathematical Modeling CSIR
Pretoria, 0001 South Africa
ngovender1@csir.co.za

[2] Mines Douai
LGCgE GCE, F-59508 Douai, France
patrick.pizette@mines-douai.fr

[3] University of Pretoria, Department of Mechanical and Aeronautical Engineering
Pretoria, 0001, South Africa
nico.wilke@up.ac.za

Key words: Hoppers, Silos, Large scale DEM, GPU, Polyhedra

Abstract. Understanding the dynamical behavior of particulate materials is extremely important to many industrial processes, with typical applications that range from hopper flows in agriculture to tumbling mills in the mining industry. The discrete element method (DEM) has become the defacto standard to simulate particulate materials. The DEM is a computationally intensive numerical approach that is limited to a moderate amount (thousands) of particles when considering fully coupled densely packed systems modeled by realistic particle shape and history dependent constitutive relationships. A large number (millions) of particles can be simulated when the coupling between particles is relaxed to still accurately simulated lesser dense systems. Massively large scale simulations (tens of millions) are possible when particle shapes are simplified, however this may lead to oversimplification when an accurate representation of the particle shape is essential to capture the macroscopic transport of particulates. Polyhedra represent the geometry of most convex particulate materials well and when combined with appropriate contact models predicts realistic mechanical behavior to that of the actual system. Detecting collisions between polyhedra is computationally expensive often limiting simulations to only hundreds of thousands of particles. However, the computational architecture e.g. CPU and GPU plays a significant role on the performance that can be realized. The parallel nature of the GPU allows for a large number of simple independent processes to be executed in parallel. This results in a significant speed up over conventional implementations utilizing the Central Processing Unit (CPU) architecture, when algorithms are well aligned and optimized for the threading model of the GPU. We recently introduced the BLAZE-DEM framework for the GPU architecture that can model millions of spherical and polyhedral particles in a realistic time frame using a single GPU. In this paper we validate BLAZE-DEM for hopper discharge simulations. We firstly compare the flow-rates and patterns of polyhedra and spheres obtained with experiment to that of DEM. We then compare flow-rates between spheres and polyhedra to gauge the effect of particle shape. Finally we perform a large scale DEM simulation using 16 million particles to illustrate the capability of BLAZE-DEM to predict bulk flow in realistic hoppers.

1. Introduction

1.1. Hoppers

Simulating the dynamics of particulate materials is critical in the design and optimization of many industrial processes. The hopper configuration is one of the most studied configurations, either to increase the fundamental knowledge about the dynamics of granular flows or to design storage devices. Experimental, empirical [1], continuous [2] and discrete models [3] are typical approaches used to study either the granular intrinsic properties or the handling of granular media. The behavior of granular media is complex. For example, when a hopper is discharged by gravity the flow rate does not depend on the height of the material. Furthermore when the height of the material is ≥ 3.5 times the hydraulic radius of the silo, the pressure at the bottom saturates due to the Janssen effect, resulting in a flow rate that remains fairly constant. The design of the hopper and the stored material influence directly the flow rate and flow pattern. In particular, the hopper angle and the outlet size are calculated to allow bulk solid flow and prevent arch formation which restricts flow.

1.2. DEM

The discrete element method (DEM), which was first described by Cundall and Strack in 1979 [4], is one of the most successful discrete methods to simulate particulate materials. The DEM was originally developed for solving problems in geotechnical engineering, but has been employed to model particulate materials in a variety of fields [5, 6].

The DEM requires that all particles in the system need to be checked for contact at each time step. This involves a considerable number of calculations depending on the particle geometry and the number of particles [7] in the system. To reduce the computational cost, particle shape is often approximated by a sphere (Figure 1(a)). This approximation however may result in the model exhibiting unrealistic mechanical behavior, as discussed by Latham and Munjiza [8]. The clumped-sphere (Figure 1(b)) approximation [9] provides a better description of shape by using a number of spheres to represent a particle. However, this approach is limited in the number of particles and introduces non-physical artifacts into the simulation, as discussed by Horner [10]. Polyhedral shaped particles, depicted in Figure 1(c), can capture details in particle shape well and hence exhibit realistic mechanical behavior to that of the actual system [11, 12]. However, the number of polyhedral particles that can be simulated on typical workstation computers in a realistic time frame is limited, as discussed by Mack et al. [13], in which only 322 polyhedra are simulated. This limitation is due primarily to complex collision detection and storage requirements of polyhedra.

The developed BLAZE-DEM code [14, 15] aims to address the computational limitations of modeling polyhedra, by efficiently implementing polyhedra contact algorithms on the GPU architecture. This paper gives a brief overview of the BLAZE-DEM framework and dedicates the rest of the effort to validate the utility of BLAZE-DEM to model hopper discharge for polyhedral and spherical particles.

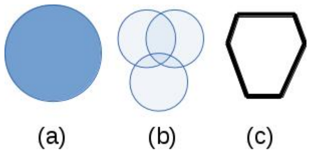


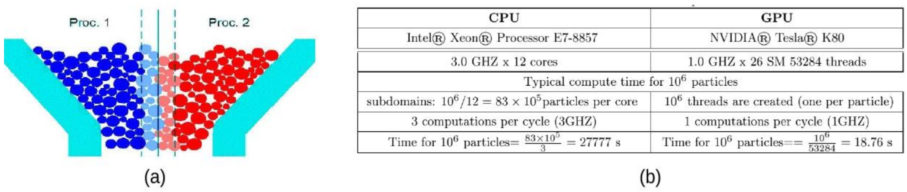
Figure 1: (a) Single sphere, (b) clumped sphere and (c) polyhedron representations of a corn kernel.

Once the particles that are in contact is determined, the resultant forces acting on the particles can be calculated. As with all numerical simulations there is a trade-off between model accuracy and computational speed. In [16, 17] the number of particles that can be modeled is limited due to the large computational cost of the associated FEM simulations and the solution of the coupled dynamic equilibrium problem associated with multiple bodies being simultaneously in contact. This allows for the simulation of strongly coupled problems associated with the compaction of particles, in addition to weakly coupled problems found in typical bulk flow analysis. The BLAZE-DEM framework focuses only on typical bulk flow problems that allows for some assumptions to be made without significant loss of accuracy but drastically benefit the computational efficiency of the numerical model. We only consider binary contact to resolve the associated contact forces on particles. In addition we consider the original contact force model proposed by Cundall et al. [17]. The constitutive model is described by a spring-dashpot coupled in parallel, which offers an appropriate model fidelity for the problems under consideration. The model is computationally efficient. This is in contrast to the computational expense of the higher fidelity FEM-DEM model [16]. Once the forces are calculated an explicit integration scheme is used to determine the resultant motion of all particles in the system.

1.3. Computational Implementation

Since DEM computations are compute bound, the number and complexity of simulated particles has scaled with increased computational power over the past three decades. In the last few years the trend of increasing Central Processing Unit (CPU) clock speed has stopped due to the physical limits on the materials used in the manufacturing of computer hardware. While computational power still scales with Moore's Law, this scaling is now achieved through increasing the number of computing cores on a single chip as opposed to make a single core faster. Leading this evolution from multi-core to many core processing chips is the Graphical Processor Unit (GPU). Designed to convey information to screen that has millions of pixels which need to be updated simultaneously to render visual information to the user, the GPU is a massively parallel processor that can perform billions arithmetic operations in parallel (7.52 TFLOPS). The NVIDIA developed CUDA programming model [18] provides access to the GPU from a variety of high level programming languages such as C++, Java and Python.

An emerging trend in the past few years is the implementation of scientific and engineering solutions on GPU's [19, 20]. Utilizing the GPU for computation over the conventional GPU offers significant speed-ups when computations are well aligned for the parallel threading model of the GPU. Parallelism on the CPU is limited to domain decomposition [21] as depicted in Figure 2(a) [22]. The GPU however is a many core processor enabling parallelism at a particle level, with each particle having its own thread. Figure 2(b) depicts the theoretical performance of a Xeon CPU and Tesla GPU for task of performing computations for 10 millions particles. The CPU can only launch 12 threads (Particles are processed in a serial loop for each thread), while the GPU can launch 53284 threads per cycle. This gives the GPU an enormous edge over the CPU in DEM calculations which are data parallel resulting in a speed up of 500 when taking into consideration cost and power consumption.



(a)

CPU	GPU
Intel® Xeon® Processor E7-8857	NVIDIA® Tesla® K80
3.0 GHZ x 12 cores	1.0 GHZ x 26 SM 53284 threads
Typical compute time for 10^6 particles	
subdomains: $10^6/12 = 83 \times 10^5$particles per core	10^6 threads are created (one per particle)
3 computations per cycle (3GHZ)	1 computations per cycle (1GHZ)
Time for 10^6 particles= $\frac{83\times10^5}{3}$ = 27777 s	Time for 10^6 particles== $\frac{10^6}{53284}$ = 18.76 s

(b)

Figure 2: (a) Domain decomposition on CPUs [22] and (b) Comparison of CPU and GPU parallel solutions.

There has been little development in terms of simulations based on the DEM using the GPU from the both academic and commercial sectors. Although the GPU is an ideal match for certain DEM simulations the current learning curve associated with GPU development is high as the technology is fairly new compared to the traditional CPU platform. Furthermore only an efficiently implemented GPU DEM solution will yield significant performance benefits over the CPU.

The DEM code used in this paper is built on the BLAZE-DEM GPU framework developed by the authors [14]. The framework and its associated algorithms [15] has thus far been successfully used in the simulation of tumbling mills [23]. BLAZE-DEM can model (i) tens of millions of spherical particles and (ii) millions of polyhedral particles in a realistic time frame on a desktop computer using a single GPU. To the best of the author's knowledge there are no published works on large scale simulations of hoppers using the GPU.

2. OVERVIEW BLAZE-DEM CONTACT MODEL

A linear spring dash-pot model is used to calculate the normal force between particles given by :

$$\mathbf{F}_{\mathrm{N}} = (\mathrm{K_n}\delta)\bar{\mathbf{n}} - \mathrm{C_n}(\mathbf{V}_{\mathrm{R}} \cdot \bar{\mathbf{n}})\bar{\mathbf{n}}, \tag{1}$$

where δ is the penetration depth, $\mathbf{V}_{\mathrm{R}} = \mathbf{V}_1 - \mathbf{V}_2$ is the relative translational velocity, $\mathrm{K_n} = \frac{\mathrm{m_{eff}}}{\mathrm{t^2_{contact}}} \ln(\epsilon)^2 + \pi^2$ is the spring stiffness, $\mathrm{C_n} = \frac{2\ln(\epsilon)\sqrt{\mathrm{K_n m_{eff}}}}{\sqrt{\ln(\epsilon)^2+\pi^2}}$ is the viscous damping coefficient, $\bar{\mathbf{n}}$ the normal at contact, ϵ is the coefficient of restitution and $\mathrm{m_{eff}} = (\frac{1}{\mathrm{m_1}} + \frac{1}{\mathrm{m_2}})^{-1}$ is the effective mass of the particles. The contact time $\mathrm{t_{contact}}$ is determined by the properties of the material. However in most cases experimental data is not readily available for a particular material. For such cases $\mathrm{K_n}$ is chosen such that that physical quantities of interest (such as energy) are conserved during integration for the typical range of velocities observed in a simulation [15]. Typical DEM simulations use a spring stiffness and time-step that limits the maximum penetration depth to $\delta_{max} \leq 0.05\mathrm{r}$ where r is the radius of the smallest particle [19, 24, 25].

Tangential Contact. A stick-slip columbic model [15, 19, 25] is used to calculate the tangential force magnitude between particles given by :

$$\mathrm{F_T} = -\min\left[\mu \parallel \mathbf{F}_{\mathrm{N}} \parallel, \frac{\min\left[\min\left[\,\|\mathbf{V}_{1\mathrm{T}}\|\,,\|\mathbf{V}_{2\mathrm{T}}\|\right],\, \mu\,\|\mathbf{V}_{\mathrm{T}}\|\right]\mathrm{m_{eff}},}{\triangle \mathrm{t}}\right], \tag{2}$$

where $\mathbf{V}_{1\mathrm{T}}$ and $\mathbf{V}_{2\mathrm{T}}$ are the tangential velocities of each particle,$\mathbf{V}_{\mathrm{T}} = (\mathbf{V}_{\mathrm{R}} - (\mathbf{V}_{\mathrm{R}}.\bar{\mathbf{n}})\bar{\mathbf{n}})$ is the relative tangential velocity and μ the coefficient of friction.

Angular motion. In addition to translation forces a particle also experiences a torque as a result of contact given by :

$$\mathbf{\Gamma} = (\mathbf{r} \times \mathbf{F}_{\mathrm{N}}) \tag{3}$$

where $\mathbf{r}$ is the vector from the center of mass to the contact point $\mathbf{PC}(x, y, z)$. A detailed description of the models and numerical techniques used in BLAZE-DEM can be found in [14, 15, 23]. In a DEM simulation the model parameters are chosen to either match experimental results or to reproduce a desired behavior. Tuning these parameters is a tedious task with the plethora of different models used in DEM simulations. For DEM to be useful in hopper simulations we need to be able to accurately capture the effect of geometrical changes in the hopper using the same model parameters. In this paper we use the experimentally determined properties.

3. Experimental Setup

In order to verify the suitability of the GPU based BLAZE-DEM code for hopper simulation, we performed experiments using lab-scale plexi-glass hoppers at three discharge angles ($\beta = 30^o, 60^o, 90^o$) as depicted in Figure 3(a). We printed 2000 regular dodecahedron and spherical particles as depicted in Figure 3(b) and packed them in alternating colors (Figure 3(a)) to observe the flow patterns. The hopper is filled by dropping particles down the center in order to obtain a random loose packing (three tests we performed for each discharge angle). The friction coefficients between particles and particles with the boundaries are evaluated by a dedicated started angle experiment similar to that of Abriak et al. [26]. Note that for spheres, the test consists of a clump of three spheres to prevent rolling. The average frictional values was found to be $\mu_{particle} = 0.35$ and $\mu_{wall} = 0.30$.

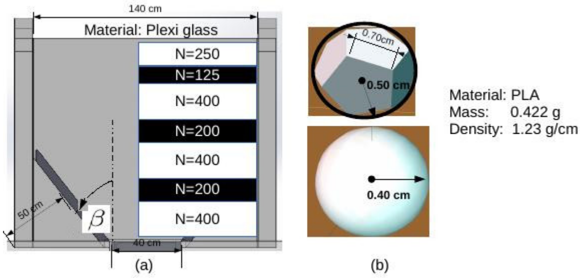


Figure 3: (a) Experimental setup indicating the number of particles N in each layer, discharge angle beta and dimensions of the inlet and outlet. (b) particle specifications.

4. Comparisons of results

Figure 4(a)-(c) shows the flow rates with polyhedra for DEM and experiment while Figure 4(d)-(f) shows the flow rates with spheres for DEM and experiment. We see very good agreement with both spheres and polyhedra against experimental results. We notice that the fastest flow is for the case of $\beta = 30^o$ as it has the steepest inclination, with $\beta = 90^o$ not discharging completely as expected for both spheres and polyhedra. The model parameters used are : $\epsilon = 0.4$ $\mathrm{K_n} = 3.2 \times 10^5\mathrm{N.cm}^{-1}$ for a time-step of 1×10^{-5}.

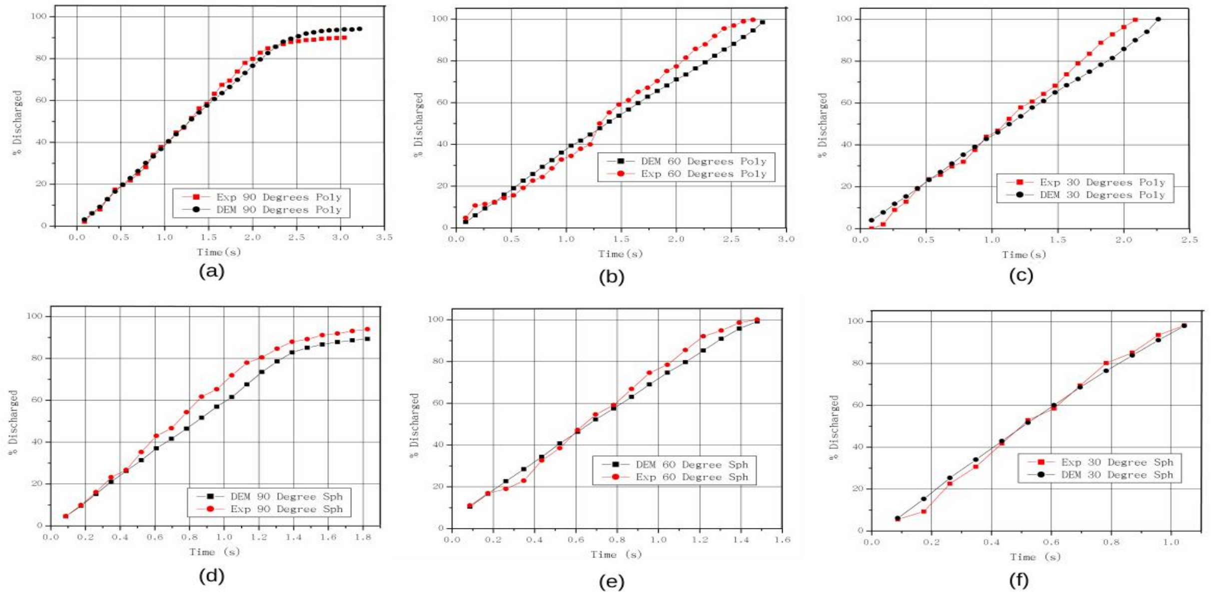


Figure 4: (a-c) Flow rates of polyhedra and (d-f) flow rates for spheres.

Figure 5 shows the flow patterns for the three hopper configurations with polyhedral particles. We notice that for 90 degrees we have funnel flow with the particles in the center discharging first and those in the sides last, while we have mass flow for both 60 and 30 degrees.

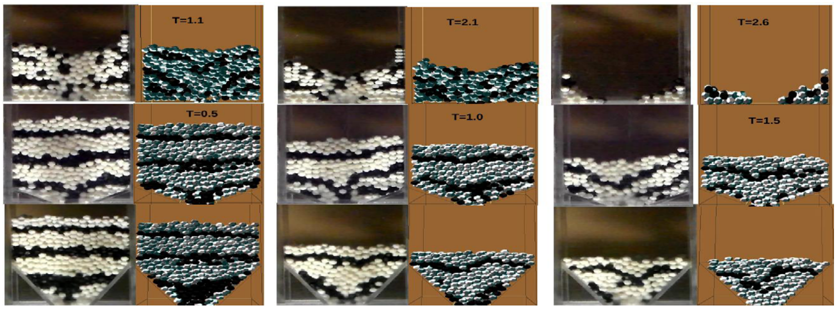
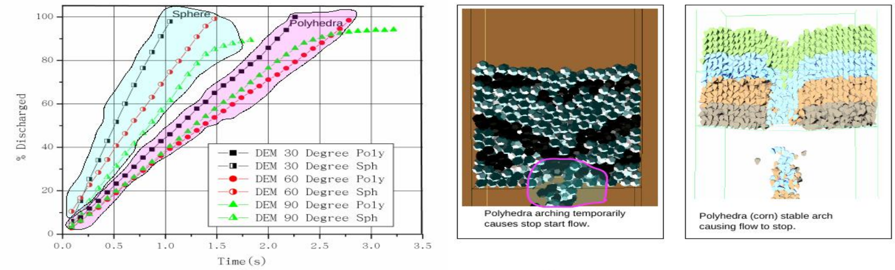


Figure 5: Flow patterns for polyhedra with experiment and DEM (time in seconds).

4.1. Effect of particle shape

To ascertain the effect of particle shape we used spheres that had the same mass and material properties of the polyhedra as depicted in Figure 3(a). Figure 6 shows the DEM predicted flow rates between spheres and polyhedra. We notice that the spheres flow faster, which can be attributed to polyhedra being able to rotate and pack more efficiently as depicted in Figure 6(b) and (c). We also notice that with polyhedra the flow-rate for the hopper angle of $\beta = 60^o$ is slower than that of $\beta = 90^o$, which is opposite to what we observe with spheres. While it is possible to increase friction (rolling and tangential) for spheres to obtain a better match, the trends between different geometries is clearly dependent on particle shape.



(a) (b) (c)

Figure 6: (a) Combined flow rates for polyhedra and spheres, (b) transient arch forming for polyhedra flow, (c) stable arch for polyhedral simulation of corn.

4.2. Large-scale simulations

The parallel computational efficiency of the BLAZE-DEM framework on the GPU architecture allows us to create simulations that are closer to reality by increasing the number of particles we can simulate in a shorter time frame as depicted in Table 1.

Table 1: Comparison to other codes (mono sized particles) *Kepler class GPU.

Author	**Shape**	**Physics Fidelity**	**N particles**	**C Number**
Harida et.al [20]	Clumped	Low	1.64×10^4	0.66×10^6
Longmore et.al [19]	Clumped	High	2.56×10^5	1.49×10^6
Radake et.al [27]	*Sphere	High	20×10^6	20×10^6
Nvidia SDK (2014) [28]	*Sphere	Low	2.5×10^5	125×10^6
BLAZE-DEM [23]	*Sphere	High	60×10^6	100×10^6
Note: No published GPU/CPU parallel polyhedra codes				Compute Time (N=5×10^5)
BLOCKS [29]	Poly^{cpu}	Highest	5×10^3	186 days
iDEM [29]	Poly^{cpu}	Low	5×10^5	2.8 days
BLAZE-DEM [15]	Poly^{gpu}	High	32×10^6	32 min

To gauge the large-scale performance of our code we simulated a hopper with a length of 128 cm and filled height of ≈245 cm and variable depth (16 – 256) cm depending on the number of particles . We filled the hopper using a chute which is typical of reality, resulting in an uneven fill level as depicted in Figure 7. Figure 7 shows the hopper with particles colored by velocity. We notice that the left-side of the hopper which has the higher fill level starts to discharge at a faster rate than the right side until the height on both sides reach an equilibrium.

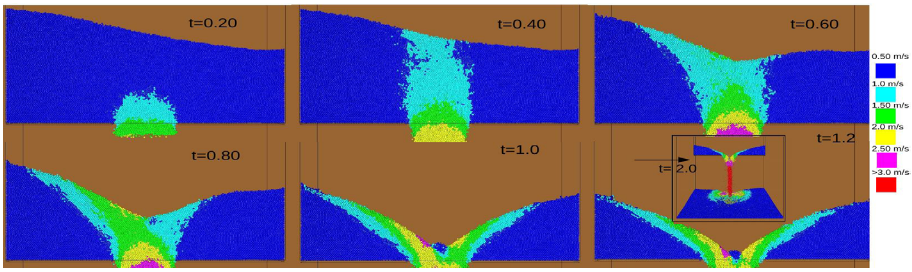


Figure 7: Translational velocity magnitude of flow for a large-scale simulation.

Figure 8 shows the rate of change of flow (% discharged /△t) . We notice an erratic behavior up-to 0.90 seconds, after which it becomes stable as an equilibrium level is reached within the hopper. We also notice that the discharged particles fall into the same area as depicted at (t=2.0) in Figure 7 regardless of the packing within the hopper. This is of importance as the discharged particles typically fills trucks or container vessels, which remain static during the filling process, thus any change in the discharge stream will require them to be dynamic.

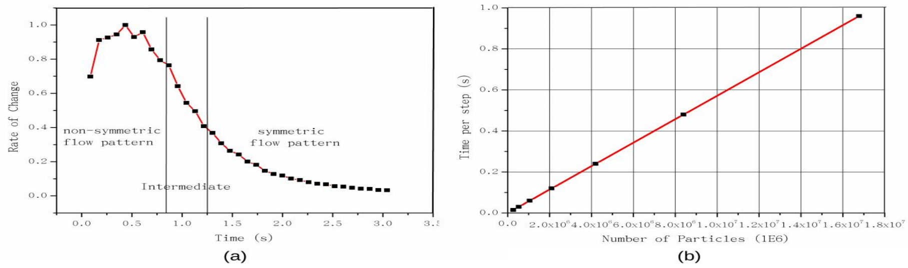


Figure 8: (a) Rate of change of flow, (b) computational time as a function of particle number.

5. Conclusion

In this paper we validated the BLAZE-DEM code against experiment for both polyhedra and spherical particles. We found an excellent agreement in both the flow-rate and pattern of the particles. We then made comparisons between spherical and polyhedral shaped particles and demonstrated that there is a difference in the predicted results. Finally we demonstrated the power of our GPU DEM implementation by performing a large scale simulation. Thus using a typical desktop computer an engineer can now obtain results on design changes and flow dynamics in a realistic time frame. Furthermore this increase in computing power can also be exploited to increase the fidelity of particle shape which we have demonstrated to have an effect of particle discharge.

REFERENCES

[1] W. Beverloo, H. Leniger, J. Van de Velde, The flow of granular material through orifices, Journal of Chemical Engineering Science 15 (1961) 260–296.

[2] V. Vidyapati, S. Subramaniam, Granular flow in silo discharge: Discrete element method simulations and model assessment, Industrial and Engineering Chemistry Research 52 (2013) 13171–13182.

[3] P. Cleary, M. Sawley, DEM modelling of industrial granular flows: 3D case studies and the effect of particle shape on hopper discharge, Applied Mathematical Modelling 26 (2002) 89–111.

[4] P. Cundall, Strack, A discrete numerical model for granular assemblies, Geotechnique 29 (1979) 47–65.

[5] P. Cleary, The filling of dragline buckets, Math. Eng. Ind. 29 (1998) 1–24.

[6] B. Mishra, R. Rajamani, Simulation of charge motion in ball mills. Part 1: experimental verifications, Int. J. Mineral Process 40 (1994) 171–186.

[7] C. Radeke, B. Glasser, J. Khinast, Large-scale powder mixer simulations using massively parallel GPU architectures, Chemical Engineering Science 65 (2010) 6435–6442.

[8] J. Latham, A. Munjiza, The modelling of particle systems with real shapes, Philosophical Transactions of The Royal Society of London Series A: Mathematical Physical and Engineering Sciences 362 (2004) 1953–1972.

[9] H. Abou-Chakra, J. Baxter, U. Tuzun, Three-dimensional particle shape descriptors for computer simulation of non-spherical particulate assemblies, Advanced Powder Technology 15 (2004) 63–77.

[10] D. Hohner, S. Wirtz, V. Emden, H.K. Scherer, Comparison of the multi-sphere and polyhedral approach to simulate non-spherical particles within the discrete element method, Powder Technology 208 (2011) 643–656.

[11] D. Zhao, E. Nezami, Y. Hashash, J. Ghaboussi.J., Three-dimensional discrete element simulation for granular materials, Computer-Aided Engineering Computations: International Journal for Engineering and Software 23 (2006) 749–770.

[12] D. Markauska, Investigation of adequacy of multi-sphere approximation of elliptical particles for DEM simulations, Granular Matter 12 (2010) 107–123.

[13] S. Mack, P. Langston, C. Webb, York.T., Experimental validation of polyhedral discrete element model, Powder Technology 214 (2011) 431–442.

[14] N. Govender, D. Wilke, S. Kok, R. Els, Development of a convex polyhedral discrete element simulation framework for NVIDIA Kepler based GPUs, JCAM 270 (2014) 63–77.

[15] N. Govender, D. Wilke, S. Kok, Collision detection of convex polyhedra on the NVIDIA GPU architecture for the discrete element method, Journal of Applied Mathematics and Computation http://dx.doi.org/10.1016/j.amc.2014.10.013.

[16] A. Munjiza, A combined finite-discrete element method in transient dynamics of fracturing solids., Int. J. Eng. Computation 12 (1995) 145–174.

[17] P. Cundall, Formulation of a three-dimensional distinct element model - part i: a scheme to detect and represent contacts in a system composed of many polyhedral blocks, Int. J. of Rock Mech 25 (1988) 107–116.

[18] J. Sanders, E. Kandrot, CUDA by example, Vol. 12, 2010.

[19] J. Longmore, P. Marais, M. Kuttel, Towards realistic and interactive sand simulation: A GPU-based framework, Powder Technology 235 (2013) 983–1000.

[20] T. Harada, GPU Gems 3: Real-time rigid body simulation on GPUs, Vol. 3, 2008.

[21] J. H. Walther, F. Sbalzarini, Large-scale parallel discrete element simulations of granular flow, Engineering Computations 26 (2009) 688–697.

[22] P. Cleary, M. Sawley, Three-dimensional modeling of industrial granular flows, Proceeding of CFD in the minerals and process industries, CSIRO, Melbourne, Australia, 1999.

[23] N. Govender, R. Rajamani, D. Wilke, S. Kok, Discrete element simulation of mill charge in 3d using the blaze-dem gpu framework., Journal of minerals engineering http://dx.doi.org/10.1016/j.mineng.2015.05.010.

[24] P. W. Cleary, Recent advances in DEM modelling of tumbling mills, Minerals Engineering 14 (2001) 1295–1319.

[25] N. Bell, Y. Yu, Particle-based simulation of granular materials, Eurographics/ACM SIGGRAPH Symposium on Computer Animation 25 (2005) 29–31.

[26] N. Abriak, Local friction effect of the global behaviour of granular media, Mathematical and Computational Modelling 28 (1998) 121–133.

[27] G. Neubauer, C. A. Radek., GPU Based Particle Simulation Framework With Fluid Coupling Ability, NVIDIA GTC 2014, San Jose,USA, 2014.

[28] NVIDIA, Cuda 6 (May 2014).
URL `http://www.nvidia.com/cuda`

[29] S. Jaelee, PhD Thesis:developments in large scale discrete element with polyhedral particles simulations, University of Illinois at Urbana-Champaign,, www.uiuc.edu, 2014.