DEM SIMULATION OF MILL CHARGE IN 3D VIA GPU COMPUTING

*R.K.Rajamani[1] and S.Callahan[2, 3], J.Schreiner[2, 3]

[1]Metallurgical Engineering Department, University of Utah
135 South 1460 East
Salt Lake City, Utah-84112, USA
*(*Corresponding author: rajkrajamani@gmail.com)*

[2]The Scientific Computing and Imaging Institute, University of Utah
72 South Central Campus Drive Room 3750
Salt Lake City, Utah 84112

[3]Sofware Development Center, University of Utah
615 Arapeen Room 310
Salt Lake City, Utah 84108

ABSTRACT

Discrete Element Method simulation of charge motion in ball, SAG and autogenous mills has become a standard for lifter design, power draft evaluation, etc. Both two-dimensional and three-dimensional codes are being used. The two-dimensional code dominates the user market since the code completes a simulation in one or two hours. On the other hand, three-dimensional codes have been in the domain of specialists who can handle the complexity of the code as well as have access to parallel computing hardware. For the first time, this barrier is overcome via computing with graphic processing units (GPU). Commodity graphics cards now allow parallel computation with hundreds of processors. It is shown that the computational time has been reduced by a factor of fifty using GPU computing. Hence, SAG charge simulation results in three dimensions are achievable in a few hours using affordable desktop machines. The method shown here, opens up the three dimensional simulation as a routine task for mill designers and researchers. On the technical side, the simulations can be elevated to a higher level of complexity to simulate breakage and transport.

First, the speedup factor of fifty is established for the Nvidia GeForce GTX 580 graphics card. Next, four simulation tasks, consisting of AG, SAG and ball mills are reported. The simulation of AG and SAG mills are completed in less than 8 hours. The ball mill simulation with 1.25 million spheres is completed in 27 hours. In the near future, these compute times can be halved with the advances in graphic card hardware.

KEYWORDS

Discrete element method, grinding, simulation, GPU, computing, comminution, tumbling mill

INTRODUCTION

Ever since the introduction discrete element method (DEM) for the simulation of grinding mills there has been a phenomenal growth in the variety of ways this technique is used in the mining industry. In the late 90s two dimensional DEM codes have been the norm and it continues to be even today because of its ease of execution on a single central processing unit (CPU) personal computer. A handful of organization, perhaps three or four, have the capability of three dimensional grinding mill simulation, yet it is available only to specialist who can handle the complex tasks to run the codes. As a result the 3D codes are not prevalent in the hands of mill managers, metallurgists, mill design teams, and operators.

At the outset it is useful to discuss the merits of 3D code in comparison with 2D code. The 2D code is easy to execute in a matter of one or two hours on single CPU. It has been heavily used in hundreds of mining operations for annual or semi annual replacement of shell lifters. This code has impacted the production, capacity and liner life of ball mills, autogenous mills and semi-autogenous mills. This being so it is hard to imagine what other benefits a 3D code would bring. It is easy to argue that 3D code animations are more accurate because the momentum of balls and rock particles in the axial direction of the mill is accounted for. It has been used for the simulation of slurry and pebble motion in the pulp lifter. More importantly, we have not discovered the insights possible with the 3D code because it has not been readily available to researchers and mill designers. Even if it is available, as in the case at the University of Utah, the execution times are of the order of weeks on a single CPU for a typical plant size mill, which then becomes awkward and laborious to pursue on a routine basis.

All of these issues are disappearing with the introduction of graphic processing unit (GPU) commonly found in the video card of every PC in use today. Computing on the GPU is hundreds of times faster than a CPU. In this manuscript we show how plant scale mill simulation can be done in a matter of hours and 10s of hours for very large mill simulations.

BACKGROUND LITERATURE

In 2001 Venugopal and Rajamani presented the DEM 3D computational methodology and showed comparison with power draft in a laboratory scale 90 cm diameter mill. In the same year Rajamani and Mishra (2001) used the same 3D code for the prediction of power draft in plant scale mills. A parallelized version of the same 3D code produced a speed up of 3 to 4 over single CPU (Rajamani et al., 2003). Herbst and Nordell (2001) combined 3D DEM with smoothed particle hydrodynamics (SPH) and finite element method (FEM) for charge motion, ore particle breakage and liner wear.

From the early days Cleary (2001a) has been strong proponent of 3D simulation. Cleary (2001b) demonstrated the sensitivity of charge behavior and power draft of a 5-m ball mill to liner geometry and charge composition using 3D code. There are continued advances in the simulation of breakage and slurry flow incorporating all the details in three dimensional simulation. Morrison and Cleary (2008) describe the evolution of "Virtual Comminution Machine", a simulation code that simulates breakage and slurry transport in tumbling mills. In this simulation both the discrete element method and smoothed particle hydrodynamics are employed for slurry and pebble flow through the grate slots and the pulp lifter. The basis of slurry flow via the porosity of ball and ore particle charge in tumbling mills is described in detail by Cleary et al. (2006). Coupling DEM of particles with smoothed particle hydrodynamics is done via Darcy drag law and Kozeny-Carman model of permeability. This virtual comminution machine package would have the capability to simulate all aspects of say a SAG mill during the design phase. Cleary and Morrison (2009) show that 3D DEM combined with SPH is a tool for analyzing mineral processing equipment such as mills, twin deck screen and so on.

In a more recent study Alatalo et al. (2010) compared the experimental deflection of a lifter in a 1.44 × 1.22 m pebble/ ball mill with 3D predictions made with EDEM, the commercial DEM code. They concluded that 3D predictions were closer to experimental values than 2D.

In view of advances in the last 20 years it is clear that simulation of comminution machines is moving into three dimensional codes in the 21st century. In tandem, the advances in computing, GPU, OpenGL, OpenCL and CUDA technologies are helping this 3D simulation move up much more rapidly.

GPU BASED THREE DIMENSIONAL MILL SIMULATION

DEM algorithm

The details of discrete element calculations are described in a large number of publications. Here, we describe the computational steps necessary to execute the algorithm and how this algorithm is implemented on the GPU.

Initially the mill design and operating variables are set in the code. In particular, the interior envelop of the mill including shell lifters, end lifters and cone heads in the feed and discharge ends are set using a code such as AutoCAD. Next, the charge of desired filling and size class/density class composition is set in the code. These data are usually set with reference to the three dimensional (X, Y, Z) axis.

The computations are done for small time step of 10^{-5} seconds. We denote the time step calculations as a frame. Thus a typical tumbling mill calculation would require a million frames.

In each frame, the first task is to search for contacts between different objects initially set in the code. In the current GPU code, the objects are cylinder, lifter bars, spheres (balls) and end plates. For the sake of clarity, we assume 100,000 spheres in the simulation.

The contact list is created by taking a single sphere and searching in the near neighbourhood for other objects that are overlapping with it. In a single CPU this searching scheme would proceed from the 1st sphere to the 100,000th sphere. The near neighbourhood search scheme is referred to as “boxing” in the DEM literature.

The next step is force calculation. A sphere may have four or more contacts depending on its location within the mill. At each contact a collision model is used to calculate the net force between the pair of objects and finally resultant force on the sphere is resolved in the three coordinate directions. The contact model commonly used is the spring and dashpot model. Thus the resultant force on each of the 100,000 spheres is updated one by one on a single CPU.

Integration of force by specialized algorithms such as Verlet algorithm yields velocities and net displacement of the sphere. Here too the calculation proceeds one by one for each sphere on a single CPU. Therefore, the nature of the DEM algorithm is ideal for computing on a GPU where multiple processors are available. Hence, the entire data can be copied to GPU memory and the multiple processors in the GPU can simultaneously do the compute for a number of spheres in a single clock cycle.

The sphere calculation procedure is mapped to the GPU depending on its architecture. In this development we have used Nvidia’s GTX 580 card, a consumer graphics card priced at $500. This video card has 512 cores and 3GB memory. The code development was done on a desktop PC with i7 – 920 CPU under Windows Vista operating system. This code is written in OpenCL for general computing and OpenGL for graphical display of the mill.

The GPU consists of a set of single instruction multiple thread multiple-processors which are mapped into blocks. Each block processor has an instruction unit and several scalar processor cores which are mapped into threads.

Each scalar thread executes with its own data set. Accordingly, the sphere calculation processes are divided into parallel blocks, each of which contains certain number of spheres. Likewise, both the sphere force and sphere motion calculations are distributed following the block-thread architecture.

GPU DEM PSEUDOCODE

The tasks that should be computed for every ball are called a kernel in GPU language. Following are the kernels implemented in the code.

1. Run ball binning / boxing kernels to determine the proximity of balls. For each cell in grid make a list of balls.
2. Run collision kernels to update contact links. For each ball, determine collision with lifter, plate, line, cylinder and point objects.
3. Run force kernels. Compute normal and shear force, and sum forces on each ball.
4. Run motion kernel. Move each ball as per the net force acting on it.
5. Render the mill and balls in OpenGL window.
6. Update data state for next time step.

There are many issues in fully utilizing the 512 cores to do parallel computing on the GPU. The binning/ boxing kernel is not a trivial task to parallelize because the associated information storage is not data parallel. The more difficult part is the contact link since the contact link changes from frame to frame, a number of tricks are implemented in order to make it as much parallel as possible. The DEM calculations are done in double precision, which is a severe issue for graphics card, which is optimized for single-precision floating-point arithmetic. For current GPUs, the speed of double precision calculations is about 1/8th that of single precision. These are some of the issues which reduce the efficiency of GPU calculations.

Four milling simulation projects were taken up to illustrate the power of GPU DEM computing. Table 1 gives the details of mill design and operating variables for two AG mills, one SAG mill, and one ball mill. The AG mill and SAG mill data were taken by Powell and Valery (2006). In parallel computing speedup gained with GPU over CPU must be established first. The speedup value indicates how effectively the cores in GPU are utilized. For this purpose, the SAG mill project B in table 1 was taken up as a test case. The simulation for 5,000 particles is done with both the CPU and GPU giving the results as shown in Table 2. Clearly, the speedup is 50 in favour of GPU. This is really remarkable since a 50 hr simulation on a CPU can be done in just one hour on the GPU. Considering the GPU is nothing but an ordinary desktop PC with a graphics card, the speedup value of 50 is remarkable, since the capital investment in a single PC is far less compare to a parallel PC hardware.

Table 1 – Mill design and operating conditions

Project	Mill type	Diameter × Length (m)	Mill speed (rpm)	Mill filling (%)	Number of rows	Shell lifters (high: low) h × w × θ
A	Autogenous	8.37 × 4.03	11.0	41	56	0.15 × 0.15 × 15 0.075 × 0.3 × 30
B	Semi-autogenous	10.12 × 4.722	10.0	17.5	64	0.225 × 0.15 × 22 0.05 × 0.2 × 30
C	Semi-autogenous	8.26 × 4.189	11.0	23.7	56	0.175 × 0.125 × 22 0.05× 0.2 × 30
D	Ball mill	7.32 × 10.98	12.2	38.0	96	0.1 × 0.075 × 28 0.1× 0.075 × 28

Table 2 – GPU speed up for 5,000 ore particles simulation in a 10.12 × 4.72 m semi-autogenous mill

Mill simulation time (sec)	Single CPU compute time (sec)	GPU compute time (sec)	Speed up
0.2	1,192	25	48
1.0	6,245	136	46
3.0	21,227	407	52
5.0	34,457	637	54

Next, we show the simulation done for the four milling projects in Table 3. The SAG and AG mill simulations are completed in less than 8 hours. The ball mill simulation, in which one and a quarter million balls are simulated, is completed in 27 hours. As graphics card technology improves these simulation times may be halved in the next few months of 2011.

Table 3 – GPU compute times for mill simulation projects

Project	Mill type	Number of balls/ ore particles	Mill simulation time (sec)	Frame advance per second	GPU memory (MB)	GPU compute time (hr)
A	28 ft AG	212,704	10.9	39	389	7.76
B	34 ft SAG	91,357	12.0	93	170	3.58
C	28 ft SAG	244,565	10.9	35	450	8.65
D	24 ft BALL	1,259,414	9.83	5	2,200	27.30

* Each frame advances simulation time by 2×10^{-5} sec.

Figure 1, 2, 3, and 4 show the charge motion in the respective mill simulations. In these simulations we are able to watch the mill charge animation as the calculations are advancing frame by frame in real time. This real time feature is useful for mill lifter design.

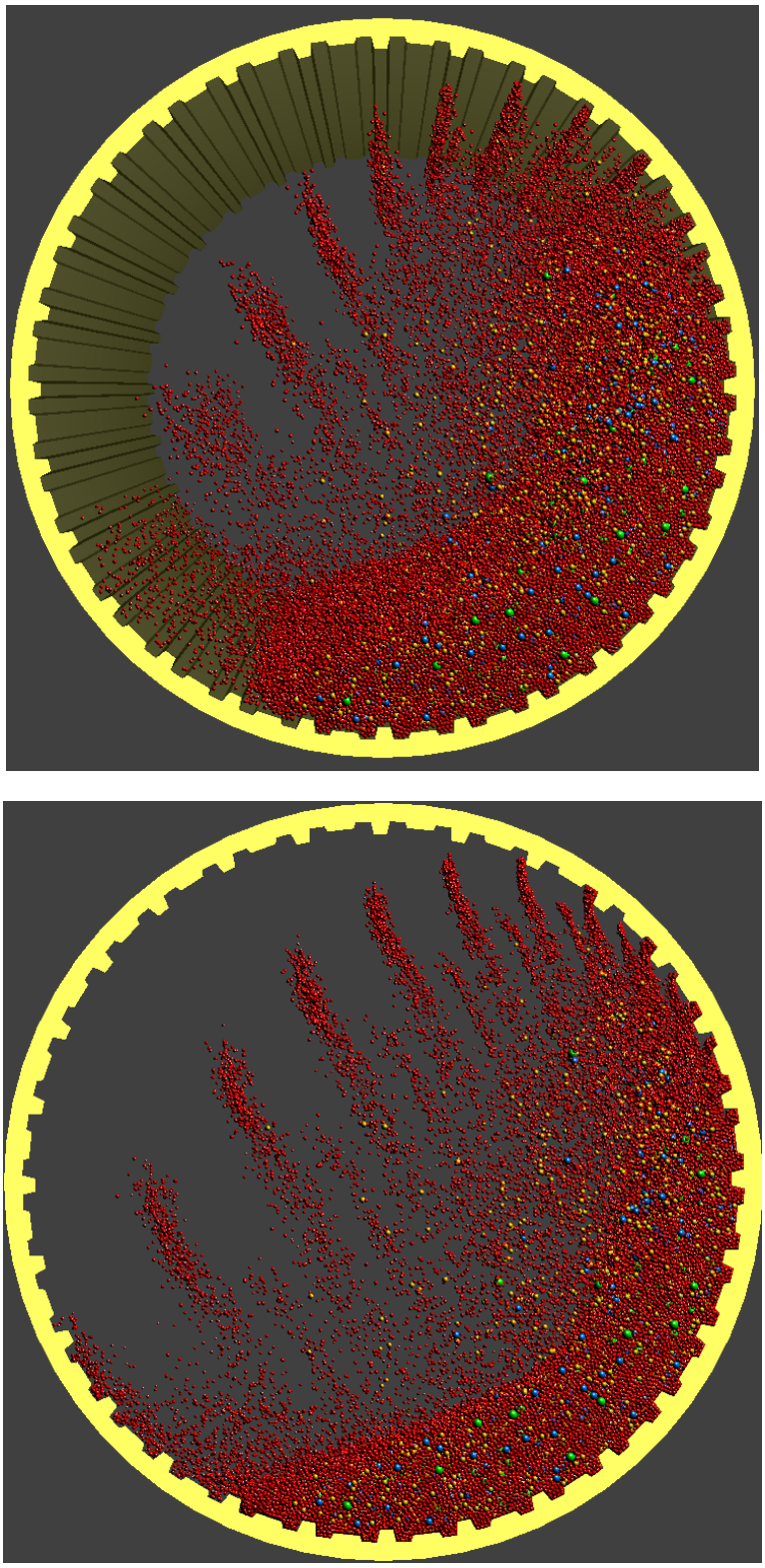

Figure 1 – 8.37 m (28 ft) Autogenous mill simulation: isometric and orthogonal view of charge motion

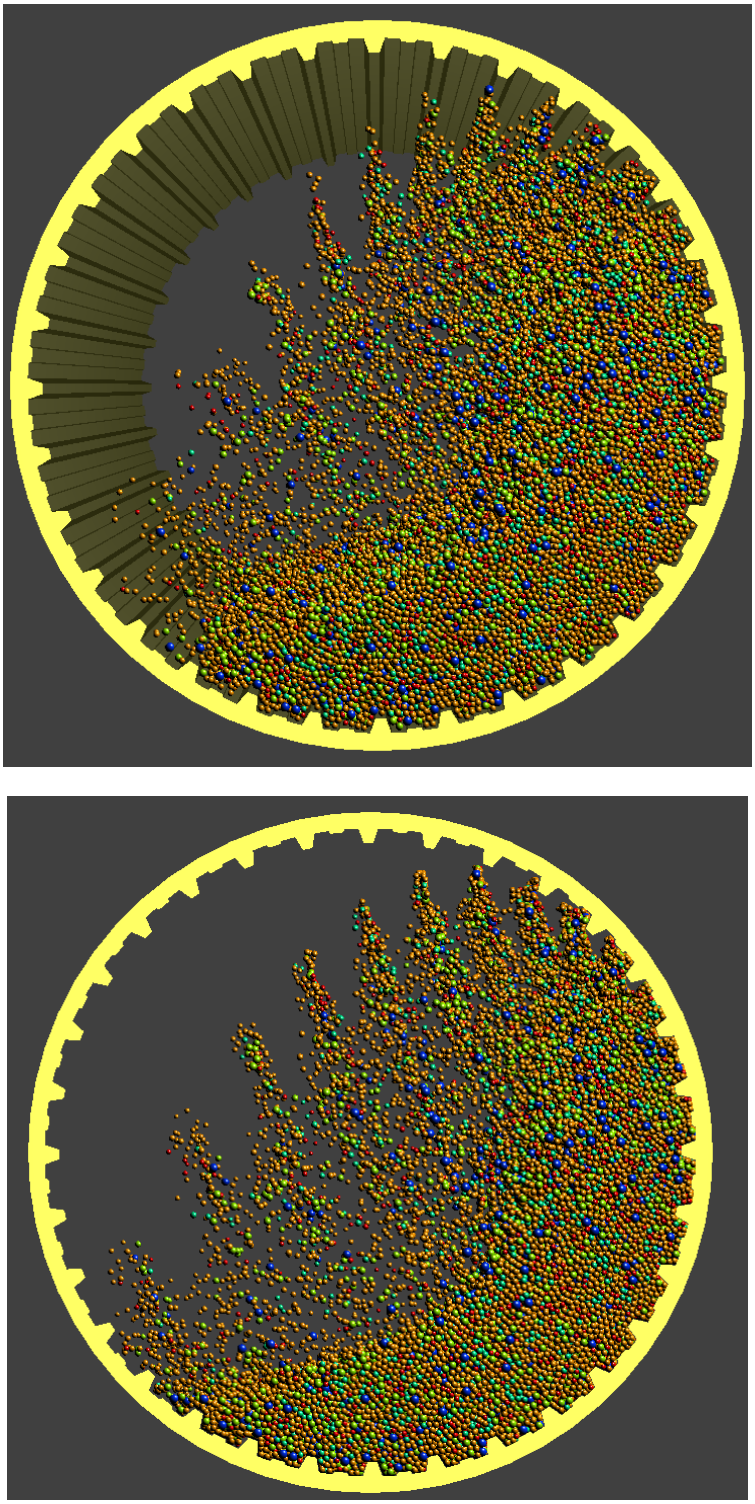

Figure 2 – 10.12 m (34 ft) Semi-autogenous mill simulation: isometric and orthogonal view of charge motion

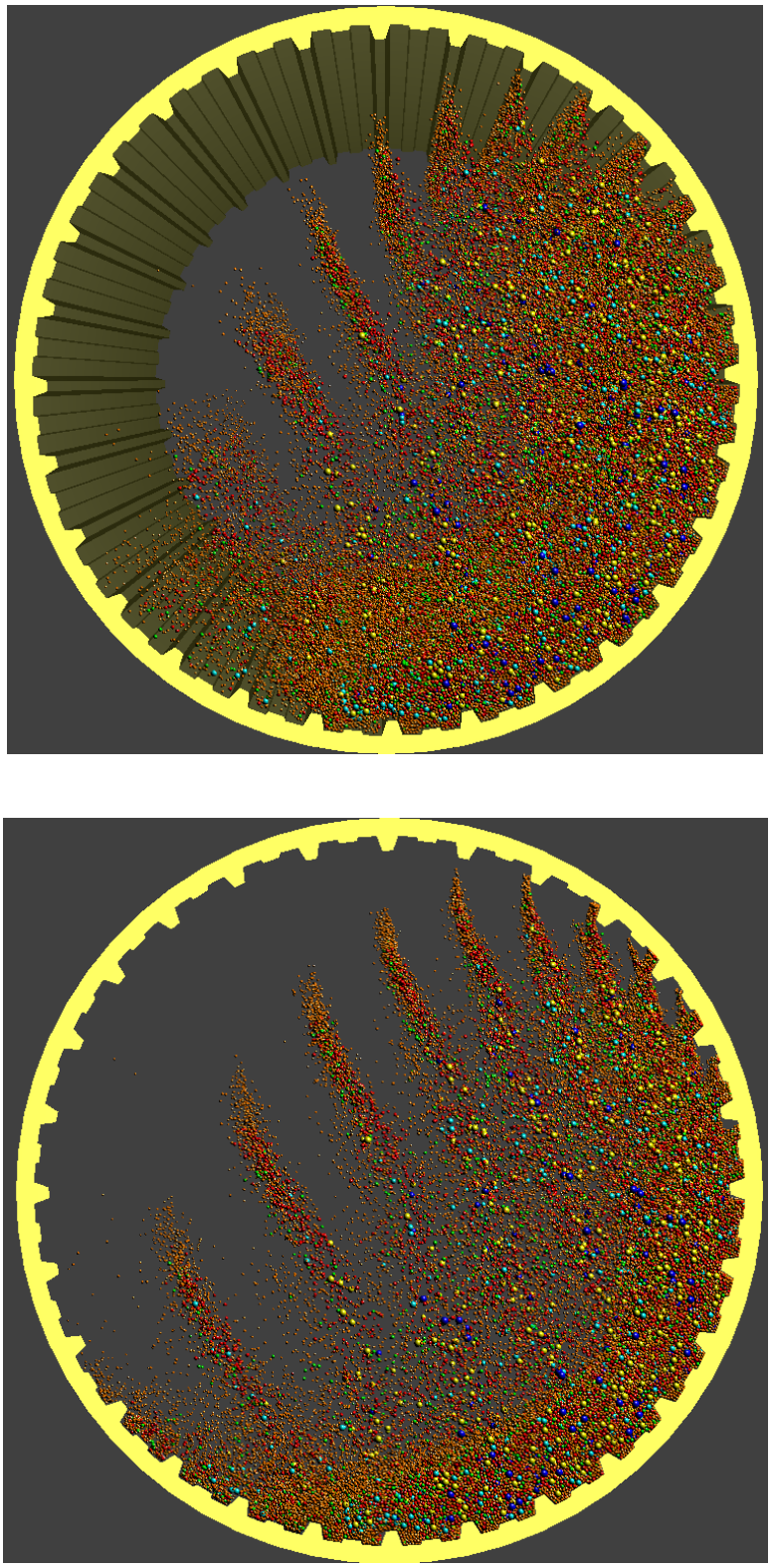

Figure 3 – 8.26 m (28 ft) Semi-autogenous mill simulation: isometric and orthogonal view of charge motion

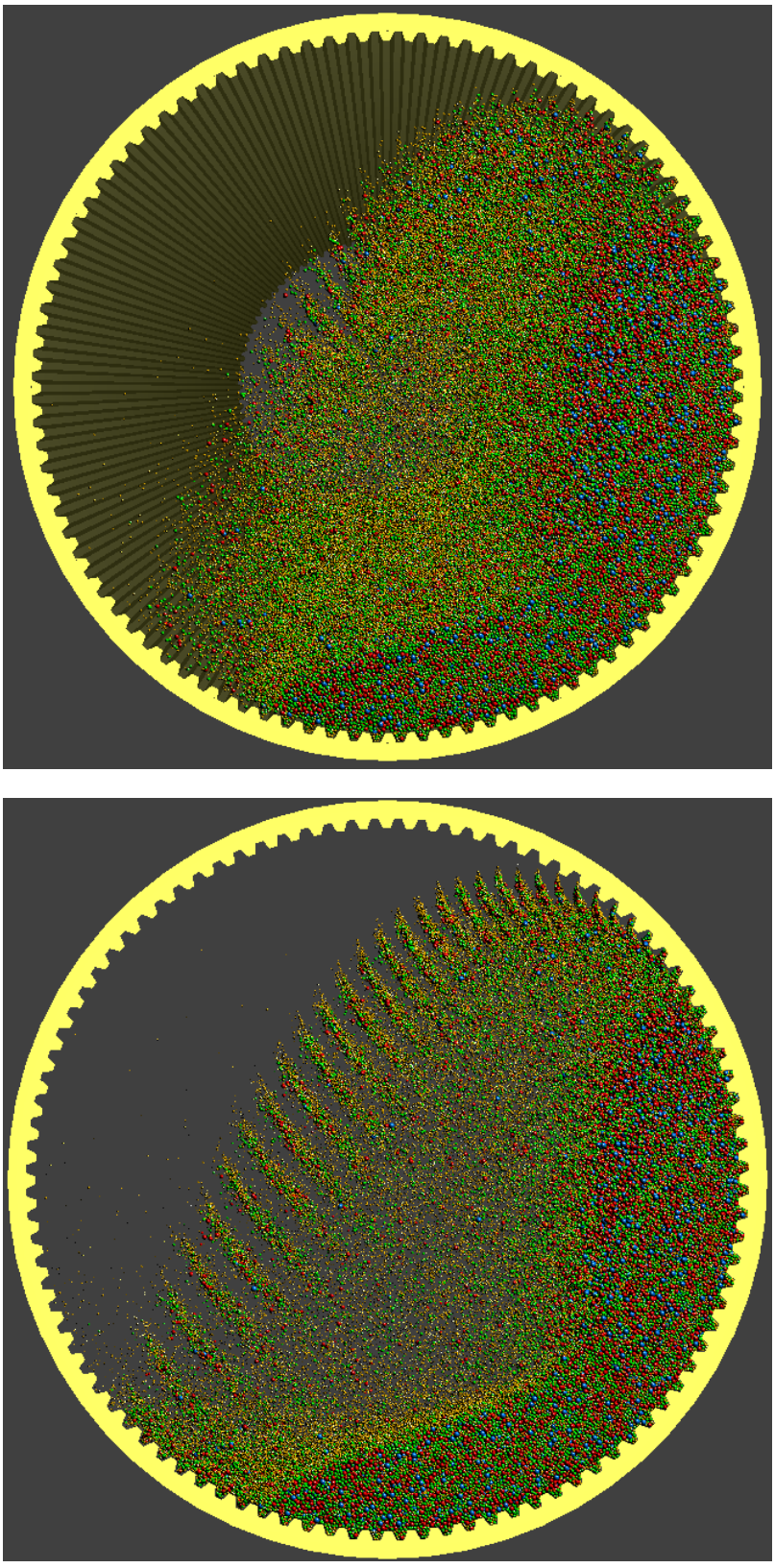

Figure 4 – 7.32 m (24 ft) Ball mill simulation: isometric and orthogonal view of charge motion

CONCLUSIONS

GPU computing is ideally suited for DEM simulations. It is shown that DEM computational algorithm can be broken up into a few kernels and the kernels can be deployed in multiple cores of the graphics card for parallelizing the DEM computations.

In particular, with the Nvidia GTX 580 card a speedup of 50 was achieved. Simulation times of SAG and AG mills were limited to 8 hours or less. Even a ball mill simulation with 1.25 million balls was completed in 27 hours. Newer and powerful graphics cards are appearing in the market almost monthly. Hence, in the near future the code presented here may be executed in less than half the time reported in this manuscript.

ACKNOWLEDGEMENTS

The authors would like to thank the financial support provided by the Technology Commercialization Office, University of Utah. R. Rajamani would like to thank Dr. C. L. Lin, Research Professor, University of Utah, who inspired him to take up GPU computing.

REFERENCES

Alatalo, J., Palsson B. & Tano, K. (2010). Comparing Experimental Measurements of Mill Lifter Deflections with 2D and 3D DEM Predictions. DEM5 Proceedings, Queen Mary University, London, UK, 194 – 198

Cleary, P. W. (2001a). Recent Advances in DEM Modelling of Tumbling Mills. Minerals Engineering, 14 (10), 1295 – 1319

Cleary, P. W. (2001b). Charge Behaviour and Power Consumption in Ball Mills: Sensitivity to Mill Operating Conditions, Liner Geometry and Charge Composition. International Journal of Mineral Processing, 63, 79 – 114

Cleary, P. W., Sinnott M. & Morrison, R. (2006). Prediction of Slurry Transport in SAG Mills using SPH Fluid Flow in a Dynamic DEM based Porous Media. Minerals Engineering, 19, 1517 – 1527

Cleary, P. W. & Morrison, R. (2009). Particle Methods for Modelling in Mineral Processing. International Journal of Computational Fluid Dynamics, 23(2), 137 – 146

Herbst, J. & Nordell, L. (2001). Optimization of the Design of SAG Mill Internals using High Fidelity Simulation. International Conference on Autogenous and Semiautogenous Grinding Technology, SAG 2001, Vancouver, BC, Canada, 150 – 164

Hromnik, M. & Govender I. (2010). Large Scale GPU Implementation of the Discrete Element Method Applied to Modeling the Environment in the Positron Particle Tracking Experiment. DEM5 Proceedings, Queen Mary University, London, UK.

Mishra, B. K. & Rajamani, R. K. (2001). Three Dimensional Simulation of Plant Size SAG Mills. International Conference on Autogenous and Semiautogenous Grinding Technology, SAG 2001, Vancouver, BC, Canada, 48 – 57.

Morrison, R.D. & Cleary, P. W. (2008). Towards a Virtual Comminution Machine. Minerals Engineering, 21, 770 – 781

Powell, M.S. & Valery W. (2006). Slurry Pooling and Transport Issues in SAG Mills. International Conference on Autogenous and Semiautogenous Grinding Technology, SAG 2006, Vancouver, BC, Canada, 134 – 152.

Rajamani, R. K., Mishra, B. K., Joshi A. & Park J. (2003). Two- and Three-Dimensional Simulation of Ball and Rock Charge Motion in Large Tumbling Mills. Proceedings of the American Society of Civil Engineers, Geotechnical Special Publication, 117

Venugopal, R. & Rajamani, R K. (2001). 3D Simulation of Charge Motion in Tumbling Mills by the Discrete Element Method. Powder Technology, 115, 157 – 166