II International Conference on Particle-based Methods – Fundamentals and Applications
PARTICLES 2011
E. Oñate and D.R.J. Owen (Eds)

COSIMULATION OF MBD (MULTI BODY DYNAMICS) AND DEM OF MANY SPHERES USING GPU TECHNOLOGY

JOON SHIK YOON *, JI SOO PARK †, CHEOL O AHN †† AND JIN HWAN CHOI †

*Department of Mechanical Engineering
Seoul National University
Seoul 151-742, Korea
email: jueno@functionbay.co.kr

† Department of Mechanical Engineering
Kyung Hee University
Yongin 449-791, Korea
email: jspark83@functionbay.co.kr, jhchoi@khu.ac.kr

†† Metariver Technology Co., Ltd.
Seongnam 463-841, Korea
email: coahn@metariver.kr

Key words: MBD, DEM, Particle Dynamics, Cosimulation

Abstract. In this paper, dynamic simulation model which have many sphere particles and MBD (Multi Body Dynamics) entities, i.e. bodies, joints, forces, is built and simulated.

Many sphere particles are solved with DEM (Discrete Element Method) and simulated with GPU technology. Fast algorithm is applied to calculate hertzian contact forces between many sphere particles (from 100,000 to 1,000,000) and NVIDIA's CUDA is used to accelerate the calculation. Explicit integration method is applied to solve the many spheres.

MBD (Multi Body Dynamics) entities are simulated with recursive formulation. Constraints are reduced by recursive formulation and implicit generalized alpha method is applied to solve dynamic model.

Many sphere particles and MBD (Multi Body Dynamics) entities are co-simulated within commercial software RecurDyn. The interaction forces between many sphere particles and rigid body mesh are calculated and applied to each body to simulate two parts simultaneously.

These models are built and simulated; fork lifter with sand model, oil in oil tank model, oil filled engine system and water filled washing machine model. All models are simulated with NVIDIA's GPU and the result is shown.

1 INTRODUCTION

Today, parallel GPUs have begun making computational inroads against the CPU and general purpose GPU can be used in simulation algorithm to amplify the performance. The parallel implementation of algorithm, when executed on a ubiquitous Graphics Processing Unit (GPU) card, yields a 30 fold speedup over a similar algorithm executed on the Central Processing Unit (CPU). With the introduction of NVIDIAs Compute Unified Device Architecture (CUDA) [1], GPUs are now able to run C code natively on the device instead of relying on interpreted code.

A discrete element method (DEM) is any of family of numerical methods for computing the motion of a large number of particles of micrometre-scale size and above. As GPU can boost the calculation speed of parallel programming, DEM(Discrete Element Method) is becoming widely used in granular and discontinuous materials, especially in granular flows, powder mechanics, and rock mechanics. Some examples are granular matter (rock, sand, soil), powders, liquid and solutions. And typical industries using DEM are chemical, pharmaceutical, mining, agriculture and food handling, powder metallurgy and digital printing. With advances in computing power and numerical algorithms, it has become possible to simulate millions of particles numerically.

While many particles are solvable with DEM and parallel programming, it is called Particle Dynamics in this study; the only Particle Dynamics have a limitation on system modeling which can cover system with constraints and motions. Whole dynamic simulation mostly handle constraints and motions and even can be connected with other technique like control, optimization. RecurDyn[2] is commercial software and simulates multi body dynamics and finite element together. If the particle dynamics is included in MBD, the synergy effect of the simulation can be enlarged.

To connect two kinds of the simulation, MBD and Particle Dynamics, the dividing domain algorithm and interaction algorithm is essential. Within these algorithms, the cosimulation should be performed well even though two simulations are quite different. In this study, cosimulation of MBD and Particle Dynamics is introduced and the algorithms of dividing and interaction are also explained.

This paper is organized as follows. After introduction basic theories of Particle Dynamics on DEM and MBD are shown in charter 2.1 and 2.2 respectively. Integration methods of each method are described in chapter 2.3. Cosimulation of two methods is given in chapter 2.4. and GPU acceleration technique is given in chapter 2.5. Chapter 3 is numerical experiment for Particle Dynamics validation. Chapter 4 is cosimulation of Particle Dynamics and MBD. Final chapter summarize the conclusion of this study.

2 FORMULATION

2.1 PARTICLE DYNAMICS (DEM)

Particle-based techniques are used in many applications. DEM is a numerical method for computing the behavior of large number of solid-particle system. This simulation method consists of the idea of determining the kinematic force to each finite-sized particle. All forces acting on each particle are modeled and calculated at every discrete-time step.[3][4] In time

integration algorithm, most of the MBD solvers are using implicit algorithm but DEM adopts explicit time-integration. The trajectories of particles are updated by Newton's law of motion, according to the following equations:

$$\frac{d\mathbf{v}}{dt} = \frac{\sum \mathbf{F}}{m} \tag{1}$$

$$\frac{d\boldsymbol{\omega}}{dt} = \frac{\sum \mathbf{M}}{I} \tag{2}$$

where $\mathbf{v}$ is the particle velocity, $\mathbf{F}$ is the summed force acting on a particle, m means the mass of a particle, $\boldsymbol{\omega}$ is the angular velocity, and $\mathbf{M}$ and I denote the moment of force and the moment of inertia.

A contact model between two particles is given by the Hertzian contact model and Voigt model, which consists of a spring dashpot and a slider for the friction in the tangential component[5]. The contact forces $\mathbf{F}_n$, compressive, and $\mathbf{F}_t$, shear, are calculated by the following equations:

$$\mathbf{F}_{n,ij} = k_n |\delta|^{1.5} \mathbf{n}_{ij} + c_n \mathbf{u}_{n,ij} |\delta|^{0.25} \tag{3}$$

$$\mathbf{F}_{t,ij} = \min\left[k_t |\delta_t|^{1.5} \mathbf{t}_{ij} + c_t \mathbf{u}_{t,ij} |\delta_t|^{0.25}, \mu_f |\mathbf{F}_n| \mathbf{t}_{ij} \right] \tag{4}$$

where k and c designate the spring and the damping coefficients, the vectors $\mathbf{n}_{ij}$ and $\mathbf{t}_{ij}$ are the unit vectors from the i-th particle to the j-th one in normal and tangential components, δ and $\mathbf{u}$ are the deformation by contact and relative velocity of two particles, respectively.

The force acting on a body can be obtained from the particles which contact on the body by summing their force.

2.2 Multi Body Dynamics (MBD)

Rigid body dynamics can be modeled using various formulations (Jalón and Bayo [6]). In this investigation, the recursive formulation is used. This section provides an introduction to the recursive formulation. The coordinate systems for two contiguous rigid bodies in 3D space are shown in Fig. 1. The two rigid bodies are connected by a joint, and an external force $\mathbf{F}$ is acting on the rigid body j. The X-Y-Z frame is the inertial or global reference frame and the x′-y′-z′ is the body reference frame with respect to the X-Y-Z frame. The subscript i means the inboard body of body j in the spanning tree of a recursive formulation (Bae et al. 2001)[7].

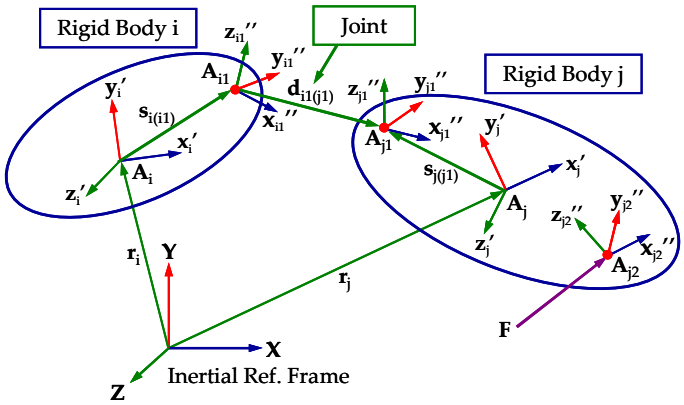

Figure 1. Two contiguous rigid bodies

The equations of motion for a constrained mechanical system (García de Jalón et al. [8]) in the joint space (Wittenburg [9]) are then obtained by using the velocity transformation method as follows:

$$\mathbf{F} = \mathbf{B}^{\mathrm{T}} \left(\mathbf{M}\,\dot{\mathbf{Y}} + \boldsymbol{\Phi}_{\mathbf{Z}}^{\mathrm{T}}\boldsymbol{\lambda} - \mathbf{Q} \right) = \mathbf{0} \tag{5}$$

where $\boldsymbol{\Phi}$ and $\boldsymbol{\lambda}$, respectively, denote the cut joint constraint and the corresponding Lagrange multiplier. $\mathbf{M}$ is a mass matrix and $\mathbf{Q}$ is a force vector including the external forces in the Cartesian space.

2.3 Implicit and Explicit Integrator

For the multibody systems, there are various methods of implicit and explicit solution procedures that are used to solve the semi-discrete equations of motion along with the constraint. In implicit solution procedure, a solution for the system displacements that simultaneously satisfies the equations of motion and constraints is sought at each time step given the solution at the previous time step. Since the equations are nonlinear, Newton-Raphson equilibrium iterations are performed to guarantee that an equilibrium solution is reached at each time step (Brenan et al. [10], Haug and Deyo [11], Hairer and Wanner [12]).

Implicit solution procedures are unconditionally stable. However, the time step should be at least an order of magnitude smaller than the smallest natural period that needs to be resolved. An advantage of implicit solution procedures over explicit procedures is that the time step can be much larger than the smallest natural period of the system, which can be very small for very stiff systems. There is a very close relationship between the solution methods and the constraints modeling methods. The floating frame approach is usually used in conjunction with the Lagrange multiplier method for imposing the constraints.

In explicit solution procedures (Hughes and Belytschko [13]), a solution for the accelerations that satisfies the equations of motion and constraints is sought at each time step. If a mass matrix is used, then the system's equations of motion are uncoupled at each time step and they can be directly solved for the accelerations. A typical explicit algorithm starts by evaluating the vector of internal forces from the known positions and velocities at time step t. Then, internal forces are added to the external forces. The equations of motion are then directly used to calculate the accelerations at time step $t + \Delta t$. A time integration formula

such as the trapezoidal rule is used to integrate the acceleration into the velocities and positions at time step $t + \Delta t$.

Explicit temporal integration techniques are only conditionally stable because the time step must be smaller than the equation's characteristic time. If the same time step is used for the entire system, then that time step must be smaller than the smallest natural period of all bodies. This imposes a severe time step restriction and generally means that a very large number of time steps are needed to obtain the dynamic response.

Multi body dynamics have constraints and therefore constraints are satisfied with implicit integration. The governing equations of multibody dynamics, which are derived previous sections, is solved using the implicit generalized-alpha method (Chung and Hulbert [14]), which has a nonlinear stepping equation. By the way particles are have huge DOF compared with MBD and there are no constraint equations. The particles are solved with explicit integration and it is accelerated with parallel processing using GPU.

2.4 Cosimulation of Particle Dynamics and MBD

Simulation of many particles in multi body systems needs strategy of cosimulation. Many particles are solved with explicit integration and multi body systems are solved with implicit integratoion. Figure 2 shows a sample idea of cosimulation about MBD and particles. The whole modeling is composed with a spring, a box and many spheres. The spring and the box is solved with RecurDyn[2](MBD Solver) while many particles are solved with SAMADII^{TM}[15] (Particle Solver). Two solver divides the model with boundaries and in this sample the box is the boundary. The boundary force is given by the Particle Solver and boundary position, velocity is given by the MBD Solver.

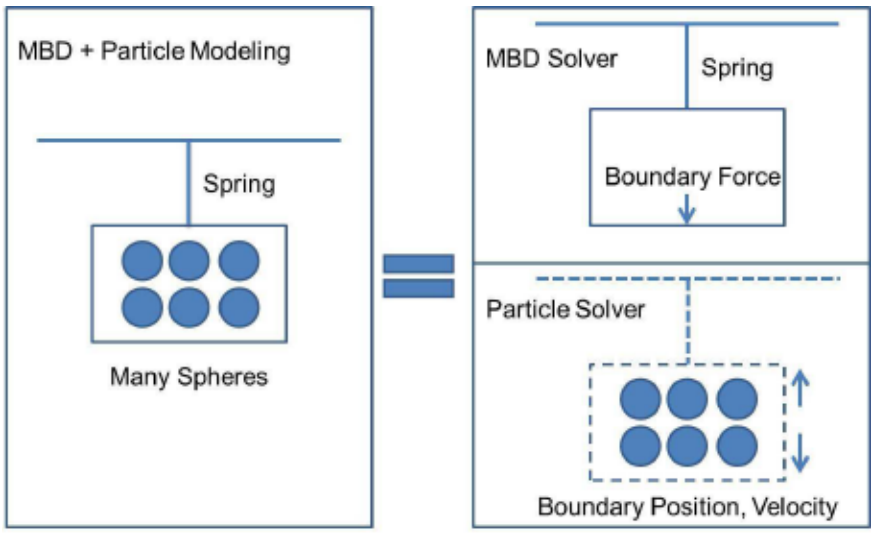

Figure 2. Modeling of MBD and Particle Dynamics

2.5 Simulation Acceleration Techniques (GPU)

The main disadvantage of particle-based simulation with a very large number of particles is that it requires a very heavy computing resources to obtain satisfactory results. But it is relatively not so difficult to parallelize particle systems. Perhaps GPU computing is the best way to achive ability to handle large number of particle system efficiently.

Therefore we adopted the DEM solver using GPU system, the SAMADII^{TM} (Metariver Technology co., ltd.) [15]. It was developed to simulate particle-based system using GPU.

GPUs are massively parallel multithreaded devices capable of executing a large amount of active threads. GPU has multiple streaming multiprocessors, each of which contains multiple scalar processor cores. A function that executes on the GPU consists of multiple threads executing code in a single instruction, multiple data (SIMD) fashion. That is, each thread in a kernel executes the same code, but on different data. The libraries CUDA from NVIDIA allows one to use the streaming microprocessors mounted in high-end graphics cards as general-purpose computing hardware. Presently, the raw computational power of these multiprocessors can reach one Teraflop, which is several hundred times the throughput of a modern scalar CPU.

The SAMADII^{TM} was developed with hardware acceleration by GPU as well as software acceleration using the cell-linked list algorithm to accelerate DEM simulation.[16] The simulation space is partitioned into cells and the particles are then assigned to the cells so that it is easier to find the neighbors of a given particle. At the beginning of a simulation, an array that contains a list of cell neighbors for each cell is created. This method dramatically reduces the number of unnecessary inter-particle distance calculations.

3 PARTICLE DYNAMICS VALIDATION

A validation model is total weigth of spheres in a box. 1,331 spheres are filled in a box as Figure 3. The weigth of the sphere is 12.62kg and therefore the total weigth of the spheres are 16797.22kg. And as a result of the simulation, the force imposed to the box by many spheres is 16765.38kg and there are only 0.19% error in wieight of the total weight of the spheres.

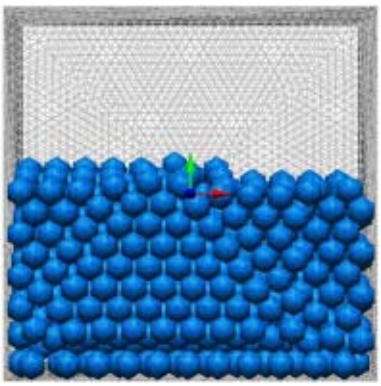

Figure 3. Total sphere weigth in a box model

4 COSIMULATION NUMERICAL EXPERIMENT

In this section simulation result of a cosimulation model is shown. The model is modeled in MBD software RecurDyn and the particles are cosimulated by SAMADII™. Figure 4 shows a waterfall model. The left figure is modeling of the waterfall and the other figures are simulation results of the spheres.

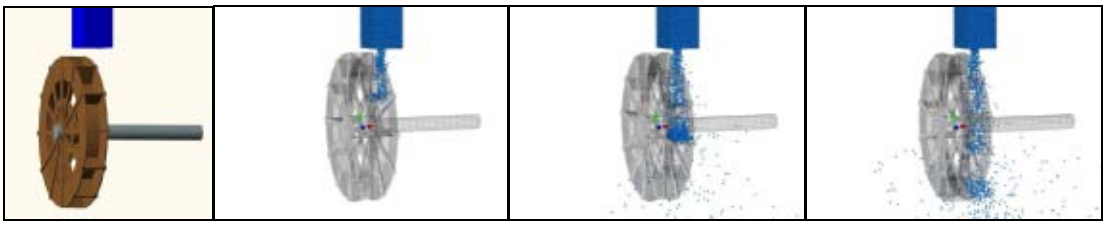

Figure 4. Waterfall modeling and simulation results

The model is simulted with one GPU TESLA2050 for many spheres. Table 1 listed the specification of the models; number of spheres, simulation time, stepsize, CPU time and so on.

Table 1. Result of cosimulation models

	Waterfall
Number of spheres	52,000
Radius of spheres	0.01m
Simulation endtime	2 sec
Stepsize of MBD	1.0e-3
Stepsize of Particle	2.89e-6
CPU time	5 hour 51 mim

5 CONCLUSIONS

In this study Particle Dynamics is introduced which have high performance using GPU technology. And cosimulation of MBD and Particle Dynamics is also explained with connecting algorithm. The Particle Dynamics is validated with total sphere weight model. For the cosimulation result, waterfall model is modeled and simulated.

REFERENCES

[1] NVIDIA Corporation, NVIDIA CUDA: Compute Unified Device Architecture, Programming Guide, in, NVIDIA Corporation, Santa Clara, 2008.
[2] RecurDyn User Manual, http://eng.functionbay.co.kr/, (2010).
[3] P. A. Cundall and O. D. L. Strack, A Discrete Numerical Model for Granular Assemblies, Geotechnique, 29, 47-65 (1979).

[4] R. W. Hockney and J. W. Eastwood, Computer simulation using particles, McGraw-Hill, New York (1981).
[5] Mindlin, R.D. and Deresiewicz, H., Elastic Spheres in Contact under Varying Oblique Forces, Trans. ASME, J. Appl. Mech., 20, 327-344 (1953).
[6] Jalón JG, Bayo E, Kinematic and Dynamic Simulation of Multibody Systems, *Springer-Verlag New-York*, 1994

[7] Bae D. S., Han J. M., Choi J. H., and Yang S. M., A Generalized Recursive Formulation for Constrained Flexible Multibody Dynamics, *International Journal for Numerical Methods in Engineering*, Vol. 50, pp.1841-1859, 2001.

[8] García de Jalón D. J., Unda J., and Avello A., Natural coordinates for the computer analysis of multibody systems, *Computer Methods in Applied Mechanics and Engineering*, Vol. 56, pp.309-327, 1986.

[9] Wittenburg J, Dynamics of Systems of Rigid Bodies, *BF Teubner, Stuttgart*, 1977

[10] Brenan KE, Campbell SL, and Petzold LR, Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations, *North-Holland, New York*, 1989

[11] Haug EJ and Deyo R, Real-Time Integration Methods for Mechanical System Simulation, *Springer-Verlag, Berlin*, 1990

[12] Hairer E and Wanner G, Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems, *Springer, Berlin*, 1994

[13] Hughes TJR and Belytschko T, A precis of developments in computational methods for transient analysis, *ASME J. Appl. Mech*. 50, 1033–1041, 1983

[14] Chung J and Hulbert GM, A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-Alpha method. *J. Appl. Mech.*, 60, 371.375, 1993.

[15] Metariver, http://www.metariver.kr/, (2011).
[16] W. Mattson and B. M. Rice, Near-neighbor calculations using a modified cell-linked list method, Comput. Phys. Commun., Vol. 119, 135-148 (1999)