MULTIBODY DYNAMICS 2011, ECCOMAS Thematic Conference
J.C. Samin, P. Fisette (eds.)
Brussels, Belgium, 4-7 July 2011

A PARALLEL GPU IMPLEMENTATION OF THE ABSOLUTE NODAL COORDINATE FORMULATION WITH A FRICTIONAL/CONTACT MODEL FOR THE SIMULATION OF LARGE FLEXIBLE BODY SYSTEMS

Naresh Khude*, Dan Melanz*, Ilinca Stanciulescu† and Dan Negrut*

* Department of Mechanical Engineering
University of Wisconsin, 1513 University Avenue, Madison, WI, USA, 53706
e-mails: khude@wisc.edu, melanz@wisc.edu, negrut@engr.wisc.edu,
web page: http://www.sbel.wisc.edu

† Department of Civil and Environmetal Engineering
Rice University, 6100 Main Street MS-519, Houston, Texas, USA, 77005
e-mails: ilinca2@rice.edu

Keywords: Multibody Dynamics Simulation, Absolute Nodal Coordinate Formulation (ANCF), Discrete Element Method (DEM), Graphical Processing Unit (GPU), Gradient Deficient ANCF, Parallel Computing

Abstract. *This contribution discusses how a flexible body formalism, specifically, the Absolute Nodal Coordinate Formulation (ANCF), is combined with a frictional/contact model using the Discrete Element Method (DEM) to address many-body dynamics problems; i.e., problems with hundreds of thousands of rigid and deformable bodies. Since the computational effort associated with these problems is significant, the analytical framework is implemented to leverage the computational power available on today's commodity Graphical Processing Unit (GPU) cards. The code developed is validated against ANSYS and FEAP results. The resulting simulation capability is shown to result in one order of magnitude shorter simulation times for systems with a large number of flexible beams that might typically be encountered in hair or polymer simulation.*

1 THEORETICAL BACKGROUND - ANCF

For more than a decade the Absolute Nodal Coordinate formulation (ANCF) has been widely used to carry out the dynamics analysis of flexible bodies that undergo large rotation and large deformation. This formulation is consistent with the nonlinear theory of continuum mechanics and relatively easy to implement. Also, it leads to a constant mass matrix which makes it computationally more efficient compared to other nonlinear finite element formulations.

The fully parameterized ANCF beam element was originally introduced in [1]. The locking problems of Fully Parameterized ANCF finite elements based on the continuum mechanics approach have been addressed in the literature [2, 3]. These locking problems adversely impact the performance of ANCF finite elements especially for thin and stiff structures. Since the main focus of this work is to simulate the frictional contact problem between several slender beams, the original ANCF beam elements were not a good choice for this problem. Instead, the gradient deficient ANCF 3D beam elements, also referred to as low order cable elements in [3, 4], are used to model the slender beams. These are two node beam elements where one position vector and only one gradient vector are used as nodal coordinates $\mathbf{e}_i = [\mathbf{r}^T \ \mathbf{r}_x^T]_i^T$. Thus each node has 6 coordinates: three components of global position vector of the node and three components of position vector gradient at the node. It should be noted that the gradient deficient ANCF beam element does not describe a rotation of beam about its own axis so the torsional effects cannot be modeled [3]. This formulation shows no shear locking problems for thin and stiff beams and it is computationally efficient compared to the original ANCF due to reduced nodal coordinates.

The global position vector of an arbitrary point on the beam centerline is given by

$$\mathbf{r}(x, \mathbf{e}) = \mathbf{S}(x)\mathbf{e} \tag{1}$$

where $\mathbf{e} = \left[\mathbf{e}_1^T \ \mathbf{e}_2^T\right]^T \in \mathbb{R}^{12}$ is the vector of element nodal coordinates. The shape function matrix for this element is defined as $\mathbf{S} = [S_1\mathbf{I} \ S_2\mathbf{I} \ S_3\mathbf{I} \ S_4\mathbf{I}\ ] \in \mathbb{R}^{3\times 12}$ where $\mathbf{I}$ is the 3×3 identity matrix and the shape functions $S_j, j = 1,...,4$ are defined as [4]

$$\begin{aligned} s_1 &= 1 - 3\xi^2 + 2\xi^3, \quad s_2 = l(\xi - 2\xi^2 + \xi^3) \\ s_3 &= 3\xi^2 - 2\xi^3, \quad s_4 = l(-\xi^2 + \xi^3) \end{aligned} \tag{2}$$

Here $\xi = x/l,$ and l is the element length. Using the principle of virtual work for the continuum, the element equation of motion is obtained as:

$$\mathbf{M}\ddot{\mathbf{e}} + \mathbf{Q}_s = \mathbf{Q}_e \tag{3}$$

where $\mathbf{Q}_s$ is the vector of generalized element elastic forces, $\mathbf{Q}_e$ is the vector of generalized element external forces, and $\mathbf{M}$ is the symmetric consistent element mass matrix defined as

$$\mathbf{M} = A\int_0^l \rho \mathbf{S}^T\mathbf{S}dx \tag{4}$$

Here ρ and v are the element mass density and volume, respectively. The expression for the mass matrix given in (4) is derived using the virtual work of the inertia forces. Note that the element mass matrix is not a function of the time-dependent nodal coordinates.

The generalized element external force vector ($\mathbf{Q}_e \in \mathbb{R}^{12}$) due to gravity can be obtained as

$$\mathbf{Q}_e = A\int_0^l \mathbf{S}^T \mathbf{f}_g \, dx \tag{5}$$

where $\mathbf{f}_g = [0, -\rho g, 0]^T$ is the gravity force vector considering Y as the vertical axis. If a concentrated/point force is applied to an element at some point, the generalized element external force vector ($\mathbf{Q}_e \in \mathbb{R}^{12}$) in this case is obtained using the principle of virtual work as

$$\mathbf{Q}_e = \mathbf{S}^T \mathbf{f} \tag{6}$$

where $\mathbf{f}$ is an external point force and $\mathbf{S}$ is the shape function matrix defined at the point of application of the force.

The strain energy expression for the gradient deficient ANCF beam element is

$$U = \frac{1}{2}\int_0^l EA(\varepsilon_{11})^2 \, dx + \frac{1}{2}\int_0^l EI(\kappa)^2 \, dx \tag{7}$$

where $\varepsilon_{11} = \frac{1}{2}\left(\mathbf{r}_x^T \mathbf{r}_x - 1\right)$ is the axial strain and the magnitude of curvature vector κ is given as [4]

$$\kappa = \frac{\left|\mathbf{r}_x \times \mathbf{r}_{xx}\right|}{\left|\mathbf{r}_x\right|^3} \tag{8}$$

The vector of the element elastic forces ($\mathbf{Q}_s \in \mathbb{R}^{12}$) is determined from the strain energy expression as

$$\mathbf{Q}_s = \int_0^l EA(\varepsilon_{11})\left(\frac{\partial \varepsilon_{11}}{\partial \mathbf{e}}\right)^T dx + \int_0^l EI(\kappa)\left(\frac{\partial \kappa}{\partial \mathbf{e}}\right)^T dx \tag{9}$$

For the gradient deficient ANCF beam element, the equation of motion and the expressions for element mass matrix and element external force are the same as in case of a fully parameterized ANCF beam element. However, computing the element elastic force is much easier in the former case. Since in the shape functions only one spatial coordinate (ξ) is used, the numerical integration is carried out using the Gauss-quadrature formula in one dimension only.

2 THEORETICAL BACKGROUND - DEM

The Discrete Element Method (DEM) was proposed by Cundall to model the mechanical behavior of granular material [5, 6]. The DEM can be classified as a penalty method, where the force acting between two colliding bodies is computed based on the associated interpenetration. DEM approaches have been widely used in rock mechanics, molecular dynamics, and granular dynamics simulation. The basic idea behind the DEM frictional contact model is to introduce a fictitious spring-damper element that is placed between two bodies when they collide. The spring and damping coefficients can be derived from the continuum theory [7] or calibrated based on experimental data. The schematic of DEM contact is shown in Figure 1.

Numerous contact force models have been developed over the past decades [8-11]. This paper uses the volumetric-based Coulomb friction contact model introduced in [12]. In this model, the normal contact force $\mathbf{F}_n$ is given as

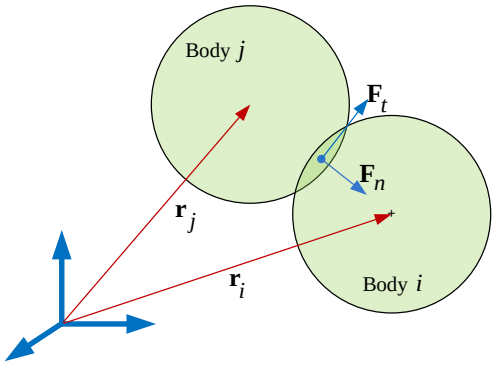

Figure 1: Schematic of DEM contact

$$\mathbf{F}_n = \frac{k_f}{h_f} V(1 + a\mathbf{v}_{cn})\boldsymbol{n} \tag{10}$$

where k_f and h_f are the elastic modulus and depth of the Winkler elastic foundation [12], respectively; $\mathbf{v}_{cn}$ is the relative velocity of the colliding spheres in normal ($\boldsymbol{n}$) direction; V is total penetration volume, and the damping term a is a function of the coefficient of restitution and of the initial impact velocity, see [13]. The friction model presented in [13] is used where the Coulomb friction force $\mathbf{F}_t$ is given as

$$\mathbf{F}_t = -\mu_c \mathbf{F}_n dir_\varepsilon(\mathbf{v}_{ct}, \upsilon_\varepsilon) \tag{11}$$

where μ_c is the Coulomb friction coefficient, $\mathbf{v}_{ct}$ is the relative tangential velocity of the colliding spheres, υ_ε is a velocity tolerance, and the term $dir_\varepsilon(\mathbf{v}_{ct}, \upsilon_\varepsilon)$ is defined as

$$dir_\varepsilon(\mathbf{v}_{ct}, \upsilon_\varepsilon) = \begin{cases} \dfrac{\mathbf{v}_{ct}}{|\mathbf{v}_{ct}|}; & |\mathbf{v}_{ct}| \geq \upsilon_\varepsilon \\ \dfrac{\mathbf{v}_{ct}}{\upsilon_\varepsilon}\left(\dfrac{3}{2}\dfrac{|\mathbf{v}_{ct}|}{\upsilon_\varepsilon} - \dfrac{1}{2}\left(\dfrac{|\mathbf{v}_{ct}|}{\upsilon_\varepsilon}\right)^3\right); & |\mathbf{v}_{ct}| < \upsilon_\varepsilon \end{cases} \tag{12}$$

The DEM approach can be easily implemented with the explicit numerical integration method; however, it requires a very small integration step-size to maintain the stability and accuracy of the numerical solution. The integration step size depends on the size of the spheres, the material properties associated with the bodies in contact, and the relative velocity of the spheres. Because of the rigid body assumptions, the stiff springs induced by the DEM model can lead to high transients in the system dynamics.

3 SEQUENTIAL IMPLEMENTATION OF ANCF AND DEM

In DEM the colliding bodies are assumed to be rigid. To extend this approach for the frictional contact between slender beams, a spherical decomposition of the colliding beams is implemented in this paper. Specifically, as shown in Figure 2, each flexible beam can be considered as a chain of spheres that overlap and which are distributed equally along the axis of the gradient deficient ANCF beam elements. At each time step the collision detection between the spheres in all the beams is performed and the normal and tangential contact forces between the colliding spheres are calculated using DEM. The spherical decomposition approach along

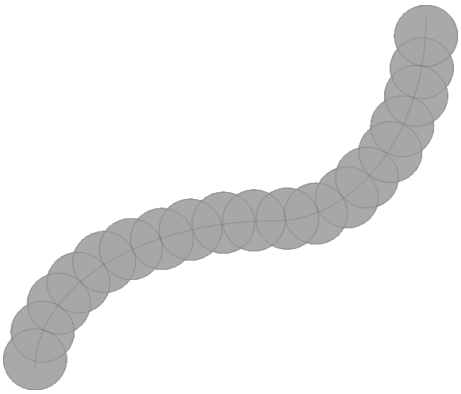

Figure 2: Spherical decomposition along the centerline of deformed beam

with DEM provides the magnitude and direction of the contact force and its location relative to the beam element. These contact forces are treated as externally applied point forces in the nonlinear dynamics analysis with ANCF. For thin beams the moments of tangential/frictional contact forces about the beam centerline can be neglected. It should be noted that this approach allows multiple contacts between the colliding beams and also self-contact in case of a long, highly deformable beam.

The spherical decomposition approach simplifies the process of collision detection between the slender beams. The collision detection problem can be a bottleneck in the simulation of physical systems involving a large number of bodies. For the hair simulation problem one has to consider systems with hundreds of thousands of flexible beams interacting through frictional contact, for which collision detection between millions of spheres needs to be done at each time step. A parallel implementation on the GPU provides significant speed up for such problem (see sec. 4.4).

A sequential ANCF and DEM algorithm has been implemented using both MATLAB and C. Originally programmed to handle only one to two beams, the sequential version is used for validation and debugging purposes. Subsequent versions of the sequential implementation can handle multiple beams, used for benchmarking. There are two main stages to the serial implementation: a preprocessing stage and a time stepping loop.

3.1 Preprocessing stage

The preprocessing stage of the ANCF algorithm prepares the data required for solving the degrees of freedom of the beam at each time step in the time loop. To this end, there are five vectors and one matrix initialized in the preprocessing stage. A square mass matrix with dimensions equal to the number of degrees of freedom of the beam is initialized using the steps shown in Figure 3. The mass matrix is constant and does not need to be recomputed at every time step. Following the mass matrix, an external force vector due to gravity is initialized. This part of external force vector contains only forces due to gravity and remains constant over time. Next, the position vector, which holds the degrees of freedom of the beam, is created using the steps in Figure 3 to represent the initial location and orientation of the beam. Lastly, the internal force, acceleration, and velocity vectors, each with a size equal to the number of degrees of freedom of the beam, are initialized to zero.

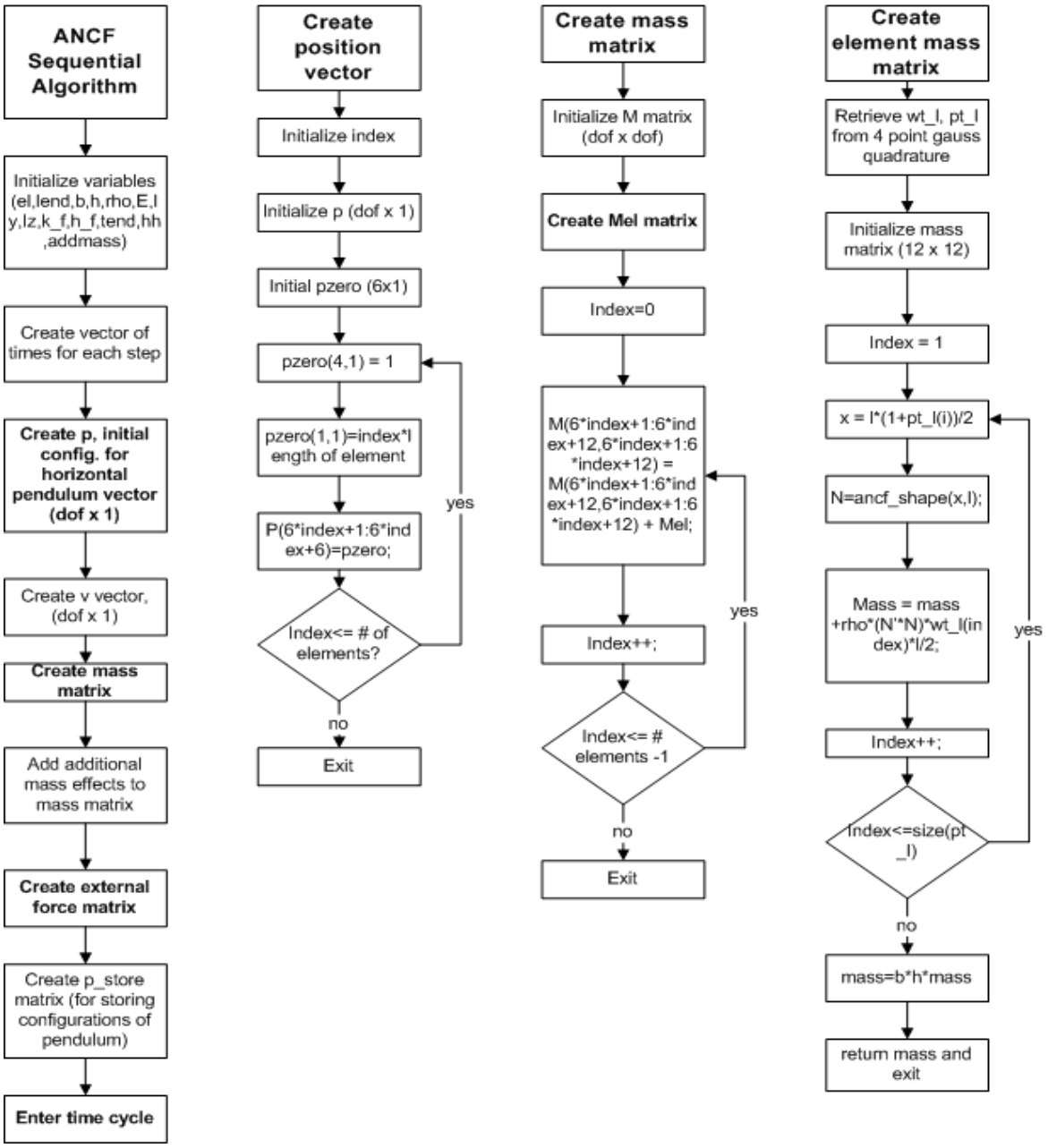

Figure 3: Preprocessing stage of the ANCF algorithm.

3.2 Time stepping loop

After the preprocessing stage, the contact external force vector, the internal force vector, acceleration vector, velocity vector, and position vector need to be updated at every time step. At each time step the collision detection between the spheres is performed and the contact external force vector is computed using DEM as explained before. The internal force vector is computed using Eq. (9), which is by far the most computationally intensive portion of the algorithm. The equation of motion (Eq. (3)) is solved to obtain the acceleration vector and the position and velocity vector are updated using an explicit numerical integrator. This position and velocity data is used in the next time step to compute the contact external force vector and the internal force vector; and the cycle is repeated as shown in Figure 4. In this paper the symplectic Euler integration method is used for the simulation as it is more stable than the regular forward Euler method.

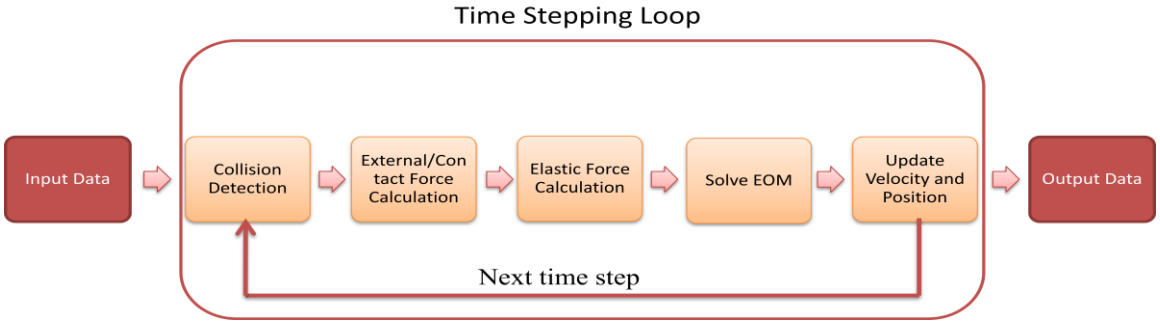

Figure 4: Time looping stage of the ANCF and DEM algorithm.

4 PARALLEL IMPLEMENTATION ON GPU

Being computationally very intensive, the DEM and ANCF methodologies stand to benefit from the use of parallel computation. In the simulation of complex mechanical systems with many flexible beams (e.g., hair or polymer simulation), the equations of motion of each beam can be solved in parallel. The computation of the nonlinear internal force and the external force can also be done in parallel at element level. All these aspects are anticipated to lead to significant reductions in simulation times for large flexible body systems.

4.1 Overview of GPU hardware and software

Over the last five years it has become apparent that future increases in computational speed are not going to be fueled by advances in sequential computing technology. There are three main walls that the sequential computing model has hit [14]. Firstly, there is the power dissipation wall caused by the amount of energy that is dissipated per unit area by ever smaller transistors. On a per unit area basis, the amount of energy dissipated by a Pentium 4 processor comes slightly short of that associated with a nuclear power plant. Since the amount of power dissipated scales with the square of the clock frequency, steady further clock frequency increases, which in the past were responsible for most of the processing speed gains, are unlikely. Forced cooling solutions could increase absolute clock rates, but come at a prohibitively high price and cannot trump a general trend.

The second wall, that is, the memory wall, arose in sequential computing as a manifestation of the gap between processing power and memory access speed, a gap that grew wider over the last decade. A single powerful processor will likely become data starved, idling while information is moved back and forth between the chip and RAM over a bus typically clocked at 10 to 30 GB/s. Ever larger caches alleviate the problem, yet technological and cost constraints associated with large caches cannot reverse this trend. This aspect will most likely be addressed by a disruptive technology such as photonic integrated circuits that promise to provide a new solution to bandwidth demand for on/off-chip communications [15].

Thirdly, investments in Instruction Level Parallelism (ILP), which mainly draws on instruction pipelining, speculative execution, and branch prediction to speed up execution, represent an avenue that has already exhausted its potential. Both processor and compiler designers capitalized on opportunities for improved performance in sequential computing, which came at no cost to the software developer. Augmenting branch prediction and speculative execution beyond current instruction horizons comes at a complexity/power price that in many cases increases exponentially with horizon depth.

While the sequential computing model paradigm seems to have lost, at least temporarily, its momentum, increases in flop rate over the next decade are guaranteed by vigorous miniaturization rates. Moore's law is conjectured to remain valid over the next decade, with 22 nm

CMOS manufacturing technology slated to be made available by Intel in 2011, 15 nm in 2013, 11 nm in 2015, and 8 nm in 2017 [16]. This packing of more transistors per unit area, which translates into having more cores per chip, stands to address in the immediate future demands for higher flop rates and larger memory sizes in Scientific Computing. The success stories of this trend are the recent chip designs, code name Fermi and Knights, released by NVIDIA and Intel, respectively. The former packs approximately three billion transistors to lead to co-processors with 512 cores. Knights Ferry, announced in mid 2010, packs 32 cores in what is predicted to be the first in a family of products that belong to Intel's Many Integrated Core (MIC) architecture vision which is slated to deliver using 22 nm technology a 50 core Knights Corner co-processor in 2011.

Originally designed for handling the computations involved in real-time, high-definition 3D graphics, the Graphic Processor Unit, or GPU, is ideal for problems that can be represented as data-parallel computations. As long as the same sequence of operations is executed for each data element and branches are kept to a minimum, the GPU's memory latency can be hidden with arithmetic calculations. As shown in Figure 5, more transistors are devoted on the GPU to data processing rather than data caching and control flow. To make use of this computational power, NVIDIA introduced a general purpose parallel computing architecture, called CUDA. The CUDA parallel programming model makes it easy to exploit the parallelism in a program by using C programming and drawing on a minimal set of language extensions.

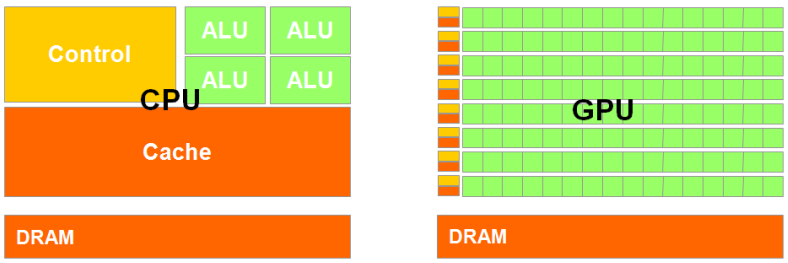

Figure 5: The GPU is specialized for compute-intensive, highly data parallel computation due to its graphics rendering origin. This manifested in microprocessor design that have a very large number of ALUs.

4.2 Memory organization

As described in the sequential algorithm section of this report, there are several opportunities for parallelism in the ANCF algorithm. In the case of multiple beams, the corresponding vectors and matrices are stacked on top of each other in memory, shown in Figure 6, to create a global and contiguous data segment for each of the important arrays (position, velocity, acceleration, mass, etc.). It is intended that organizing the code in this manner improves code clarity and eases the memory accesses of the GPU. Parallel implementations for the internal force computation, acceleration update function, and position/velocity update functions have been created.

4.3 Parallel implementation of ANCF

Sequential Preprocessing Stage

Initialize position, velocity, acceleration vectors. Create external force vector and constant mass matrix.

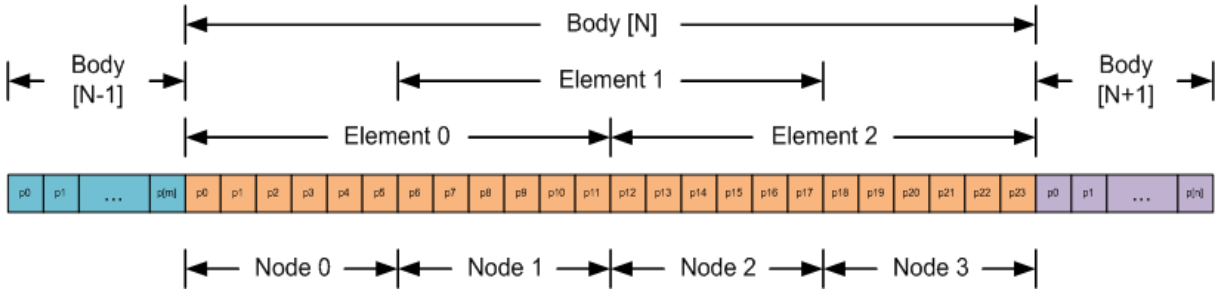

Figure 6: Memory representation of the global position vector. Degrees of freedom are organized into nodes, which are overlapped to represent elements and bodies.

Internal Force Computation

The calculation of each of the element internal force vectors is independent of the body internal force vector. Aside from the fact that half of one element internal force vector is added to half of the adjacent element internal force vector, the body internal force vector could be calculated entirely in parallel, element-wise. To get around this overlap, the internal force vector is parallelized on a nodal basis, with each node depending on the 18 degrees of freedom from the two elements that comprise it. A node lookup table is created during the preprocessing stage of the program to communicate the internal force function inputs to each thread. Due to the large amounts of computation that go into calculating the internal force for a single node, there is a large amount of register pressure, with a significant amount of data being pushed onto local memory. Efforts to reduce register usage include splitting the kernel into several sub-kernels and employing shared memory to store the position and nodal property inputs.

Acceleration Update Computation

The acceleration update must be done on a per body basis due to the dense nature of the inverse of the mass matrix. Each thread calculates the acceleration for one body by multiplying the body's mass matrix by the corresponding external minus internal force vector. Since the same force values are being called repeatedly to multiply the elements of the mass matrix, the internal force vector and external force vector are loaded into shared memory.

Position and Velocity Update Computation

The position and velocity update computations are composed of multiplying a scalar value (the time step) by a vector (the previous position/velocity vectors). This particular computation is termed "embarrassingly parallel" and each thread can handle one entry of the global position/velocity vector.

Return to host and increment time

Perform data copying/processing and branch to internal force computation.

4.4 Collision detection example

A first set of numerical experiments gauged the efficiency of the parallel collision detection algorithm developed. The reference used was a *sequential* implementation from Bullet Physics Engine, an open source physics-based simulation engine [17]. The CPU used in this experiment (relevant for the Bullet results) was AMD Phenom II Black X4 940, a quad core 3.0 GHz processor that drew on 16 GB of RAM. The GPU used was NVIDIA's Tesla C1060. The operating system used was the 64-bit version of Windows 7. The test was meant to gauge the relative speedup gained with respect to the serial implementation. This test stopped when

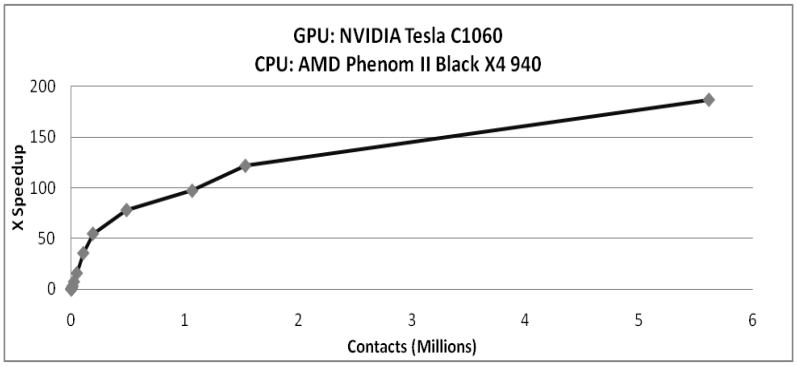

Figure 7: Overall speedup when comparing the CPU algorithm to the GPU algorithm. The maximum speedup achieved was approximately 180 times [18].

dealing with about six million contacts (see horizontal axis of Figure 7), when Bullet ran into memory management issues. The plot illustrates a 180 fold relative speedup when going from sequential Bullet to the in-house developed GPU-based parallel implementation.

5 NUMERICAL EXPERIMENTS AND RESULTS

Several numerical experiments are carried out in order to *(a)* Validate the gradient deficient ANCF against FEAP [19]and ANSYS; *(b)* Compare the gradient deficient ANCF beam elements against fully parameterized ANCF beam elements; *(c)* Assess the potential of the ANCF and DEM implementation; *(d)* Carry out a convergence analysis of the gradient deficient ANCF beam elements; and *(e)* Assess the computation time-speedup obtained due to the parallel GPU implementation. The details of the models used in the numerical experiments and the simulation parameters are given in Table 1. For all the numerical experiments the flexible pendulums/beams are assumed to be in horizontal configuration initially with no initial velocity.

5.1 Validation of the gradient deficient ANCF elements

Model 1 is used to study the motion of a highly deformable 3D pendulum (beam pinned at one end) under the effect of gravity. The pendulum tip displacements are compared with those using 40 solid elements (27-node) in FEAP [19]. Figure 8 shows that the results are in good agreement and the gradient deficient ANCF beam elements do not suffer from shear locking problems. The symplectic Euler integrator with the step size of 1.0*e*-4 *sec* is used in this simulation. Figure 9 shows the total energy using gradient deficient ANCF elements remains constant as the system is conservative. It also shows that the explicit integration scheme used in this simulation remains stable. The tip displacements of the flexible pendulum are compared with those obtained using fully parameterized ANCF beam elements. Figure 10 shows considerable difference between the two results which is attributed to the locking problem of original ANCF beam elements in case of thin beams. Model 1 is also used for nonlinear dynamics analysis in ANSYS using BEAM4 elements. The ANSYS results using 64 BEAM4 elements are compared with those using 16 gradient deficient ANCF beam elements in Figure 11. The difference in the results can be attributed to the linear strain functions of BEAM4 elements in ANSYS [20].

Parameters	Model 1	Model 2	Model 3
Length (m)	1	1	3
Cross-section Area(m^2)	0.02 x 0.02	π x 0.01^2	π x 0.01^2
Material Density (kg/ m^3)	7200	7200	7200
Modulus of elasticity (Pa)	2.0E7	2.0E7	2.0E7
Second moment of inertia (m^4)	1.33*e*-8	7.85*e*-10	7.85*e*-10
Tip Mass (kg)	--	--	5
Number of 3D beam elements	4	1	8
External Force	Gravity	Gravity + Contact	Gravity + Contact
Integration Step-size	1.0E-4	1.0E-5	1.0E-5

Table 1: The model and simulation parameters

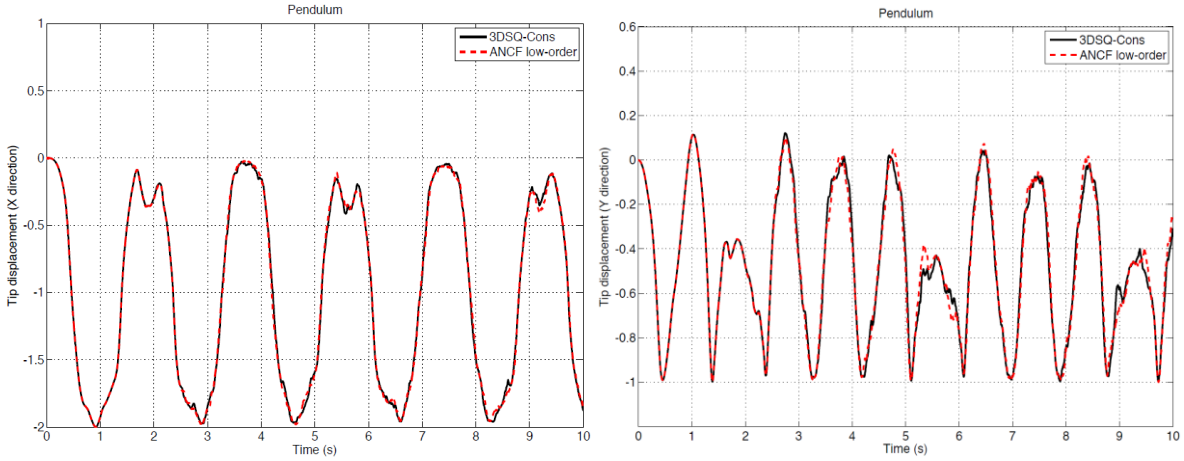

Figure 8: X and Y displacement of a pendulum-tip (ANCF and FEAP comparison)

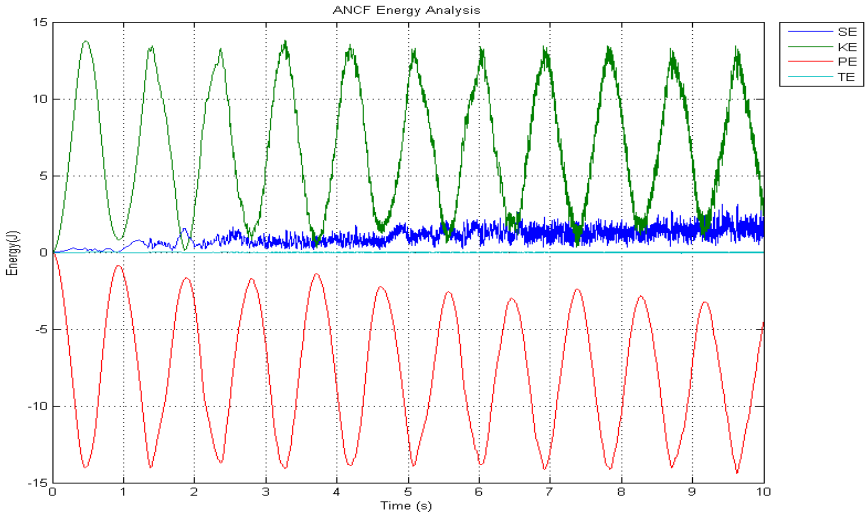

Figure 9: Energy analysis – GD ANCF (SE: strain energy, KE: kinetic energy, PE: potential energy, TE: total energy)

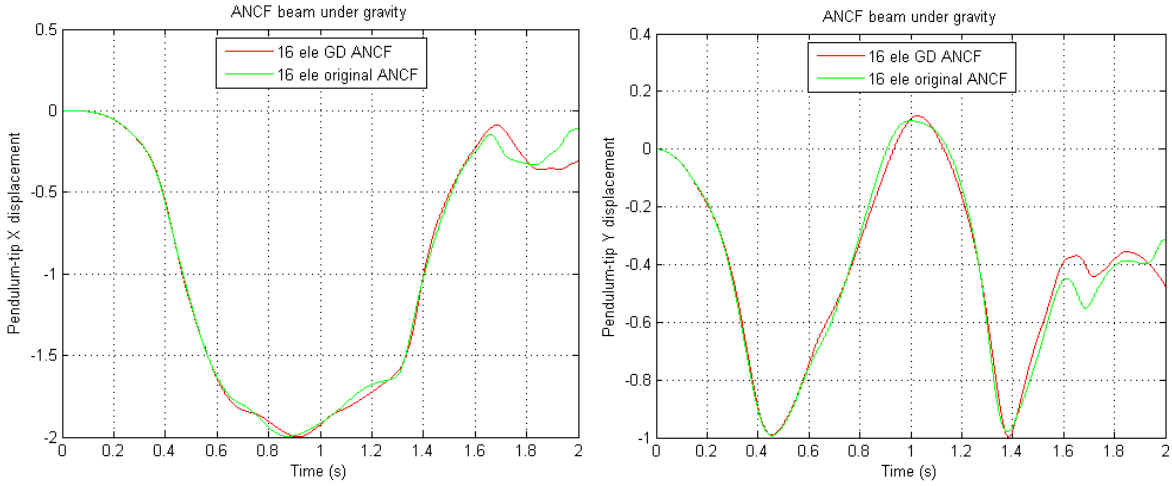

Figure 10: X and Y displacement of a pendulum-tip (ANCF and GD ANCF comparison)

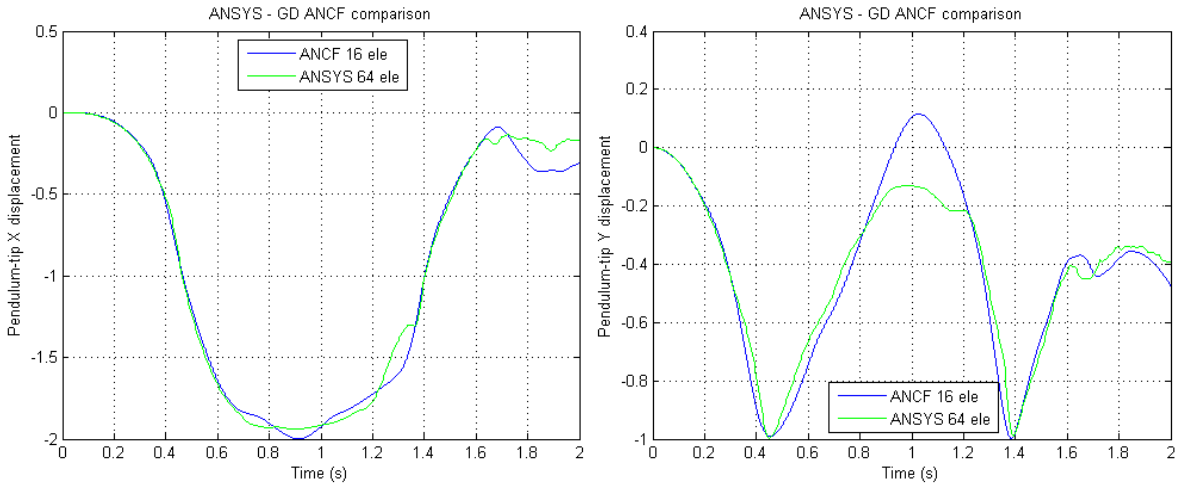

Figure 11: X and Y displacement of a pendulum-tip (ANSYS and GD ANCF comparison)

5.2 ANCF with frictional contact

In the second set of numerical experiments the frictional contact phenomenon between the flexible beams (pinned at one end) is demonstrated using Model 2. For the contact force model, the Coulomb friction coefficient $\mu_c = 0.3$ is used. The damping factor a was selected to obtain a coefficient of restitution of 0.975 so that a small amount of energy is dissipated through damping. The contact stiffness k_f / h_f used in this model is 2.0e9. Snapshots of the numerical simulation of this model are shown in Figure 12.

Model 3 is used to simulate the frictional contact between a long flexible beam (pinned at one end) with a tip mass and a rigid cylinder. Figure 13 shows a sequence of three snapshots of the numerical simulation of this model. It should be noted that the DEM approach used in this paper allows the contact force modeling between the rigid bodies and the flexible beams. Figure 14 shows the snapshots of simulation of 8 instances of Model 3. In this parametric study, the number of elements varied in each beam from one element to eight elements. A rigid horizontal cylinder with radius of 0.01 m was fixed at position (1, -1.5) so that the elements would come into contact with the rigid cylinder as they moved. The beams tend to

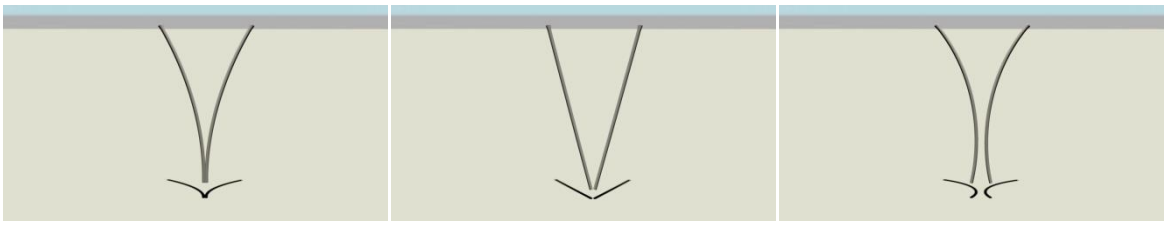

Figure 12: Snapshots from simulation of Model 2

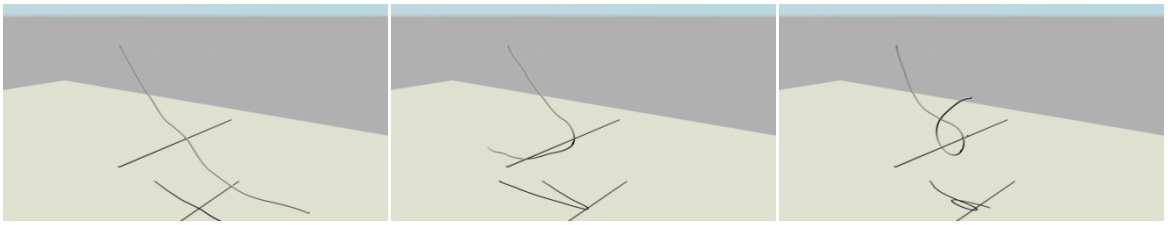

Figure 13: Snapshots from simulation of Model 3

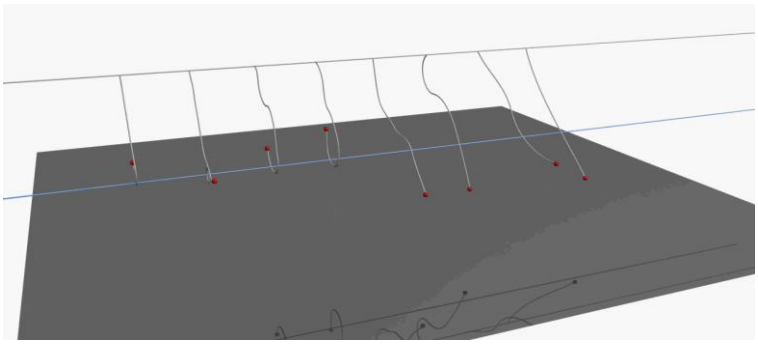

Figure 14: Snapshot from parametric study of model 3.

wrap around the rigid cylinder due to inertia of the tip mass. The overall bending stiffness of the beam tends to decrease as the number of elements increases and hence different trajectories of the beam tip are observed, as seen in Figure 14. The videos of these simulations are available at [21].

5.3 Convergence analysis of ANCF and DEM

In order to investigate the convergence of the gradient deficient ANCF elements a Model 2 beam (pinned at one end) with a tip-mass of 1kg is used in this analysis. The beam is modeled using various numbers of elements and the pendulum tip positions are compared for a 4 *sec* long simulation. Figure 15 shows that as the number of elements is increased, the results tend to converge to a single value. The gradient deficient ANCF beam elements exhibit good convergence characteristics. Thus the large deformation of slender beams can be effectively modeled with very few gradient deficient ANCF beam elements.

It is also important to investigate the convergence characteristics of the ANCF with frictional contact. In this analysis the same model is used which comes into contact with the rigid cylinder fixed at position (0.5, -0.5). From Figure 16, it can be seen that the pendulum tip

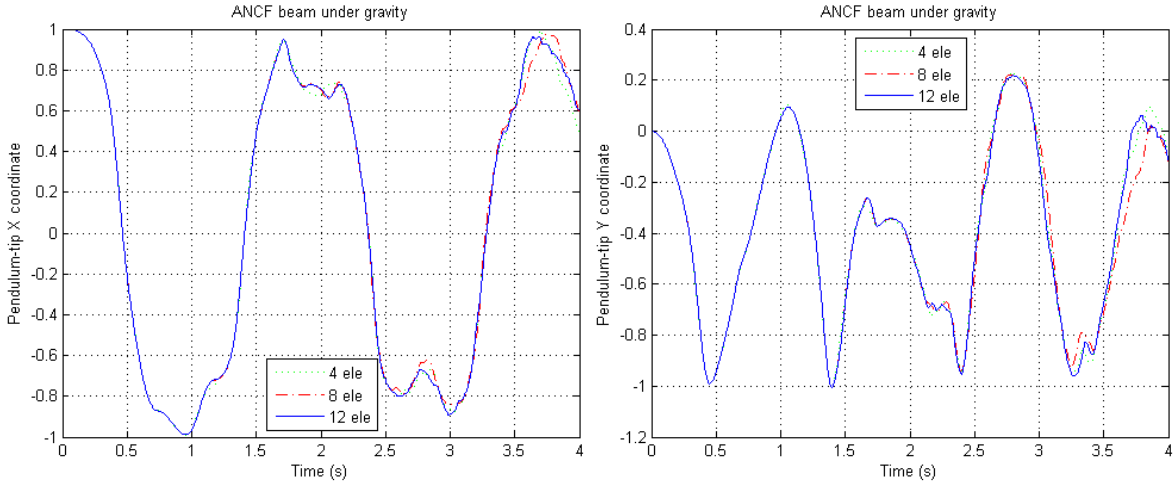

Figure 15: X and Y coordinate of a pendulum-tip (non-contact case)

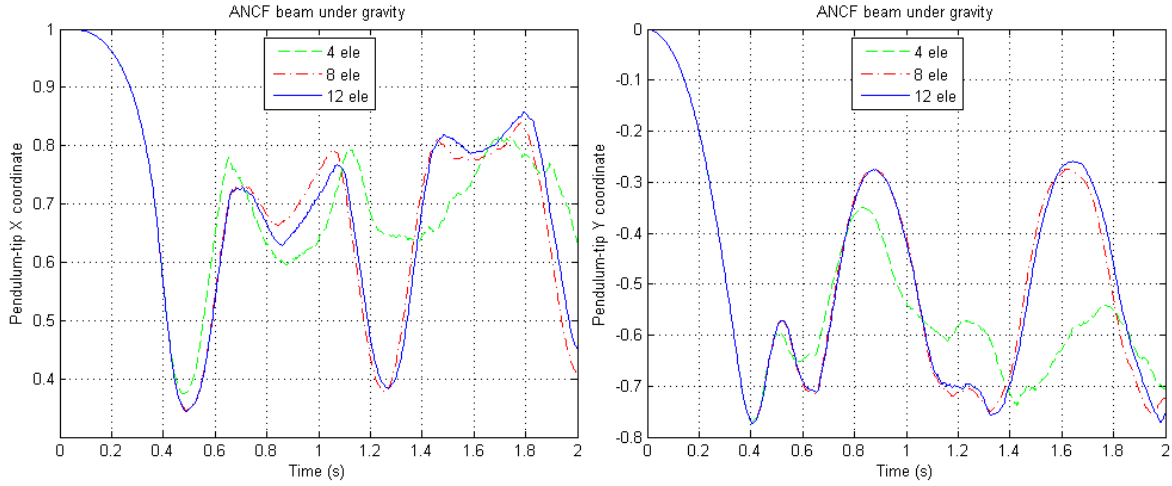

Figure 16: X and Y coordinate of a pendulum-tip (contact case)

displacements tend to converge as the number of elements is increased, which shows that the convergence characteristics of ANCF are not affected when combined with DEM approach.

5.4 Comparison of GPU and CPU computation times

In this simulation a complex mechanical system containing hundreds of thousands of flexible beams pinned at one end and with different initial conditions is used. Model 2 beams with 8 elements are used in this analysis. It is assumed that the beams do not come into contact with each other. Several instances of the CPU and GPU implementations were run on an Intel Nehalem Xeon E5520 2.26GHz processor with an NVIDIA Tesla C1060 graphics card for varying numbers of beams. On average, a 15x speedup was observed from the GPU implementation over the CPU implementation, shown in Figure 17.

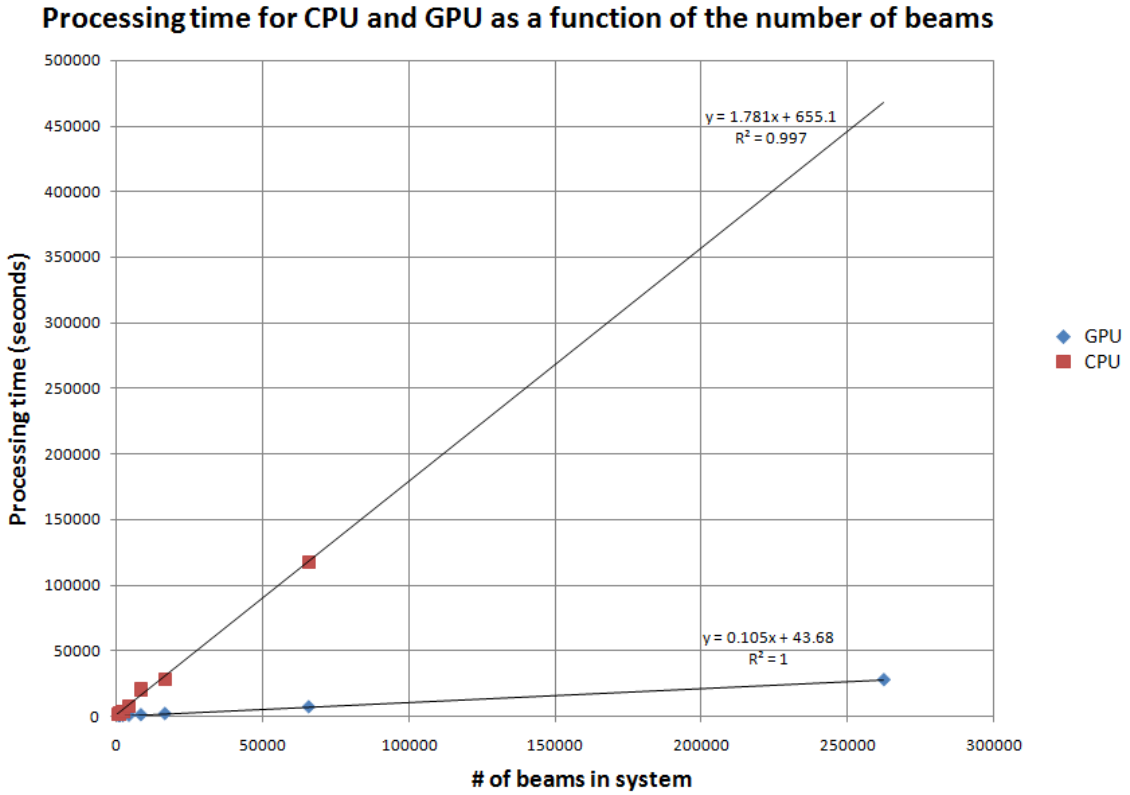

Figure 17: Processing time for the CPU and GPU implementations for varying numbers of beams.

6 CONCLUSIONS

This paper presents a methodology for combining DEM with ANCF to model the frictional contact between highly deformable thin beams. The gradient deficient ANCF beam elements used in the dynamics analysis exhibit good convergence characteristics and do not suffer from shear locking problems. The dynamics analysis results of thin beams are validated against FEAP and compared with ANSYS results.

A spherical decomposition approach is used to simplify the contact detection between beams. Both the DEM and ANCF methodologies stand to benefit from the use of parallel computation. The potential of parallel computation at the element level and at the body level in ANCF has been investigated and shown to lead on average to a 15 fold speedup when drawing on GPU computing.

The scaling results for systems with hundreds of thousands of flexible bodies (e.g., hair or polymer simulation) show that the GPU simulation approach proposed has the potential to increase the relevance of flexible multibody dynamics in addressing challenging real-life design problems across a spectrum of engineering disciplines.

ACKNOWLEDGEMENTS

The last author was financially supported in part by the National Science Foundation grant NSF-CMMI-0840442 and Function Bay, Inc.

REFERENCES

[1] Shabana, A. and R. Yakoub, *Three dimensional absolute nodal coordinate formulation for beam elements: Theory*. Journal of Mechanical Design, 2001. **123**: p. 606.

[2] Schwab, A. and J. Meijaard. *Comparison of three-dimensional flexible beam elements for dynamic analysis: finite element method and absolute nodal coordinate formulation*. in *Proceedings of the ASME 2005 IDETC/CIE*. November 5- 11, 2005. Orlando, Florida.

[3] Gerstmayr, J. and A. Shabana, *Analysis of thin beams and cables using the absolute nodal co-ordinate formulation.* Nonlinear Dynamics, 2006. **45**(1): p. 109-130.

[4] Shabana, A.A., *Computational continuum mechanics*. 2008: Cambridge University Press, New York.

[5] Cundall, P. and O. Strack, *A discrete element model for granular assemblies.* Geotechnique, 1979. **29**(1): p. 47-65.

[6] Cundall, P., *A Computer Model for Simulating Progressive, Large Scale Movements in Blocky Rock Systems. Symp.* Int. Soc. Rock Mechanics, Nancy, 1971.

[7] Johnson, K.L., *Contact Mechanics*. 1987, Cambridge: University Press.

[8] Rapaport, D., *Simulational studies of axial granular segregation in a rotating cylinder.* Physical Review E, 2002. **65**(6): p. 61306.

[9] Rapaport, D., *Radial and axial segregation of granular matter in a rotating cylinder: A simulation study.* Physical Review E, 2007. **75**(3): p. 31301.

[10] Silbert, L., D. Erta , G. Grest, T. Halsey, D. Levine, and S. Plimpton, *Granular flow down an inclined plane: Bagnold scaling and rheology.* Physical Review E, 2001. **64**(5): p. 51302.

[11] Landry, J., G. Grest, L. Silbert, and S. Plimpton, *Confined granular packings: structure, stress, and forces.* Physical Review E, 2003. **67**(4): p. 41303.

[12] Gonthier, Y., J. McPhee, and C. Lange. *On the implementation of coulomb friction in a volumetric-based model for contact dynamics*. in *Proceedings of the ASME 2007 IDETC/CIE*. September 4-7, 2007. Las Vegas, NV.

[13] Gonthier, Y., J. McPhee, C. Lange, and J. Piedboeuf, *A regularized contact model with asymmetric damping and dwell-time dependent friction.* Multibody System Dynamics, 2004. **11**(3): p. 209-233.

[14] Manferdelli, J.L., *The Many-Core Inflection Point for Mass Market Computer Systems.* CTWatch Qtrly, 2007. **3**(1).

[15] Chan, J., G. Hendry, A. Biberman, and K. Bergman, *Architectural Exploration of Chip-Scale Photonic Interconnection Network Designs Using Physical-Layer Analysis.* J. Lightwave Technol., 2010. **28**: p. 1305-1315.

[16] Skaugen, K., *Petascale to Exascale: Extending Intel's HPC Commitment:* *http://download.intel.com/pressroom/archive/reference/ISC_2010_Skaugen_keynote.pdf* in *International Supercomputer Conference*. 2010.

[17] Erwin, C. *Physics Simulation Forum*. 2010 [cited 2010 January 15]; Available from: http://www.bulletphysics.com/Bullet/wordpress/.

[18] Mazhar, H., T. Heyn, and D. Negrut, *A Scalable Parallel Method for Large Collision Detection Problems.* Multibody System Dynamics, 2011: p. 1-19.

[19] Taylor, R.L. *A Finite Element Analysis Program, Version 8.3, 2010*.

[20] Yakoub, R.Y. and A.A. Shabana, *Three dimensional absolute nodal coordinate formulation for beam elements: implementation and applications.* Journal of Mechanical Design, 2001. **123**: p. 614.

[21] SBEL. *Simulation-Based Engineering Lab, Department of Mechanical Engineering, University of Wisconsin, Madison:* http://sbel.wisc.edu/Animations/index.htm. 2010 [cited 2010 January 27].