Proceedings of the ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2012
August 12-15, 2012, Chicago, IL, USA

DETC2012-71352

A GPU Parallelization of the Absolute Nodal Coordinate Formulation for Applications in Flexible Multibody Dynamics

Daniel Melanz
Dept. of Mechanical Engineering
University of Wisconsin – Madison
Madison, WI 53706

Naresh Khude
Dept. of Mechanical Engineering
University of Wisconsin – Madison
Madison, WI 53706

Paramsothy Jayakumar
US Army Tank Automotive Research, Development, and Engineering Center
Warren, MI 48397

Mike Leatherwood
US Army Tank Automotive Research, Development, and Engineering Center
Warren, MI 48397

Dan Negrut
Dept. of Mechanical Engineering
University of Wisconsin – Madison
Madison, WI 53706

ABSTRACT

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 is computationally more efficient compared to other nonlinear finite element formulations. Kinematic constraints that represent mechanical joints and specified motion trajectories can be introduced to make complex flexible mechanisms. As the complexity of a mechanism increases, the system of differential algebraic equations becomes very large and results in a computational bottleneck. This contribution helps alleviate this bottleneck using three tools: (1) an implicit time-stepping algorithm, (2) fine-grained parallel processing on the Graphics Processing Unit (GPU), and (3) enabling parallelism through a novel Constraint-Based Mesh (CBM) approach. The combination of these tools results in a fast solution process that scales linearly for large numbers of elements, allowing meaningful engineering problems to be solved.

THEORETICAL BACKGROUND

The Absolute Nodal Coordinate Formulation (ANCF)

For almost 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 is 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 significantly deteriorate the performance of ANCF finite elements especially for thin and stiff structures. To avoid these issues, the gradient deficient ANCF 3D beam elements, also referred 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 $\mathbf{e}_i = [\mathbf{r}^T \ \mathbf{r}_x^T]^T$. 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]. However, 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

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

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 3x3 identity matrix and the shape functions $S_j, j = 1,...,4$ are defined as [4]

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

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

$$\mathbf{M}\ddot{\mathbf{e}} + \mathbf{Q}_s = \mathbf{Q}_e \quad (3)$$

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

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

Here ρ and A are the element mass density and cross sectional area, 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

$$\mathbf{Q}_e = A\int_0^l \mathbf{S}^T\mathbf{f}_g dx \quad (5)$$

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

$$\mathbf{Q}_e = \mathbf{S}^T\mathbf{f} \quad (6)$$

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

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

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]

$$\kappa = \frac{|\mathbf{r}_x \times \mathbf{r}_{xx}|}{|\mathbf{r}_x|^3} \quad (8)$$

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

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

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. Computing the element elastic force is much easier in this case. Since only one spatial coordinate (ξ) is used in the shape functions, the numerical integration is carried out using the Gauss-quadrature formula in one dimension only.

Flexible Mechanical Systems with Constraints

The kinematic constraints impose restrictions on the relative motion of the bodies in a mechanical system. These constraints are the algebraic equations of the form

$$\mathbf{\Phi}(\mathbf{q},t) = [\Phi_1(\mathbf{q},t)...\Phi_m(\mathbf{q},t)]^T = 0 \quad (10)$$

where m is the total number of independent constraint equations that must be satisfied by the generalized coordinates $\mathbf{q} = [\mathbf{q}_1^T ... \mathbf{q}_n^T]^T \in \mathbb{R}^p$. Here n is the total number of bodies and p is the total number of coordinates present in the system. If each body in the system is assumed to be the beam then the generalized coordinates of each body (beam) are defined as $\mathbf{q}_b = [\mathbf{e}_1^T ... \mathbf{e}_{ele}^T]^T$; $\mathbf{e}_1, ..., \mathbf{e}_{ele} \in \mathbb{R}^{12}$ where ele is number of ANCF

beam elements used in beam b. Several types of mechanical joints can be easily modeled in ANCF. Some of these joints and their associated constraints are as follows: The spherical joint between two nodes of any two bodies will require the position vector of each node to be identical. The revolute joint will have additional two constraints to the spherical joint constraints. In this case, the gradient vectors of the two nodes will remain in a plane perpendicular to the axis of revolute joint. There are also additional constraints due to the element connectivity in each beam. The element connectivity can be modeled as a fixed joint between the nodes. Here the common node between two elements is treated as two different nodes attached to each other through the fixed joint. This fixed joint requires all the nodal coordinates of the two nodes be identical.

The generalized coordinates of the system change in time under the effect of applied forces such that these constraint equations are satisfied at all times. The time evolution of the system is governed by the Lagrange multiplier form of the constrained equations of motion

$$\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^{T}(\mathbf{q},t)\boldsymbol{\lambda} + \mathbf{Q}_{\text{int}}(\mathbf{q}) = \mathbf{Q}_{\text{ext}}(\dot{\mathbf{q}},\mathbf{q},t) \tag{11}$$

where $\mathbf{M} \in \mathbb{R}^{p\times p}$ is the generalized mass, a constraint Jacobian matrix is $\boldsymbol{\Phi}_{\mathbf{q}} = \left[\frac{\partial \Phi_i}{\partial q_j}\right]$, for $1 < i < m, 1 < j < p$ and $\mathbf{Q}_{\text{ext}}(\dot{\mathbf{q}},\mathbf{q},t) \in \mathbb{R}^{p}$ is the applied force on the generalized coordinates $\mathbf{q} \in \mathbb{R}^{p}$ and $\mathbf{Q}_{\text{int}}(\dot{\mathbf{q}},\mathbf{q},t) \in \mathbb{R}^{p}$ is the vector of generalized elastic forces. The solution of these equations $\mathbf{q}(t)$ must also satisfy the constraint equations (e.g. Eq. (10)). These constraint equations lead in Eq. (11) to the presence of the reaction force $\boldsymbol{\Phi}_{\mathbf{q}}^{T}(\mathbf{q},t)\boldsymbol{\lambda}$, where $\boldsymbol{\lambda} \in \mathbb{R}^{m}$ is the Lagrange multiplier associated with the kinematic constraints.

The Constraint-Based Mesh approach uses bilateral constraints to enforce the equilibrium conditions across the boundaries of elements. This method contrasts the traditional finite element method, where elements share nodes. Consider a beam that is built up of two elements, such as in Figure 1. Although each ANCF beam element is defined by two end nodes, the traditional method would only have three global nodes because there is one shared node at the element connection. The Constraint-Based Mesh approach has a global total of four nodes, two of which are constrained in equilibrium. Despite requiring more data space for the same number of element connections, the lack of dependence on other elements makes the Constraint-Based Mesh approach ideally suited for parallel algorithms and hardware.

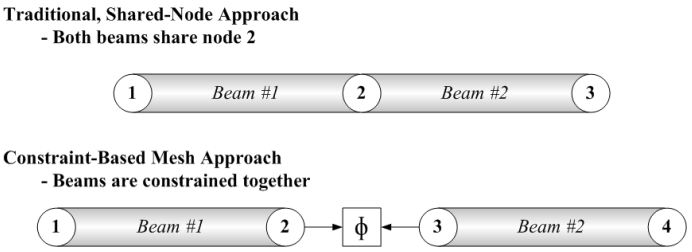

Figure 1. Beams are typically constrained by sharing nodes. A new approach uses kinematic constraints to enhance parallelism.

The Solution of Index-3 Differential Algebraic Equations (DAE)

Eq. (10) and Eq. (11) together form a system of index-3 DAE. Several low order numerical integration schemes have been effectively used to solve index-3 DAE [5]. Here we will consider the NEWMARK integration scheme. The NEWMARK method was originally used in the structural dynamics community for the numerical integration of a linear set of second order ODEs. In the NEWMARK formulation, the discretization of the multibody dynamics equation of motion yields

$$(\mathbf{M}\ddot{\mathbf{q}})_{n+1} + (\boldsymbol{\Phi}_{\mathbf{q}}^{T}\boldsymbol{\lambda})_{n+1} + (\mathbf{Q}_{\text{int}} - \mathbf{Q}_{\text{ext}})_{n+1} = 0 \tag{12}$$

Given the acceleration $\ddot{\mathbf{q}}_{n+1}$ at the new time step t_{n+1}, the new position and velocity are obtained as

$$\mathbf{q}_{n+1} = \mathbf{q}_n + h\dot{\mathbf{q}}_n + \frac{h^2}{2}\left[(1-2\beta)\ddot{\mathbf{q}}_n + 2\beta\ddot{\mathbf{q}}_{n+1}\right] \tag{13}$$

$$\dot{\mathbf{q}}_{n+1} = \dot{\mathbf{q}}_n + h\left[(1-\gamma)\ddot{\mathbf{q}}_n + \gamma\ddot{\mathbf{q}}_{n+1}\right] \tag{14}$$

where h is the integration step size and $\gamma \geq \frac{1}{2}$, $\beta \geq \frac{(\gamma + 1/2)^2}{4}$. The discretization of the constraint equation (with integration step size h) gives

$$\boldsymbol{\Phi}(\mathbf{q}_{n+1}, t_{n+1}) = 0 \tag{15}$$

It should be noted that in the NEWMARK method, $\mathbf{q}$ and $\dot{\mathbf{q}}$ are expressed as a function of $\ddot{\mathbf{q}}$ using the integration formulas given by Eq. (13) and Eq. (14).

A Newton's method can be used to solve the system of nonlinear equations defined by Eq. (12) and Eq. (15) for the set of unknowns $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$. The iterative algorithm of Newton's method requires at each iteration (k), the solution of the linear system

$$\begin{bmatrix} \hat{\mathbf{M}} & \boldsymbol{\Phi}_{\mathbf{q}}^T \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\ddot{\mathbf{q}} \\ \Delta\boldsymbol{\lambda} \end{bmatrix}^{(k)} = \begin{bmatrix} -\mathbf{e}_1 \\ -\mathbf{e}_2 \end{bmatrix}^{(k)} \tag{16}$$

Where $\mathbf{e}_i$ are the residuals in satisfying the set of the discretized equation of motion and constraint equations which are scaled such that

$$\begin{aligned} \mathbf{e}_1 &= \left(\mathbf{M}\ddot{\mathbf{q}}\right)_{n+1} + \left(\boldsymbol{\Phi}_{\mathbf{q}}^T\boldsymbol{\lambda}\right)_{n+1} + \left(\mathbf{Q}_{\text{int}}\right)_{n+1} - \left(\mathbf{Q}_{\text{ext}}\right)_{n+1} \\ \mathbf{e}_2 &= \frac{1}{\beta h^2}\boldsymbol{\Phi}(\mathbf{q}_{n+1}, t_{n+1}) \end{aligned} \tag{17}$$

This scaling is done in order to improve the conditioning of the Jacobian Matrix for Newton's method [5]. The matrix $\hat{\mathbf{M}}$ in Eq. (16) is defined as

$$\hat{\mathbf{M}} = \frac{\partial \mathbf{e}_1}{\partial \ddot{\mathbf{q}}} = \mathbf{M} + \beta h^2 \left[\left(\boldsymbol{\Phi}_{\mathbf{q}}^T\boldsymbol{\lambda}\right)_{\mathbf{q}} + \frac{\partial \mathbf{Q}_{\text{int}}}{\partial \mathbf{q}} - \frac{\partial \mathbf{Q}_{ext}}{\partial \mathbf{q}} \right] - h\gamma \left[\frac{\partial \mathbf{Q}_{ext}}{\partial \dot{\mathbf{q}}} \right] \tag{18}$$

Here $\frac{\partial \mathbf{Q}_{\text{int}}}{\partial \mathbf{q}}$ is the most compute-intensive term which represents the tangent stiffness matrix associated with nonlinear ANCF formulation.

The tangent stiffness matrix for the ANCF beam element $(\mathbf{K}_s \in \mathbb{R}^{12\times 12})$ is derived from the expression for the element elastic force as $\mathbf{K}_s = \frac{\partial \mathbf{Q}_s}{\partial \mathbf{e}}$ which becomes

$$\begin{aligned} \mathbf{K}_s = &\int_0^l EA(\varepsilon_{11}) \frac{\partial}{\partial \mathbf{e}} \left(\frac{\partial \varepsilon_{11}}{\partial \mathbf{e}} \right)^T dx + \int_0^l EA \left(\frac{\partial \varepsilon_{11}}{\partial \mathbf{e}} \right)^T \left(\frac{\partial \varepsilon_{11}}{\partial \mathbf{e}} \right) dx \\ &+ \int_0^l EI(\kappa) \frac{\partial}{\partial \mathbf{e}} \left(\frac{\partial \kappa}{\partial \mathbf{e}} \right)^T dx + \int_0^l EI \left(\frac{\partial \kappa}{\partial \mathbf{e}} \right)^T \left(\frac{\partial \kappa}{\partial \mathbf{e}} \right) dx \end{aligned} \tag{19}$$

Here each integral can be evaluated using Gauss-quadrature formula.

An Overview of the Graphical Processing Unit (GPU)

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 2, 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.

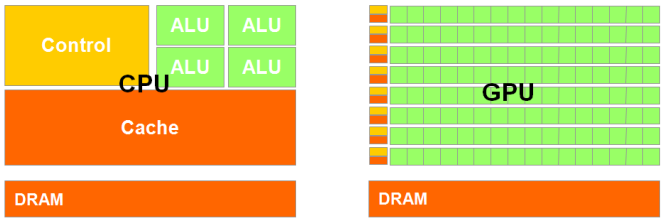

Figure 2. The GPU is specialized for compute-intensive, highly data parallel computation due to its graphics rendering origin. This is manifested in microprocessor designs that have a very large number of arithmetic logical units.

IMPLEMENTATION

The implementation of ANCF on parallel hardware can be divided into seven stages. The GPU allows for each of these stages to be performed with different execution configurations. For example, the internal force update is performed on an element-parallel level while the velocity update can be performed on a coordinate-parallel level. The seven stages are detailed below:

Sequential Preprocessing Stage

Allocate space for the position, velocity, acceleration vectors. Preprocess the collision object vector to indicate the position and elemental index of the object. Create external force vector and constant mass matrix. Transfer data to device.

Update collision objects and perform collision detection

The collision geometry of each beam is regarded as the union of a string of spheres that overlap each other. As explained in the theory portion of the paper, the global coordinate of any point on the beam can be determined by multiplying the shape function vector determined at a length *l* along the beam by the element nodal coordinates.

Update contacts

The contact force due to each collision is based on the collision normal and the contact depth. The contact force, represented in Cartesian coordinates, is then mapped onto the element that the sphere belongs to by multiplying the point force by the shape function associated with the element. Since multiple contacts can occur on the same element, the contacts must be computed individually and then reduced based on the elemental index. These contact forces are then added to the external force vector.

Internal force computation

Calculating the internal forces for the individual elements can be computed in parallel. The internal forces are based on the element's corresponding absolute nodal coordinates. This stage also handles the construction of the stiffness matrix.

Update acceleration vector

The acceleration for the entire system is solved using the NEWMARK method. The acceleration is iteratively computed by solving a linear system using the Bi-Conjugate Stable Krylov method that is included in the sparse matrix library, CUSP.

Position and velocity update computation

The position and velocity update computations are composed of multiplying a scalar value (the step size) by a vector (the corresponding velocity/acceleration vectors). Each thread can handle one entry of the global position/velocity vector. This step also handles the resetting of the global contact force vector.

Check for convergence

Convergence is considered complete when the norms of the right-hand side vector and the acceleration increment have reached a tolerance. If convergence criterion is not met, the algorithm branches back to the collision detection stage for another iteration.

NUMERICAL EXPERIMENTS AND RESULTS

Several numerical experiments are carried out in order to *a)* Validate the Constraint-Based Mesh approach against the traditional shared-node approach; *b)* Compare the GPU parallel implementation of ANCF to the CPU serial version; *c)* Validate the GPU implementation against other nonlinear finite element tools.

Validation of the Constraint-Based Mesh Approach

The Constraint-Based Mesh approach uses bilateral constraints to enforce the equilibrium conditions across the boundaries of elements. The resulting set of differential algebraic equations that describes this system is solved using the NEWMARK implicit time stepping algorithm. In this numerical experiment, the generalized 3D motion of two ANCF beams is studied: one using the traditional, shared-node approach and one using the Constraint-Based Mesh Approach. The beams, shown in Figure 3, have a circular cross section of radius equal to 0.01 [m] and are comprised of eight elements that are each 1 [m] in length. The beams have a density of 7200 [kg/m^3], a modulus of elasticity of 2.0e7 [Pa], and are under the effect of gravity. The beams are pinned at one end and initially held horizontally with respect to the gravitational field. Upon starting the simulation, the beams are released. The position of the beam tips over the course of 2 seconds can be seen in Figure 4.

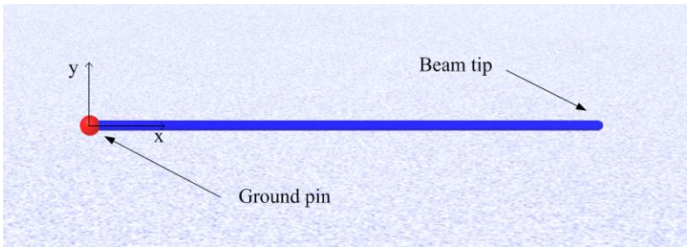

Figure 3: A flexible beam pendulum with a ground pin constraint.

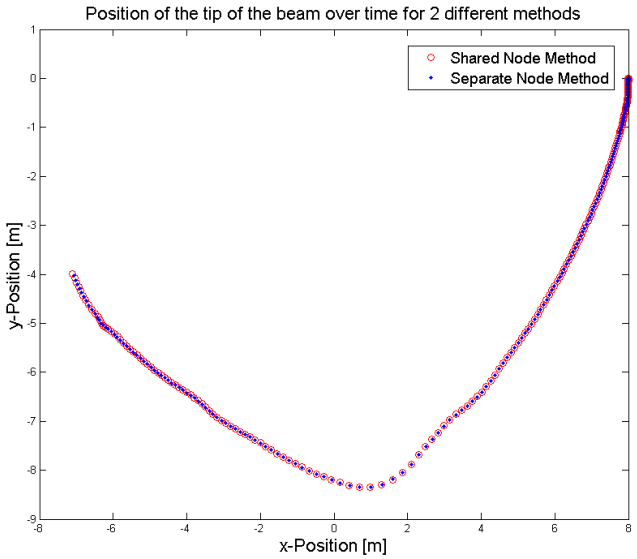

Figure 4. Position of the beam tip over time. Two beams were simulated: 1) using the traditional, shared-node approach and 2) using the Constraint-Based Mesh technique.

It can be seen that the two different solution techniques match well with each other. These results show that the fixed constraint can be used to mimic the sharing of nodes.

Several measures of energy were observed for this system. The strain, potential, kinetic, and total energy were plotted in Figure **5**.

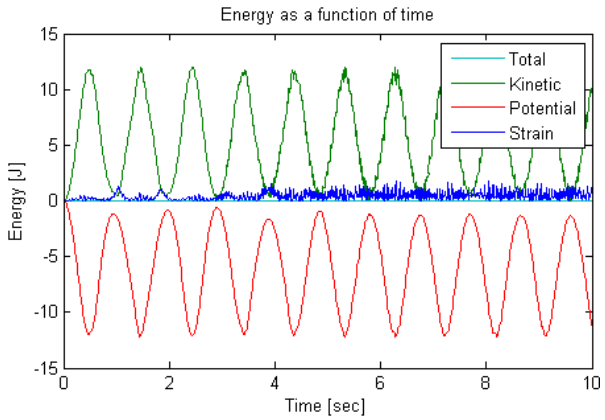

Figure 5: Energy as a function of time for a flexible beam pendulum.

Comparison of the GPU Implementation of ANCF to the CPU Serial Version

Being computationally intensive, the ANCF methodology stands to benefit from the use of parallel computation. The scaling analysis performed in Figure 6 shows the amount of time taken to simulate the dynamics of beams that are not in contact. The scaling analysis demonstrates that the amount of time devoted to solving large mechanical systems of flexible bodies using a sequential approach is intolerable.

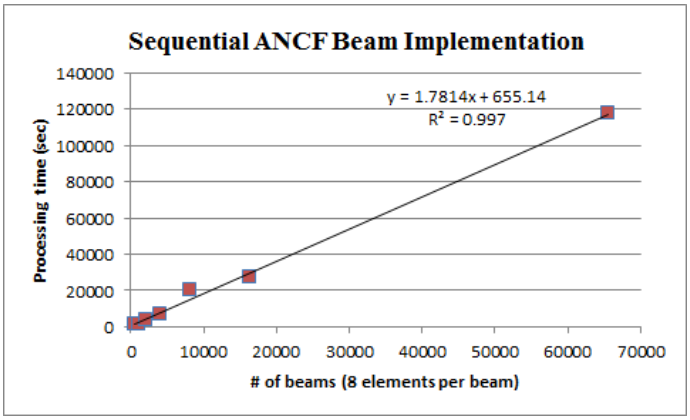

Figure 6. Scaling analysis for a serial implementation of ANCF beams.

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 the element level.

In an effort to compare the CPU and GPU implementations, a mechanical system containing hundreds of thousands of flexible beams pinned at one end is used. Each beam had a circular cross section of radius equal to 0.01 [m] and was 3 [m] in length. The beams had a density of 7200 [kg/m^3], a modulus of elasticity of 2.0e7 [Pa], and were under the effect of gravity. An integration step size of 1.0e-5 [s] was used. 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 C2070 graphics card for varying numbers of beams. On average, a 250x speedup was observed from the GPU implementation over the CPU implementation, shown in Figure 7.

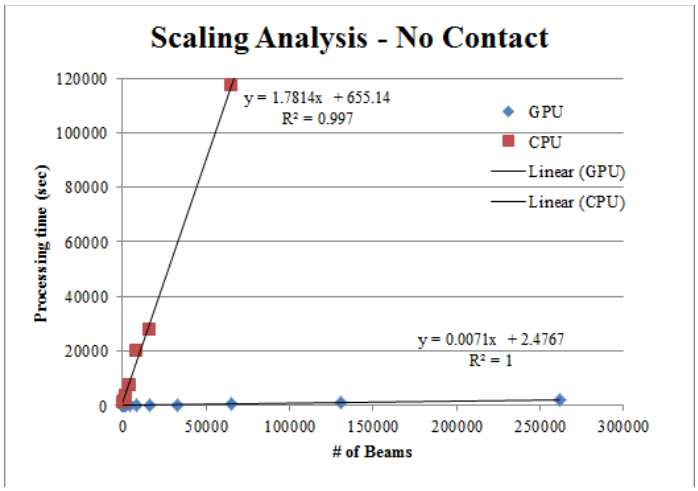

Figure 7. Processing time for the CPU and GPU implementations for varying numbers of beams. Note that the GPU implementation is much faster than the CPU implementation.

Validation of the GPU Implementation against ABAQUS and FEAP

In this set of numerical experiments, a generalized 3D motion of the ANCF beam with no contact is studied. A beam is used to study the 3D motion of a highly deformable pendulum (beam pinned at one end, parameters as in table below) under the effect of gravity and externally applied force. The pendulum is modeled using 10 ANCF beam elements, 100 BEAM33 elements in ABAQUS [6], and 100 3D continuum elements in FEAP [7]. To generate a 3D motion, a constant force of 1N in the Z direction is applied at the end node for two seconds. Figure 8, Figure 9, and Figure 10 show the displacements of the pendulum-tip. The ANCF results match well the ABAQUS and FEAP results. The gradient deficient ANCF elements do not suffer from shear locking problems. These results show that a generalized 3D motion with no torsion along the beam axis can be modeled correctly using the ANCF approach.

Parameters	Value
Contact type	No Contact
Length (m)	1
Cross-section Area(m^2)	π x 0.01^2
Material Density (kg/ m^3)	7200
Modulus of elasticity (Pa)	2.0E7
Second moment of area (m^4)	7.85E-10
Tip Mass (kg)	--
External Force	Gravity (negative y-direction)
Integration Step-Size (s)	1.0E-4

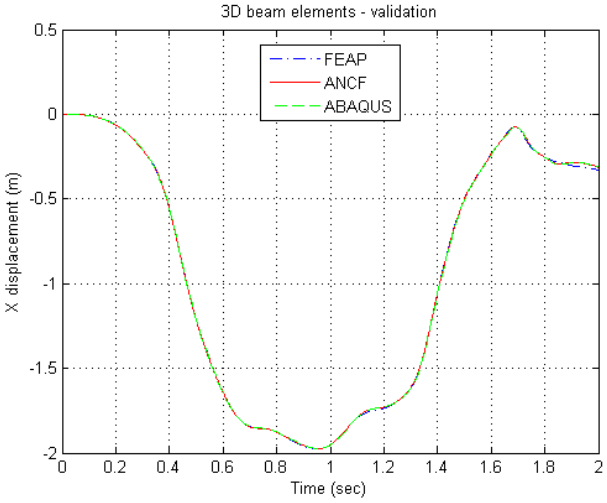

Figure 8. X-displacement of a pendulum-tip (ANCF, ABAQUS, and FEAP comparison).

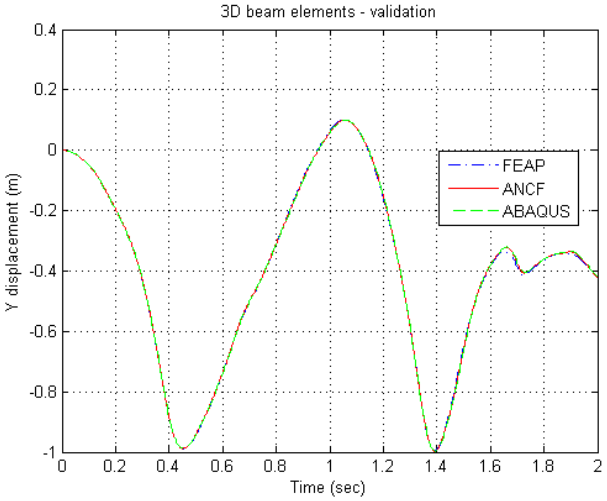

Figure 9. Y-displacement of a pendulum-tip (ANCF, ABAQUS, and FEAP comparison).

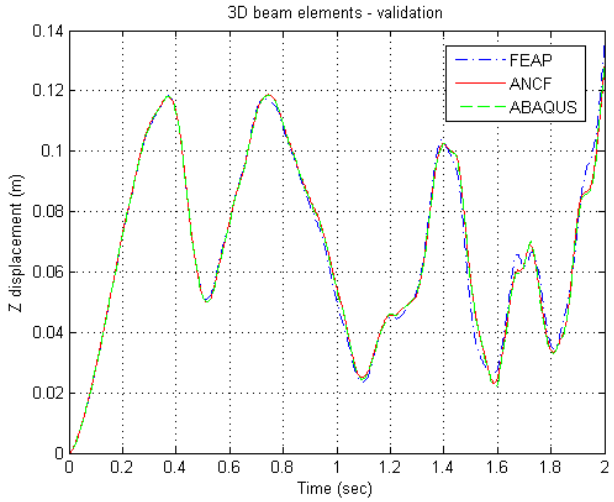

Figure 10. Z-displacement of a pendulum-tip (ANCF, ABAQUS, and FEAP comparison).

CONCLUSIONS

This paper presents a methodology for combining the implicit NEWMARK time integration algorithm, fine-grained parallel processing on the Graphics Processing Unit, and the Constraint-Based Mesh approach to solve complex flexible multibody systems. The gradient deficient ANCF beam elements used in the dynamics analysis exhibit good convergence characteristics and do not suffer from shear locking problems.

The Constraint-Based Mesh approach considers each element in the system as a separate body. Elements are connected together by kinematic constraints. Along with allowing complex structures to be created, the CBM approach lends itself to parallelization on the graphics processing unit. Each body can be operated on separately while updating the internal forces or constructing the stiffness matrices.

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.

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. Khude, N., L.O. Jay, A. Schaffer, and D. Negrut. *A Discussion of Low Order Numerical integration Formulas for Rigid and Flexible Multibody Dynamics (DETC2007-35666)*. in *6th ASME International Conference on Multibody Systems, Nonlinear Dynamics and Control*. 2007. Las Vegas, NV: ASME.
6. Systems, D. *Simulia*. 2012; Available from: http://www.3ds.com/products/simulia/overview/.
7. Taylor, R.L., *FEAP, a Finite Element Analysis Program: Version 7.1 j User Manual*. 1999: Dept. of Civil and Environmental Engineering, University of California.

