Integration of Large Deformation Finite Element and Multibody System Algorithms

Ahmed A. Shabana
Department of Mechanical Engineering,
University of Illinois at Chicago,
842 West Taylor Street,
Chicago, IL 60607-7022

Olivier A. Bauchau
Daniel Guggenheim School of Aerospace Engineering,
Georgia Institute of Technology,
270 Ferst Street,
Atlanta, GA 30332-0150

Gregory M. Hulbert
Department of Mechanical Engineering,
2350 Hayward,
University of Michigan,
Ann Arbor, MI 48109-2125

This paper presents an overview of research and development efforts that are currently being devoted to integrate large deformation finite element formulations with flexible multibody system algorithms. The goal is to develop computer simulation capabilities for the analysis of physics and engineering models with significant details. The successful development of such new and integrated algorithms will also allow modeling and simulation of systems that cannot be solved using existing computer algorithms and codes. One of the main difficulties encountered in this integration process is attributed to the fact that the solution procedures used in finite element codes differ significantly from those used in general-purpose flexible multibody system codes. Finite element methods employ the corotational formulations that are often used with incremental solution procedures. Flexible multibody computer codes, on the other hand, do not, in general, use incremental solution procedures. Three approaches are currently being explored by academic institutions and the software industry. In the first approach, gluing algorithms that aim at performing successful simulations by establishing an interface between existing codes are used. Using different coordinates and synchronizing the time stepping are among several challenging problems that are encountered when gluing algorithms are used. In the second approach, multibody system capabilities are implemented in existing finite element algorithms that are based on large rotation vector formulations. For the most part, corotational formulations and incremental solution procedures are used in this case. In the third approach, a new large deformation finite element formulation that can be successfully implemented in flexible multibody system computer algorithms that employ nonincremental solution procedures is introduced. The approach that is now being developed in several institutions is based on the finite element absolute nodal coordinate formulation. Such a formulation can be systematically implemented in general-purpose flexible multibody system computer algorithms. Nonlinear constraint equations that describe mechanical joints between different bodies can be formulated in terms of the absolute coordinates in a straightforward manner. The coupling between the motion of rigid, flexible, and very flexible bodies can also be accurately described. The successful integration of large deformation finite element and multibody system algorithms will lead to a new generation of computer codes that can be systematical and efficiently used in the analysis of many engineering applications. [DOI: 10.1115/1.2756075]

Keywords: multibody systems, large deformation finite element formulations, computational mechanics

1 Introduction

The well recognized limitations of existing multibody system formulations and computer codes explain the motivation for the work reported in this study. The goal of using multibody system techniques is to be able to systematically develop and efficiently solve the nonlinear differential and algebraic equations (DAEs) of complex and large-scale physics and engineering systems. Over the past three decades, general-purpose multibody system computer programs have drastically changed the way many products are designed and analyzed in automotive, aerospace, railroad, machine, and earth moving equipment industries, among many others. The use of general multibody system computer codes is now essential in the design process in these industries, and these codes have become an increasingly indispensable tool for many companies in order to have leading edge capabilities. Detailed computer models can be first developed and tested before a physical prototype is built, saving time and resources and contributing to making the products safer, better, and more cost effective. Multibody computer codes and algorithms, however, have been pushed to their limits, and there is a large class of new challenging problems that cannot be solved using existing tools. The solution of these new multiphysics and multiscale problems requires the development of a new generation of computer codes that integrate large deformation finite element and multibody system algorithms.

Most existing general-purpose multibody system computer codes are designed to solve systematically and efficiently rigid body systems and small deformation problems only. These codes, however, have served the industry well for more than a quarter of a century. The small deformation multibody system formulations correctly account for the nonlinear dynamic coupling between different modes of displacements that can vary at significantly different rates. This is one of the central issues in developing general-purpose computer algorithms that are based on the *floating frame of reference formulation*. Nonetheless, this formulation, the most widely used method in flexible multibody system dynamics for solving small deformation problems, is not suited for solving large deformation problems, and, as a result, it cannot be used in the analysis of many physics and engineering applications.

Contributed by the Design Engineering Division of ASME for publication in the JOURNAL OF COMPUTATIONAL AND NONLINEAR DYNAMICS. Manuscript received October 14, 2006; final manuscript received January 19, 2007. Review conducted by Subhash C. Sinha.

In contrast, existing large deformation finite element algorithms and computer codes are not designed for modeling multibody systems that represent many biological, industrial, and technological applications. For this reason, it becomes necessary to successfully integrate large deformation finite element formulations and multibody system algorithms to be able to develop a new generation of computer algorithms and codes that can be used to solve the new challenging problems being encountered. Examples of these important and challenging applications include rollover crashworthiness, very flexible cables and belt drives, pantograph/catenary systems of high speed trains, biological systems, very flexible and plastic manufacturing systems, leaf springs and tires, media transport systems, and aerospace systems such as helicopter rotor blades or flexible space antennae.

Multibody systems are characterized by two distinguishing features. First, system components undergo finite relative rotations. Second, these components are connected by mechanical joints that impose restrictions on their motion. The finite rotations introduce geometric nonlinearities; that is, multibody systems are inherently nonlinear. Mechanical joints introduce algebraic constraint equations leading to a mathematical model that consists of DAEs. Consequently, solution procedures that differ from those used in the finite element literature must be used. The differences between the finite element and multibody system solution procedures are significant and pose challenging problems when the two types of algorithms are integrated. For instance, most existing finite element algorithms are based on a corotational formulation (Belytschko et al. [1]). The use of the corotational formulations with incremental solution procedures becomes necessary when conventional nonisoparametric beam, plate, and shell finite elements that employ infinitesimal rotations as nodal coordinates are used. These conventional finite elements (Zienkiewicz [2]; Zienkiewicz and Taylor [3]), as discussed in the literature, cannot be used to correctly describe an arbitrary finite rigid body rotation. The computer implementation of the multibody system formulations, on the other hand, is based for the most part on nonincremental solution procedures. Furthermore, multibody system algorithms are designed for systems of DAEs that must be solved simultaneously and employ a different set of generalized coordinates that correctly describe arbitrary rigid body displacements.

Three approaches are currently being examined for the integration of large deformation finite element and multibody system algorithms. These are mainly the *gluing algorithms* (GAs) (cosimulation), the *finite element based direct integration method* (FEBDI), and the *multibody system based direct integration method* (MSBDI). In the GA, one attempts to combine existing finite element algorithms with existing nonincremental multibody system algorithms (Tseng and Hulbert [4]; Wang et al. [5]). In this case, the basic features of the existing algorithms, including the type of coordinates and solution procedure used, are preserved. The use of the GAs introduces several challenging problems, which are the subject of research, due to the fundamental differences between the two types of algorithms. For example, the frequency contents in large deformation variables and forces can be significantly different from those in rigid body or small deformation variables and forces. Synchronizing the time stepping, communication, and exchange of information between two different computer codes introduces several challenging problems.

Another alternative, which has been proven by preliminary results to be promising, is to use the method of direct integration. Two different direct integration methods can be used. The first is the FEBDI, in which the required multibody system capabilities are implemented in existing finite element large rotation vector formulations (Bauchau [6]; Géradin and Cardona [7]). The algorithm in this case can be based on the corotational formulations. In the second method, MSBDI—a new large deformation finite element formulation is used, together with a nonincremental solution procedure. Such a finite element formulation must correctly describe rigid body motions, and a systematic integration with multibody algorithms must be possible. The large deformation finite element *absolute nodal coordinate formulation* satisfies these requirements. Before discussing the methods of integrations, the classical finite element and multibody system formulations and algorithms are briefly reviewed in the following two sections.

2 Classical Finite Element Formulations and Algorithms

Classical finite element formulations are concerned with modeling the inertia and stiffness of structural systems, particularly the nonlinear elastic behavior. The problem of large relative rigid body displacements, which is a typical multibody system problem, was not addressed in developing these classical formulations. Furthermore, conventional beam, plate, and shell finite elements, which are widely used in the analysis of structural components, employ infinitesimal rotations as nodal coordinates (Zienkiewicz [2]; Zienkiewicz and Taylor [3]). These elements cannot be used to describe correctly a finite rigid body rotation. The linearization of the kinematic equations used in these elements has been investigated in the literature (Shabana [8]), and it is shown that a finite rigid body rotation of the element leads to a nonzero strain. To circumvent this problem, corotational formulations and incremental solution procedures have been used in the classical finite element literature. In general, the following matrix equation is used in finite element algorithms:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{Q} \tag{1}$$

In this equation, $\mathbf{M}$ is the system mass matrix, $\mathbf{K}$ the system stiffness matrix, $\mathbf{Q}$ the vector of nodal forces, and $\mathbf{q}$ the vector of nodal coordinates. For a nonlinear analysis, system mass and stiffness matrices are, in general, nonlinear, in which case these matrices are iteratively updated. Kinematic constraints on the coordinates are imposed, for the most part, using a penalty method or the augmented Lagrangian formulation (Bayo et al. [9]; Bayo and Ledesma [10]) to avoid dealing with DAEs. This is one of the differences between the algorithms used in the finite element analysis and those used in multibody system dynamics, where the kinematic constraints are imposed using a system of nonlinear algebraic equations enforced by the Lagrange multiplier technique. It is also important to mention that penalty methods can introduce a significant stiffness into the system dynamic equations, which can lead to numerical problems.

To overcome the limitations associated with conventional beam, plate, and shell elements, corotational formulations with finite rotations were developed (Simo and Vu-Quoc [11]) and can be cast in the form of Eq. (1). In these formulations, finite rotation nodal coordinates are used to correctly describe arbitrarily large rigid body rotations. Displacement coordinates are defined in an inertial frame, and absolute displacements, velocities, and accelerations are used to formulate the nonlinear finite element dynamic equations. Independent interpolations are used for the displacement and rotation fields.

While the introduction of finite rotation variables within the formulation of finite elements is a major step toward the modeling of multibody systems, kinematic constraints must also be accommodated to deal with actual systems. As previously pointed out, penalty formulations are readily included in finite element models (FEMs); however, this approach never exactly enforces the constraints. As previously mentioned, this contrasts with multibody formulations where kinematic constraints are typically enforced via the Lagrange multiplier technique. Although not a feature found in all finite element codes, a Lagrange multiplier can be included in these formulations using the augmented Lagrangian formulation proposed by Bayo et al. [9] and Bayo and Ledesma [10], with little computational effort. With this approach, kinematic constraints are enforced to machine accuracy. Note that the generalized coordinate partitioning method (Wehage [12] and Wehage and Haug [13]), the basis for the development of many

general-purpose multibody system codes, is unlikely to be implemented in finite element codes since it destroys the bandedness of the system matrix if the dependent coordinates are eliminated at the acceleration level. In the actual multibody system implementation, the augmented form of the dynamic equations of motion is used to preserve the sparse matrix structure. Sparse banded matrices are used systematically in finite element formulations for computational efficiency.

Because of the difficulties and problems that arise from directly importing existing finite element formulations and algorithms and using them in the analysis of multibody system applications, multibody system research has focused on the development of techniques that accurately and efficiently model the deformation of flexible bodies. Accuracy is achieved by using fully nonlinear formulations that define a unique displacement field, while efficiency is achieved by using only the necessary deformation modes and optimum sparse matrix structure, as will be discussed in the following section.

3 Classical Multibody System Formulations and Algorithms

Multibody systems are characterized by two distinguishing features: system components undergo finite relative rotations, and these components are connected by mechanical joints that impose restrictions on their motion. Systems without constraints or finite relative rotations do not define a multibody system problem. Multibody system algorithms have been designed to accurately describe finite reference rotations and robustly handle nonlinear algebraic constraint equations. Algebraic constraint equations introduce many fundamental and computational problems that include the choice of coordinates, the identification of the system degrees of freedom, and the numerical solution of the system of DAEs, among many others.

The main approach that has been used to study the deformation in multibody systems is called the floating frame of reference formulation. While this formulation was introduced for flexible bodies more than a century ago, and hence precedes the corotational frame formulation, the finite element floating frame of reference formulation was introduced in the early 1980s and led to a new field currently known as *flexible multibody system dynamics*. Crucial to the development of the finite element floating frame of reference formulation is the concept of the *intermediate finite element coordinate system* (Shabana [14]). This concept allows the systematic development of the large displacement nonlinear dynamic equations of motion of flexible bodies that have a complex geometry and also allows for a systematic elimination of insignificant deformation modes (Agrawal and Shabana [15]), leading to a reduced order mathematical model that can be solved efficiently using numerical integration methods.

In the floating frame of reference formulation, two sets of coordinates are used to define the configuration of the flexible body. The first set is a set of absolute coordinates that define the location and orientation of the floating frame of reference, while the second is a set of elastic coordinates that define the deformation of the body with respect to its reference. These coordinates are different from those used in the corotational finite element formulation since the gross reference motion is not described using the elastic nodal coordinates. For this reason, the floating frame of reference formulation leads to zero strain under an arbitrary rigid body motion and also leads to an exact modeling of the rigid body dynamics when conventional beam, plate, and shell elements are used. Since the deformation is defined with respect to the body coordinate system, one can, in the case of small deformations, systematically eliminate high frequency modes of deformation. The resulting reduced order model can be solved more efficiently and, at the same time, an exact modeling of rigid body dynamics is achieved. Because a coupled set of reference and elastic coordinates is used, the floating frame of reference formulation leads to a highly nonlinear mass matrix. Therefore, the centrifugal and Coriolis forces are not equal to zero. It can be shown that the dynamic equations that govern the motion of a flexible body obtained using the floating frame of reference formulation can be written as follows:

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q}_s + \mathbf{Q}_e + \mathbf{Q}_v \tag{2}$$

In this equation, $\mathbf{M}$ is the system mass matrix, $\mathbf{q}$ the vector of the system generalized coordinates, $\mathbf{Q}_s$ the vector of elastic forces, $\mathbf{Q}_e$ the vector of applied forces, and $\mathbf{Q}_v$ the vector of centrifugal and Coriolis inertia forces. Using the coordinate partitioning

$$\mathbf{q} = [\mathbf{q}_r^T \ \mathbf{q}_f^T]^T \tag{3}$$

Eq. (2) can be written as

$$\begin{bmatrix} \mathbf{M}_{rr} & \mathbf{M}_{rf} \\ \mathbf{M}_{fr} & \mathbf{M}_{ff} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_r \\ \ddot{\mathbf{q}}_f \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ (\mathbf{Q}_s)_f \end{bmatrix} + \begin{bmatrix} (\mathbf{Q}_e)_r \\ (\mathbf{Q}_e)_f \end{bmatrix} + \begin{bmatrix} (\mathbf{Q}_v)_r \\ (\mathbf{Q}_v)_f \end{bmatrix} \tag{4}$$

In this equation, subscripts r and f refer, respectively, to reference and elastic coordinates. The details of the matrices and vectors that appear in the preceding equation can be found in the literature (Shabana [14]). The dynamic coupling between the reference motion and the elastic deformation is represented by the two matrices $\mathbf{M}_{rf}$ and $\mathbf{M}_{fr}$. The study of this dynamic coupling is one of the interesting problems that has been the subject of several investigations in the multibody system dynamics literature. The dependence of the expressions of the matrices $\mathbf{M}_{rf}$ and $\mathbf{M}_{fr}$ on the choice of the body coordinate system has also been investigated. It is also important to note that the submatrix $\mathbf{M}_{ff}$, which is the mass matrix that appears in linear finite element formulations, is a constant symmetric matrix. It is clear from the structure of Eq. (4) that the equations of motion that govern the dynamics of *rigid bodies* (no deformation) and the equations of motion that govern the dynamics of *structural systems* (no reference motion) can be obtained as special cases from Eq. (4).

Multibody systems are subjected to kinematic constraints resulting from mechanical joints and specified motion trajectories. The resulting constraint equations can be written in the following vector form:

$$\mathbf{C}(\mathbf{q}_r, \mathbf{q}_f, t) = \mathbf{0} \tag{5}$$

where $\mathbf{C}$ is the vector of constraint functions and t is time. This vector of constraint equations can be differentiated twice with respect to time to define the constraint functions at the acceleration level. This leads to the following acceleration equations:

$$\mathbf{C}_{\mathbf{q}_r}\ddot{\mathbf{q}}_r + \mathbf{C}_{\mathbf{q}_f}\ddot{\mathbf{q}}_f = \mathbf{Q}_c \tag{6}$$

In this equation, $\mathbf{C}_{\mathbf{q}_r}$ and $\mathbf{C}_{\mathbf{q}_f}$ are the constraint Jacobian matrices associated, respectively, with the reference and elastic coordinates, and $\mathbf{Q}_c$ is the quadratic velocity vector that arises as the result of differentiating the constraint equations twice with respect to time. In the case of constrained motion, Eq. (4) must be modified and written as

$$\begin{bmatrix} \mathbf{M}_{rr} & \mathbf{M}_{rf} \\ \mathbf{M}_{fr} & \mathbf{M}_{ff} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_r \\ \ddot{\mathbf{q}}_f \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{\mathbf{q}_r}^T \\ \mathbf{C}_{\mathbf{q}_f}^T \end{bmatrix} \boldsymbol{\lambda} = \begin{bmatrix} \mathbf{0} \\ (\mathbf{Q}_s)_f \end{bmatrix} + \begin{bmatrix} (\mathbf{Q}_e)_r \\ (\mathbf{Q}_e)_f \end{bmatrix} + \begin{bmatrix} (\mathbf{Q}_v)_r \\ (\mathbf{Q}_v)_f \end{bmatrix} \tag{7}$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. The preceding two equations can be combined in one matrix equation that can be written as

$$\begin{bmatrix} \mathbf{M}_{rr} & \mathbf{M}_{rf} & \mathbf{C}_{\mathbf{q}_r}^T \\ \mathbf{M}_{fr} & \mathbf{M}_{ff} & \mathbf{C}_{\mathbf{q}_f} \\ \mathbf{C}_{\mathbf{q}_r} & \mathbf{C}_{\mathbf{q}_f} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_r \\ \ddot{\mathbf{q}}_f \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} (\mathbf{Q}_e)_r + (\mathbf{Q}_v)_r \\ (\mathbf{Q}_s)_f + (\mathbf{Q}_e)_f + (\mathbf{Q}_v)_f \\ \mathbf{Q}_c \end{bmatrix} \tag{8}$$

This system of algebraic equations can be solved for the accelerations and the vector of Lagrange multipliers. The coefficient matrix in this equation is sparse, and most general-purpose multibody computer programs use sparse matrix techniques for the efficient

solution of the preceding equations in the case of large and complex systems.

One method that multibody system algorithms employ for the numerical solution given a set of initial conditions is the generalized coordinate partitioning technique proposed by Wehage [12]. In this technique, independent accelerations are identified and integrated forward in time to determine independent coordinates and velocities. The dependent coordinates and velocities are obtained by solving the constraint equations at the position and velocity levels, respectively. Multibody system algorithms, therefore, ensure that the constraint equations are satisfied at the position, velocity, and acceleration levels.

The standard solution procedure used in the small deformation analysis of multibody systems consists of two steps. In the first step, constant vectors and matrices that appear in Eq. (8) are determined using a finite element preprocessor computer program. Several finite element commercial codes can generate these standard data. In the second step, the differential equations of motion and the algebraic constraint equations are solved using numerical methods. This systematic small deformation solution procedure is adopted in some cases to obtain an approximate solution of large deformation problems because of the lack of large deformation modeling capabilities in existing commercial multibody system codes. In present industry practices, multibody tools and large deformation finite element codes are used "side by side" but are not really integrated. For instance, in a preprocessing step, the linear stiffness characteristics of a complex, three-dimensional component of the system are determined using a detailed FEM; in the multibody model, the same component can be represented by discrete springs with the identified stiffness characteristics. Detailed FEMs of complex structures are also used to determine their vibration modes, which then form the basis for a subsequent multibody analysis. Another commonly used technique is the postprocessing of multibody simulation results using detailed FEMs. The dynamic loads obtained from a multibody simulation are applied to detailed FEMs of complex components to predict three-dimensional stress field and fatigue or crack growth behavior. While this side-by-side or "feed-forward" approach is able to answer important questions about the response of multibody systems, it does not represent a real integration of the two disciplines; it is limited in its predictive capabilities and unable to address the challenges previously mentioned in this paper.

It is clear from the discussion presented in this and preceding sections that there are fundamental differences between the finite element and multibody system approaches, formulations, and algorithms. Nonetheless, the integration of finite element and multibody system algorithms is necessary to be able to study complex models that include significant details. The successful integration of these two types of algorithms will lead to a new generation of computer simulation environment that will significantly impact design, analysis, and performance evaluation procedures. Three different methods are currently being examined in an attempt to integrate these two different types of algorithms. These are the GA, the FEBDI based on geometrically exact formulation, and the MSBDI based on the absolute nodal coordinate approach. The last two methods, FEBDI and MSBDI, have common features since they are based on direct integration instead of cosimulation. The three methods are briefly discussed in the following sections.

4 Gluing Algorithm

One approach to code integration is to undertake cosimulation, in which the finite element program and the multibody dynamics program run concurrently, but separately. The need exists to exchange appropriate information, for example, kinematic and force data, between the two codes. In addition, the execution of the codes must be coordinated so that their combined solutions are identical or comparable to (with a measurable error) solutions that would be obtained from "all-at-once" approaches. The term "gluing algorithm," has been adopted to classify algorithms that can perform these simulation needs (Tseng and Hulbert [4]). The foundation of the gluing approach arises from the already disparate simulation models and programs in use within the industry. Rather than attempting to partition an existing model, the goal is to glue together an already partitioned model that may contain many different finite element and multibody dynamics models. For example, a detailed vehicle model that includes tires, leaf springs, suspension and steering components and joints, chassis frame, etc., can be developed. The large deformation finite element algorithms can be used to model the tires and leaf springs as well as the chassis in the case of crash simulation, while multibody system algorithms are used to model the suspension and steering components and mechanical joints as well as other rigid bodies or bodies that experience small deformations.

Researchers have studied the decomposition of large mechanical systems with a primary focus on the decomposition strategy. Many groups have explored extending the concept of parallel computing to distributed simulation. The parallelization of both direct solution methods (Farhat and Wilson [16]) and iterative methods (Farhat and Roux [17]) has been studied extensively. A substructuring method, FETI (finite element tearing and interconnecting) (Farhat and Roux [18]), introduces extra traction variables and exhibits more flexibility for model reduction and coupling, compared to existing competing schemes. A similar algorithm, IGI, was developed (Modak and Sotelino [19]), which provides a practical way to update the interface force and in which the subsystem models can be solved independently.

In the multibody dynamics arena, researchers have also studied how to partition and parallelize systems. One strategy adopted by researchers is similar to substructuring in FEM, that is, a small global problem is formed by incorporating condensed subsystem models. This smaller problem then is solved to provide necessary information to subsystem models. The subsystem models can be subsequently solved based on this information. Kim [20] proposed a subsystem synthesis method for a dynamic analysis of vehicle multibody systems, in which each subsystem is independently analyzed with a virtual reference body and the overall vehicle system analysis is formed by synthesizing the effective inertia matrix and force vector from the virtual reference body of each subsystem. Another approach was developed by Anderson and Duan [21], in which the equations of the subsystem models are evaluated in parallel, and the results are loaded into a single systemwide equation to explicitly calculate the constraint forces.

Treating the subsystem models as control blocks and taking advantage of many sophisticated control-based simulation software packages is another common modeling approach. Kübler and Schiehlen [22] proposed a modular formulation for multibody systems based on the block representation of a multibody system with corresponding input and output quantities. This "block diagram" representation of the system can then be embedded into appropriate simulation packages such as SIMULINK.

The methods reviewed above either involve the active decomposition of the full system and require more information than just that associated with the subsystem interfaces or mandate specific requirements or structures on the formulation of the subsystems. In the context of coupling already distributed subsystems, the gluing perspective is preferred. A well designed GA should only rely on the information at the interfaces of the models that are to be coupled. Here, interface refers to the connections or common surfaces of two models. An interface can be represented by a set of interface nodes in a FEM or by a set of connecting joints in a multibody dynamics model. The typical information available at the interface can be classified, in the present context, as kinematic information and force information. The kinematic information may contain displacements, velocities, and/or accelerations of the interface. Force information refers to action-reaction forces at the interface.

Mechanics principles require that at any interface the force quantities, namely, action-reaction forces, satisfy the equilibrium

equations and the kinematic quantities satisfy the compatibility conditions, where it is assumed that the equilibrium and compatibility conditions in the internal domain of each subsystem are satisfied a priori. The GA employs an iterative process, starting with an initial guess of some of the interface quantities. These interface quantities are then updated using a prescribed iteration process to satisfy the equilibrium and/or compatibility conditions at the interface.

In general, if a proper set of interface force variables is defined such that the equilibrium conditions are satisfied, then only the compatibility conditions need to be considered during the iteration process. In this case, the interface force variables can be considered as functions of the interface kinematic quantities, and these interface force variables can be updated using the kinematic information and compatibility conditions. Similarly, if a proper set of the interface kinematic variables is defined such that the compatibility conditions are satisfied, then only the equilibrium conditions need to be considered during the iteration process. In this latter case, the interface kinematic variables are functions of the force quantities at the interface, and they can be updated by satisfying the equilibrium conditions. Different GAs ensue, depending on which group of interface quantities is considered as the defined input. Detailed descriptions of different GAs are given in (Wang et al. [5]).

In general, GAs can be classified into three groups: *T-T*, *X-T*, and *X-X*, where *X* denotes kinematic quantities and *T* denotes force quantities. In the *T-T* algorithm, each system provides kinematic quantities to the gluing coordination module. The coordinator then returns updated force quantities to the distributed systems; the systems update their dynamic states using these new force estimates. This process is iterated until convergence is reached. This algorithm provides the least intrusive change to existing finite element and multibody dynamics codes as the *T-T* algorithm has the structure of a general force element for most codes. The *X-X* GA is the inverse of the *T-T* approach, whereby force quantities are provided to the gluing coordination module from each system. The coordinator then provides updated kinematic states to the systems. This approach may be viewed as a time-dependent prescribed kinematic boundary condition to be specified for each system. The MEPI algorithm (Maggi's equations with perturbed iteration) (Tseng et al. [23]) is an example of an *X-X* GA. The mixed *X-T* GA is adopted by SIMULINK and describes the so-called "across and through" variable method adopted in the 20-SIM program, which is employed in the control block strategy of Kübler and Schiehlen [22]. In the *X-T* algorithm, there is a clear directional flow of kinematic and force information through systems.

To highlight the GA concept, attention is restricted to the *T-T* gluing approach. Assume that $\mathbf{F}$ is a properly defined interface force vector; that is, $\mathbf{F}$ contains the necessary and sufficient set of variables that can represent the force space at the interfaces considered and $\mathbf{F}$ is self-balanced; that is, the equilibrium conditions at the interfaces are automatically satisfied if $\mathbf{F}$ is employed. Let $\mathbf{e}$ be an error measure vector that represents the violation of the compatibility conditions at the interfaces, where $\mathbf{e}=0$ indicates that the compatibility conditions are fully satisfied. In the general case, $\mathbf{e}$ can be considered as a function of $\mathbf{F}$, namely,

$$\mathbf{e} = \mathbf{e}(\mathbf{F}) \tag{9}$$

Since $\mathbf{F}$ is defined in such way that the equilibrium conditions can be automatically satisfied, the objective of the GA is to bring $\mathbf{e}$ to zero, that is, find $\mathbf{F}$ such that

$$\mathbf{e} = 0 \tag{10}$$

This equation defines a set of linear or nonlinear equations, which can be solved by a properly chosen algorithm of linear or nonlinear equation solvers.

Assuming an initial guess $\mathbf{F}=\mathbf{F}^{(i)}$, $(i=0)$, we have $\mathbf{e}^{(i)}=\mathbf{e}(\mathbf{F}^{(i)})$. Then, in the general case, a GA (*T-T* method) is proposed as

$$\mathbf{F}^{(i+1)} = \mathbf{F}^{(i)} + \mathbf{\Lambda}(-\mathbf{e}^{(i)}) \tag{11}$$

where $\mathbf{\Lambda}$ is called the *gluing matrix* or *lambda matrix*, which will be constant if Eq. (10) is linear or a function of $\mathbf{F}$ if Eq. (10) is nonlinear. The gluing matrix can be obtained in a number of different ways, depending on computational cost and code formulation. The iteration of Eq. (11) simply implies that the interface forces can be updated (to satisfy the compatibility conditions) using only the kinematic information at the interface, provided that the gluing matrix is obtained. The key issue of how to obtain the lambda matrix in a systematic and efficient way based on the interface information is beyond the scope of this paper; details can be found in Wang et al. [5].

Various time stepping methods to advance the subsystem solutions can be considered depending on whether or not there is a leading subsystem and how the time steps are arranged for the information exchange between the different models. There are three typical iteration methods, namely, *parallel, leading*, and *walking*. For leading and walking, the compatibility of the whole system is not explicitly required as the simulation steps forward. Thus, they actually represent sequential methods. In the parallel method, components exchange information within the time step, and the coupled subsystems are compatible at the end of each time step. In general, the parallel time stepping method is preferred when there is no clear distinction between the different systems in the cosimulation.

5 Finite Element Based Direct Integration

As mentioned earlier, multibody systems feature two key characteristics: the presence of finite relative rotations and mechanical joints that impose restrictions on their motion. It must be noted that the presence of finite rotations is a modeling choice: indeed, it is always possible to model all components of a system as three-dimensional structures, the deformations of which are adequately described by a three-dimensional displacement field. In turn, this displacement field defines the deformation gradient tensor, and, finally, the finite rotation of the differential element is uniquely extracted from this tensor with the help of the polar decomposition theorem (Bonet and Wood [24]; Spencer [25]). In other words, the three-dimensional displacement field inherently defines the finite rotation of a differential element. While the systematic use of three-dimensional elements, "brick" elements, would considerably simplify the integration of multibody and finite element formulations, this is not a practical approach, as it would require exorbitant computational resources. Clearly, it is highly desirable to use structural elements, i.e., beams, plates, and shells. These plate and shell elements have been the subject of intense research for several decades although initially developed by Kirchhoff and then by Love more than 100 years ago. An excellent survey of this research is found in Naghdi [26]. Beam formulations have received similar attention (Cesnik et al. [27]).

5.1 Formulation of Structural Elements. The difficulties associated with the shell theory are numerous. First, the shell theory is not exact: the accuracy of a proposed theory mainly depends on the assumption concerning a through-the-thickness behavior of specific quantities. Second, due to the complicated geometric representation of a shell, few exact solutions exist, and those are for idealized situations. Third, shells are made with an ever-increasing array of materials. The heterogeneity and anisotropy of such materials render the analysis more and more involved and cause the older theories for homogeneous and isotropic materials to become outdated. Finally, the implementation of new analytical models into the FEM has encountered a number of difficult issues such as shear locking, membrane locking, and the need to connect shell elements having five degrees of freedom to those with six (Stolarski et al. [28]). It should come as no surprise that the development of beam and shell elements that are compatible with multibody formulation is a major endeavor. Existing beam and shell models fall into two categories: degenerate solid approaches and

direct approaches or "Cosserat surface" models.

In the *degenerate solid approaches*, the original three-dimensional problem is mapped onto a two-dimensional (2D) surface (Kratzig [29]). This mapping can be achieved in two ways: through a set of ad hoc assumptions or through asymptotic expansions. The Kirchhoff-type and Reissner-Mindlin-type theories are based on ad hoc assumptions that characterize the three-dimensional quantities defined in the framework of elasticity in terms of 2D quantities defined on a chosen reference surface. The governing equations for the 2D boundary value problem are then derived in a straightforward manner; integrating through the thickness eliminates the thickness coordinate. The main drawback of this approach is that there is no justification for the ad hoc assumptions that often violate basic elasticity requirements. On the other hand, theories based on asymptotic expansions take advantage of the smallness of the thickness relative to the wavelength of deformation in the shell and relative to the radii of curvature of the undeformed shell (Goldenweizer [30]). They use asymptotic expansions through the thickness to deduce a series of 2D problems corresponding to different kinds of deformation of shells, such as membrane, bending, and boundary effects. Although these theories are mathematically elegant and do not introduce ad hoc assumptions, it is hard to identify to which behavior the equations deduced from a certain order correspond (Stolarski et al. [28]), which is especially the case for composite shells. Also, a numerical implementation of such theories can be very difficult, if not impossible.

Direct approaches or Cosserat surfaces model the three-dimensional shell structure as a 2D "oriented" continuum, which could be described as a surface associated with a director, without starting from three-dimensional continuum mechanics. All the physical laws for three-dimensional bodies such as conservation of mass or balance of momentum are assumed for the quantities defined on the surface (Libai and Simmonds [31]). It is appropriate to refer to these theories as "2D surface mechanics theories" (Kratzig [29]). It is obvious that these theories isolate themselves from the three-dimensional continuum, and their intricate formulation often obscures physical interpretation.

Both degenerate solid and direct approaches can be implemented using the *total Lagrangian* or *updated Lagrangian* method (Bathe [32]). While the choice between the two approaches is a matter of computational efficiency when elastic material constitutive laws are used, one method might become better suited than the other when using nonlinear constitutive laws. For instance, for problems involving large displacements and rotations but small strain, the total Lagrangian formulation is convenient, whereas the updated Lagrangian formulation is more natural when using the incremental formulation of plasticity.

While the above discussion focuses on shell theory, the same distinctions can be found for beam theories. In fact, much of the work done to adapt structural elements to multibody formulations focused on beams.

5.2 Selection of Nodal Degrees of Freedom. Once a structural element approach has been selected, the next step in the development of a finite element is the selection of nodal degrees of freedom. Typically, beam theory employs the position of a point on its reference axis and an orthonormal triad defining the orientation of the cross-sectional plane; shell theory employs the position of a point on its reference plane and a unit vector, "director," normal to this plane. If a minimum set of parameters is used, this gives six and five degrees of freedom per node for the beam and shell, respectively. Of course, it is not required to use a minimum set of parameters to characterize finite rotations: for instance, the nine direction cosines could be used as nodal degrees of freedom.

Parametrization of structural elements is aligned closely with the modeling of rigid body dynamics, a field at the heart of the development of multibody dynamics. Rigid bodies were initially represented by six degrees of freedom, three displacements, and three rotations, often selected to be Euler angles. In view of the singularities associated with these angles, the use of Euler parameters was suggested (Huston [71], Nikravesh et al. [33]); later, the efficiency of "natural coordinates" was demonstrated (García de Jalón et al. [34]); finally, the use of direction cosines was also proposed (Betsch and Steinmann [35]). Similar types of parametrizations were used to develop beam finite elements: the Cayley parametrization (Simo and Vu-Quoc [11]), the finite rotation vector (Cardona and Géradin [36]), and the direction cosines (Betsch et al. [37]), to name just a few examples.

5.3 Algebraic Constraint Equations. The use of finite rotation as nodal variables is an important step toward integrating finite elements and multibody system formulations. However, multibody systems are characterized by constraint equations that must be accommodated in the solution process. In multibody system formulations, the coordinate partitioning technique has been widely used for dealing with constraint equations (Wehage [12]; Wehage and Haug [13]); a number of more efficient and reliable coordinate reduction techniques were later developed by a number of authors; many of the approaches are related to Maggi's formulation of constraint dynamics (Kurdila et al. [38]). Unfortunately, many of these techniques destroy the banded character of the system equations of motion although the coordinate partitioning can be formulated in a manner that retains the bandwidth of the system. While this is of little consequence when dealing with a small number of degrees of freedom, as is the case for rigid multibody systems (Unda et al. [39]), it would drastically impact the performance of finite element codes that rely on sparse matrix solution techniques (Bathe [32]). On the other hand, the augmented Lagrangian formulation proposed by Bayo et al. [9] and Bayo and Ledesma [10] are readily implemented within the framework of finite element solution procedures with little computational penalty. Géradin and Cardona [7] further refined the approach and addressed the numerical issues associated with the formulation.

5.4 Time Integration Algorithms. Since multibody systems are inherently dynamic problems, the last aspect of the numerical implementation is the choice of a time integration algorithm to solve the DAEs of motion. At first, a distinction must be made between explicit and implicit schemes. Typically, explicit schemes are conditionally stable; i.e., the integration process is stable if the time step size Δt satisfies the condition $\Delta t < \alpha/\omega_{\max}$, where α is a constant and $\omega_{\max}$ is the highest frequency of the system. Traditional multibody system models use a small number of degrees of freedom, and the highest frequency of the system is not very large; the stability condition then leads to a reasonable time step size. The Adams-Bashforth predictor-corrector explicit scheme has been widely used for the time integration of the multibody system equations of motion. Furthermore, as discussed above, the coordinate partitioning method is often used to obtain a minimum set of equations, eliminating the problem of algebraic constraints. This approach contrasts with that used in finite element codes. First, FEMs often feature a very large number of degrees of freedom due to the fine meshes used in the discretization. The highest frequency of the model is inversely proportional to the square of the size of the smallest element in the mesh; that is, the highest frequency is an artifact of the discretization process. The use of explicit schemes would prohibitively restrict the time step size, resulting in very high computational costs. Consequently, implicit schemes are used for a finite element structural dynamic analysis. The Hilber-Hughes-Taylor (HHT) scheme (Hilber et al. [40]) is the workhorse of most commercial codes, and the more recent generalized-α method (Chung and Hulbert[41]) further improves its dissipation characteristics. Note that for finite element problems, high frequency numerical dissipation is considered to be an indispensable feature of the time integration algorithm. Since finite element formulations do not typically rely on the coordinate partitioning method, the presence of algebraic constraints and the associated Lagrange multipliers must be reckoned with. Cardona

and Géradin [42] and Farhat et al. [43] have shown that the use of the HHT algorithm stabilizes the integration process in the presence of Lagrange multipliers. Lagrange multipliers have been extensively used in the finite element formulation and are the basis for a powerful parallel implementation of the method (Farhat and Roux [18]). A state-of-the-art review of time integration algorithms for structural dynamics can be found in (Hulbert [44]).

5.5 Review of Existing Implementations. The previous sections have described the many avenues that could be pursued to implement finite element based multibody formulations. Clearly, many choices and combinations are possible but researchers have thus far explored two main directions. The first approach, FEBDI, is discussed in this section, while the second approach is the basis for the MSBDI method discussed in the following section.

The first approach uses one-dimensional and 2D oriented media formulations for beam and shell elements, respectively, and the nodal degrees of freedom are three displacements and two or three rotations, for beam and shell elements, respectively. This approach builds upon the corotational formulation of Simo and Vu-Quoc [11] for beams; the approach is sometimes called the "geometrically exact formulation." Beam elements were presented by Cardona and Géradin [35], Bauchau et al. [45], and Bottasso and Borri [46], whereas Simo and Tarnow [47] and Bauchau et al. [48] developed shell elements based on the same approach. The problem of algebraic constraints in multibody systems was addressed by Cardona and Géradin [41], Bauchau et al. [45], and Bottasso et al. [49]. Most of this work was developed within the framework of energy preserving/decaying time integration schemes that provide a mathematical proof of stability for nonlinear multibody systems (Bauchau [6], Bottasso et al. [50], Bauchau et al. [51]). Within this framework, the formulations of a number of special features necessary to treat realistic engineering problems were presented: modeling of hydraulic actuators (Cardona and Géradin [52]), joint flexibility (Cardona et al. [53]) and clearance (Bauchau and Rodriguez [54]), flexible components based on component mode synthesis (Cardona and Géradin [55]), composite material behavior (Bauchau and Hodges [56]), and friction and rolling (Bauchau [57]).

6 Multibody System Based Direct Integration

The large deformation finite element approach that has been used in the last several years to examine the MSBDI method is the absolute nodal coordinate formulation. While this formulation suffers from locking problems in some application, it has desirable features that can be exploited to develop a new generation of computational algorithms that can be effectively used in solving complex and challenging problems, as demonstrated by preliminary studies (Dmitrochenko and Pogorelov [58], Garcia-Vallejo et al. [59], Garcia-Vallejo et al. [60], Gerstmayr and Shabana [61], Mikkola and Matikainen [62], Schwab and Meijaard [63], Sopanen and Mikkola [64], Takahashi and Shimizu [65], Von Dombrowski [66], and Yoo et al. [67]). The absolute nodal coordinate formulation, as reported in the literature, allows for using more general constitutive equations, correctly describes the rigid body motion, defines a unique rotation field, leads to a constant mass matrix, and automatically satisfies the principle of work and energy. Solutions for these locking problems have been proposed in the literature. Since the mass matrix obtained using the absolute nodal coordinate formulation is constant, the centrifugal and Coriolis forces are identically equal to zero. Furthermore, one can take advantage of the constant mass matrix to develop an optimum sparse matrix structure of the dynamic equations of motion. It is important to mention that if another set of coordinates, as in the case of the floating frame of reference formulation, is used, the mass matrix is nonlinear and the centrifugal and Coriolis forces appear explicitly in the dynamic equations.

In the absolute nodal coordinate formulation, the displacement field of finite element j of a very flexible body i can be written as

$$\mathbf{r}^{ij} = \mathbf{S}^{ij}\mathbf{e}^{ij} \tag{12}$$

In this equation, $\mathbf{r}^{ij}$ is the global position vector of an arbitrary point on the element j of the very flexible body i, $\mathbf{S}^{ij}$ is the finite element shape function, and $\mathbf{e}^{ij}$ is the vector of element nodal coordinates that include absolute position and slope coordinates. Using the kinematic description of Eq. (12), the equation of motion of the finite element can be written as follows:

$$\mathbf{M}_a^{ij}\ddot{\mathbf{e}}^{ij} = \mathbf{Q}_a^{ij} \tag{13}$$

where subscript a refers to absolute coordinates, $\mathbf{M}_a^{ij}$ is the constant symmetric mass matrix, and $\mathbf{Q}_a^{ij}$ is the vector of nodal forces that also include the elastic forces. Unlike the floating frame of reference formulation, the absolute nodal coordinate formulation leads to a constant mass matrix and to a nonlinear expression of the elastic forces. Using the preceding equation, the equations of motion of all the bodies modeled using the absolute nodal coordinate formulation can be written as

$$\mathbf{M}_a\ddot{\mathbf{e}} = \mathbf{Q}_a \tag{14}$$

In this equation, $\ddot{\mathbf{e}}$ is the vector of system absolute accelerations, $\mathbf{M}_a$ is the system constant and symmetric mass matrix, and $\mathbf{Q}_a$ is the vector of nodal forces.

An efficient integration of large and small deformation finite element and multibody algorithms necessitates the use of different formulations and coordinate types in order to be able to solve systems that consist of bulky rigid bodies, flexible bodies that undergo small deformations, and very flexible bodies that undergo large deformations. Therefore, the new MSBDI algorithms should integrate the floating frame of reference formulation, which can efficiently be used to model rigid bodies and flexible bodies and the absolute nodal coordinate formulation that can be used to model very flexible bodies. In this case, one needs to use the set of coordinates $\mathbf{q}_r$ to describe the rigid body motion, the mixed set of coordinates $\mathbf{q}_r$ and $\mathbf{q}_f$ to describe the small deformation, and the set of absolute coordinates $\mathbf{e}$ to describe the large deformation. Since the mass matrix associated with the absolute coordinates $\mathbf{e}$ is constant, a Cholesky transformation can be used to write the vector of absolute coordinates $\mathbf{e}$ in terms of a new set of Cholesky coordinates $\mathbf{q}_{\mathrm{Ch}}$ (Shabana [68]). This coordinate transformation leads to an identity generalized inertia matrix associated with the Cholesky coordinates.

In multibody system applications, nongeneralized coordinates can be introduced, examples of which are the geometric surface parameters that define the wheel and rail profiles in railroad vehicle systems. Such nongeneralized coordinates will be denoted as $\mathbf{s}$. Using Eqs. (8) and (14) and including the nongeneralized coordinates, which have no inertia or forces associated with them, the augmented form of the equations of motion can be written as follows:

$$\begin{bmatrix} \mathbf{M}_{rr} & \mathbf{M}_{rf} & \mathbf{0} & \mathbf{0} & \mathbf{C}^r \\ \mathbf{M}_{fr} & \mathbf{M}_{ff} & \mathbf{0} & \mathbf{0} & \mathbf{C}_{\mathbf{q}_f}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{C}_{\mathbf{q}_{\mathrm{Ch}}}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C}_{\mathbf{s}}^T & \mathbf{0} \\ \mathbf{C}_{\mathbf{q}_r} & \mathbf{C}_{\mathbf{q}_f} & \mathbf{C}_{\mathbf{q}_{\mathrm{Ch}}} & \mathbf{C}_{\mathbf{s}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_r \\ \ddot{\mathbf{q}}_f \\ \ddot{\mathbf{q}}_{\mathrm{Ch}} \\ \ddot{\mathbf{s}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_r \\ \mathbf{Q}_f \\ \mathbf{Q}_{\mathrm{Ch}} \\ \mathbf{0} \\ \mathbf{Q}_c \end{bmatrix} \tag{15}$$

In this sparse matrix structure, subscripts r, f, and Ch refer, respectively, to reference, flexible (small deformation), and Cholesky coordinates; $\mathbf{M}$ refers to the mass submatrix; $\mathbf{I}$ is the identity matrix; $\mathbf{Q}$ refers to a force vector; $\mathbf{Q}_c$ is the vector that arises from the differentiation of the constraint equations twice with respect to time; and $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. The preceding equation can be solved for the accelerations and Lagrange multipliers. Lagrange multipliers can be used to determine the constraint forces. The independent accelerations can be identified and integrated forward in time using direct numerical methods to determine the independent coordinates and velocities.

The dependent coordinates and velocities can be determined using the constraint equations at the position and velocity levels, respectively. This nonincremental solution procedure can also be effectively used for the large deformation analysis of very flexible bodies and allows for modeling joint constraints as well as other nonlinear constraints and forcing functions.

It is important to point out that, as an alternative to the use of Eq. (15), the constraint equations can be used to systematically eliminate the dependent variables. This can be accomplished using the embedding technique, which leads to a velocity transformation matrix that can be used to define the generalized inertia matrix and generalized forces associated with the independent variables. When the embedding technique is used, however, the sparse matrix structure defined by Eq. (15) is lost since the inertia of the system is projected onto the space spanned by the independent variables only.

In the three-dimensional absolute nodal coordinate formulations for beam and shell elements, the nodal degrees of freedom are three displacements and three orientation vectors, which are not constrained to be unit nor mutually orthogonal. Lower order elements that employ smaller numbers of coordinates for each node are also proposed in the literature. Preliminary studies on and results obtained using this formulation have underlined its desirable features that make it ideally suited for integrating large deformation finite element capabilities with multibody system algorithms. Within this framework, beam and cable elements were studied by Sugiyama et al. [69] and Gerstmayr and Shabana [61]; plates and shells by Dmitrochenko and Pogorelov [58] and Yoo et al. [67]. In the absolute nodal coordinate approach, different formulations can be used to derive the elastic forces in an element; various efforts in this area are reported by Takahashi and Shimizu [65], Sopanen and Mikkola [64], and Mikkola and Matikainen [62]. Multibody systems were addressed by Von Dombrowski [66], while Garcia-Vallejo et al. [59] focused on rigid-flexible systems. The issue of internal damping in structures was investigated by Garcia-Vallejo et al. [60].

Schwab and Meijaard [63] and Romero [70] presented preliminary comparisons of the performance of the geometrically exact and absolute nodal coordinate approaches. Both approaches correctly describe rigid body motion and constant strain states, a requirement for the convergence of FEMs. The absolute nodal coordinate approach is easier to formulate and implement than the geometrically exact approach due in great part to the absence of finite rotation variables in the former approach. In particular, the absolute nodal coordinate approach leads to a constant mass matrix, as previously mentioned, whereas the geometrically exact formulation leads to nonlinear inertial terms stemming from the rotation of the inertia tensor from local to global frames and from the presence of nonlinear gyroscopic terms. When using explicit time integration schemes, a constant mass matrix results in a significant computation advantage; however, it is unclear whether this advantage is retained when using implicit schemes. The overall computational efficiency of a finite element simulation is driven by the factorization of the system matrix at each time step. A rough estimate of the cost of this operation is $C=nm^2$, where n is the total number of degrees of freedom and m is the average bandwidth of the system (Bathe [32]). Keeping in mind that the absolute nodal coordinate formulation and the geometrically exact formulation are fundamentally different since the former allows for the deformation of the cross section, typical absolute nodal coordinate formulations use 12 degrees of freedom per node of a beam, three displacements, and nine slopes, whereas the geometrically exact formulation uses six degrees of freedom per node, three displacements, and three rotations. A rough estimate of the relative costs of the two approaches is then $C_{\mathrm{ANC}}/C_{\mathrm{GE}}=2^3=8$. The higher cost of the solution process is only partially offset by the lower cost of generation of element matrices. In view of the simplicity of the absolute nodal coordinate formulation, energy and momentum preserving schemes are easily developed for this approach; in fact, the simple trapezoidal integration rule is energy and momentum preserving for this formulation. These desirable features of time integration schemes are much harder to obtain when dealing with the geometrically exact formulation; the large number of papers devoted to this problem is a testimony to the difficulty of this task. Finally, it must be noted that the absolute nodal coordinate formulation, in its original form, suffers from shear and membrane locking. The many remedies to this problem that have been developed for various types of finite elements must be adapted to this formulation.

7 Summary and Conclusions

Existing finite element and multibody system algorithms have been pushed to their limits and are no longer effective in solving detailed and complex models that represent many physics and engineering applications. The analysis of these detailed and complex models requires the successful integration of large deformation finite element and multibody system algorithms. Three methods can be used to integrate the finite element and multibody system algorithms. The first is based on the GA, in which a link between existing finite element and multibody system algorithms is established. Since the two types of algorithms employ fundamentally different formulations and solution procedures, the GA leads to several problems that have been addressed in the literature. The second approach is to use the FEBDI, or the geometrically exact formulation, in which multibody features are implemented in the finite element algorithm. To avoid the problems associated with the description of the finite reference rotations, large rotation vector formulations are used. In the third approach, a MSBDI is used. In this method, a large deformation finite element formulation that can be used with a nonincremental solution procedure is integrated with multibody system algorithms that include different formulations and coordinate types to handle in an efficient manner detailed and complex models that consist of different bodies with different degrees of flexibility. To identify the most efficient formulation, a systematic comparison of all the aspects of the problem must be performed. Benchmark problems should be carefully established and all aspects, accuracy, versatility, and computational cost of the proposed formulations should be evaluated. It is possible that different solution strategies and formulations might be better suited for different problem types.

Acknowledgment

This research was supported by the U.S. Army Research Office. This financial support is gratefully acknowledged.

References

[1] Belytschko, T., Liu, W. K., and Moran, B., 2000, *Nonlinear Finite Elements for Continua and Structures*, Wiley, New York.
[2] Zienkiewicz, O. C., 1977, *The Finite Element Method*, 3rd ed., McGraw-Hill, New York.
[3] Zienkiewicz, O. C., and Taylor, R. L., 2000, *Solid Mechanics*, (The Finite Element Method Vol. 2) 5th ed., Butterworth-Heinemann, London.
[4] Tseng, F. C., and Hulbert, G. M., 2001, "A Gluing Algorithm for Network-Distributed Dynamics Simulation," Multibody Syst. Dyn., **6**, pp. 377–396.
[5] Wang, J. Z., Ma, Z. D., and Hulbert, G. M., 2003, "A Gluing Algorithm for Distributed Simulation of Multibody Systems," Nonlinear Dyn., **34**, pp. 159–188.
[6] Bauchau, O. A., 1998, "Computational Schemes for Flexible, Nonlinear Multi-Body Systems," Multibody Syst. Dyn., **2**(2), pp. 169–225.
[7] Géradin, M., and Cardona, A., 2001, *Flexible Multibody Dynamics: A Finite Element Approach*, Wiley, New York.
[8] Shabana, A., 1996, "Finite Element Incremental Approach and Exact Rigid Body Inertia," ASME J. Mech. Des., **118**(2), pp. 171–178.
[9] Bayo, E., García de Jalón, J., and Serna, M. A., 1988, "A Modified Lagrangian Formulation for the Dynamic Analysis of Constrained Mechanical Systems," Comput. Methods Appl. Mech. Eng., **71**, pp. 183–195.
[10] Bayo, E., and Ledesma, R., 1996, "Augmented Lagrangian and Mass-Orthogonal Projection Methods for Constrained Multibody Dynamics," Nonlinear Dyn., **9**, pp. 113–130.
[11] Simo, J. C., and Vu-Quoc, L., 1986, "On the Dynamics of Flexible Beams Under Large Overall Motions—The Plane Case: Parts I and II," ASME J. Appl. Mech., **53**, pp. 849–863.
[12] Wehage, R. A., 1980, "Generalized Coordinate Partitioning in Dynamic Analy-

sis of Mechanical Systems," Ph.D. thesis, University of Iowa, Iowa.
[13] Wehage, R. A., and Haug, E. J., 1982, "Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems," ASME J. Mech. Des., **104**(1), pp. 247–255.
[14] Shabana, A. A., 2005, *Dynamics of Multibody Systems*, 3rd ed., Cambridge University Press, Cambridge, England.
[15] Agrawal, O. P., and Shabana, A. A., 1985, "Dynamic Analysis of Multi-Body Systems Using Component Modes," Comput. Struct., **21**(6), pp. 1303–1312.
[16] Farhat, C. H., and Wilson, E., 1988, "A Parallel Active Column Equation Solver," Comput. Struct., **28**, pp. 289–304.
[17] Farhat, C. H., and Roux, F. X., 1994, "Implicit Parallel Processing in Structural Mechanics," Comput. Mech. Adv., **2**, pp. 1–124.
[18] Farhat, C., and Roux, F. X., 1991, "A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm," Int. J. Numer. Methods Eng., **32**, pp. 1205–1227.
[19] Modak, S., and Sotelino, E. D., 2000, "Iterative Group Implicit Algorithm for Parallel Transient Finite Element Analysis," Int. J. Numer. Methods Eng., **47**(4), pp. 869–885.
[20] Kim, S. S., 2002, "A Subsystem Synthesis Method for an Efficient Vehicle Multibody Dynamics," Multibody Syst. Dyn., **7**, pp. 189–207.
[21] Anderson, K. S., and Duan, S., 2000, "Highly Parallelizable Low-Order Dynamics Simulation Algorithm for Multi-Rigid-Body Systems," J. Guid. Control Dyn., **23**(2), pp. 355–364.
[22] Kübler, R., and Schiehlen, W., 2000, "Modular Simulation in Multibody System Dynamics," Multibody Syst. Dyn., **4**, pp. 107–127.
[23] Tseng, F. C., Ma, Z. D., and Hulbert, G. M., 2003, "Efficient Numerical Solution of Constrained Multibody Dynamics Systems," Comput. Methods Appl. Mech. Eng., **192**, pp. 439–472.
[24] Bonet, J., and Wood, R. D., 1997, *Nonlinear Continuum Mechanics for Finite Element Analysis*, Cambridge University Press, Cambridge, England.
[25] Spencer, A. J. M., 1980, *Continuum Mechanics*, Longman, London.
[26] Naghdi, P. M., 1972, "The Theory of Shells and Plates," *Handbuch der Physik*, Springer-Verlag, Berlin, Vol. 6, pp. 425–640.
[27] Cesnik, C. E. S., Hodges, D. H., and Sutyrin, V. G., 1996, "Cross-Sectional Analysis of Composite Beams Including Large Initial Twist and Curvature Effects," AIAA J., **34**(9), pp. 1913–1920.
[28] Stolarski, H., Belytschko, T., and Lee, S. H., 1995, "A Review of Shell Finite Elements and Corotational Theories," Comput. Mech. Adv., **2**(2), pp. 125–212.
[29] Kratzig, W. B., 1993, "'Best' Transverse Shearing and Stretching Shell Theory for Nonlinear Finite Element Simulations," Comput. Methods Appl. Mech. Eng., **103**(1–2), pp. 135–160.
[30] Goldenweizer, A., 1961, *Theory of Thin Elastic Shells*, Pergamon, Oxford, UK.
[31] Libai, A., and Simmonds, J. G., 1998, *The Nonlinear Theory of Elastic Shells*, 2nd ed., Cambridge University Press, Cambridge, England.
[32] Bathe, K. J., 1996, *Finite Element Procedures*, Prentice-Hall, Englewood Cliffs, NJ.
[33] Nikravesh, P. E., Wehage, R. A., and Kwon, O. K., 1985, "Euler Parameters in Computational Dynamics and Kinematics. Part I and Part II," ASME J. Mech., Transm., Autom. Des., **107**(3), pp. 358–369.
[34] García de Jalón, J., Unda, J., Avello, A., and Jiménez, J. M., 1987, "Dynamic Analysis of Three-Dimensional Mechanisms in "Natural" Coordinates," ASME J. Mech., Transm., Autom. Des., **109**, pp. 460–465.
[35] Betsch, P., and Steinmann, E., 2001, "Constrained Integration of Rigid Body Dynamics," Comput. Methods Appl. Mech. Eng., **191**, pp. 467–488.
[36] Cardona, A., and Géradin, M., 1988, "A Beam Finite Element Non-Linear Theory With Finite Rotation," Int. J. Numer. Methods Eng., **26**, pp. 2403–2438.
[37] Betsch, P., Menzel, A., and Stein, E., 1998, "On the Parametrization of Finite Rotations in Computational Mechanics. A Classification of Concepts With Application to Smooth Shells," Comput. Methods Appl. Mech. Eng., **155**(3–4), pp. 273–305.
[38] Kurdila, A., Papastavridis, J. G., and Kamat, M. P., 1990, "Role of Maggi's Equations in Computational Methods for Constrained Multibody Systems," J. Guid. Control Dyn., **13**(1), pp. 113–120.
[39] Unda, J., García de Jalón, J., Losantos, F., and Enparantza, R., 1987, "A Comparative Study on Some Different Formulations of the Dynamics Equations of Constrained Mechanical Systems," ASME J. Mech., Transm., Autom. Des., **109**, pp. 466–474.
[40] Hilber, H. M., Hughes, T. J. R., and Taylor, R. L., 1977, "Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics," Earthquake Eng. Struct. Dyn., **5**, pp. 282–292.
[41] Chung, J., and Hulbert, G. M., 1993, "A Time Integration Algorithm for Structural Dynamics With Improved Numerical Dissipation: The Generalized-α Method," ASME J. Appl. Mech., **60**, pp. 371–375.
[42] Cardona, A., and Géradin, M., 1989, "Time Integration of the Equations of Motion in Mechanism Analysis," Comput. Struct., **33**(3), pp. 801–820.
[43] Farhat, C., Crivelli, L., and Géradin, M., 1995, "Implicit Time Integration of a Class of Constrained Hybrid Formulations—Part I: Spectral Stability Theory," Comput. Methods Appl. Mech. Eng., **125**, pp. 71–107.
[44] Hulbert, G. M., 2004, "Computational Structural Dynamics," in *Encyclopedia of Computational Mechanics*, E. Stein, R. de Borst, and T. J. R. Hughes, eds., Vol. 2, pp. 169–193.
[45] Bauchau, O. A., Damilano, G., and Theron, N. J., 1995, "Numerical Integration of Nonlinear Elastic Multi-Body Systems," Int. J. Numer. Methods Eng., **38**, pp. 2727–2751.
[46] Bottasso, C. L., and Borri, M., 1997, "Energy Preserving/Decaying Schemes for Non-Linear Beam Dynamics Using the Helicoidal Approximation," Comput. Methods Appl. Mech. Eng., **143**, pp. 393–415.
[47] Simo, J. C., and Tarnow, N., 1994, "A New Energy and Momentum Conserving Algorithm for the Nonlinear Dynamics of Shells," Int. J. Numer. Methods Eng., **37**, pp. 2527–2549.
[48] Bauchau, O. A., Choi, J. Y., and Bottasso, C. L., 2002, "On the Modeling of Shells in Multibody Dynamics," Multibody Syst. Dyn., **8**(4), pp. 459–489.
[49] Bottasso, C. L., Borri, M., and Trainelli, L., 2001, "Integration of Elastic Multibody Systems by Invariant Conserving/Dissipating Algorithms—Part I: Formulation," Comput. Methods Appl. Mech. Eng., **190**, pp. 3669–3699.
[50] Bottasso, C. L., Borri, M., and Trainelli, L., 2001, "Integration of Elastic Multibody Systems by Invariant Conserving/Dissipating Algorithms—Part II: Numerical Schemes and Applications," Comput. Methods Appl. Mech. Eng., **190**, pp. 3701–3733.
[51] Bauchau, O. A., Bottasso, C. L., and Trainelli, L., 2003, "Robust Integration Schemes for Flexible Multibody Systems," Comput. Methods Appl. Mech. Eng., **192**(3–4), pp. 395–420.
[52] Cardona, A., and Géradin, M., 1990, "Modeling of a Hydraulic Actuator in Flexible Machine Dynamics Simulation," Mech. Mach. Theory, **25**(2), pp. 193–207.
[53] Cardona, A., Géradin, M., and Doan, D. B., 1991, "Rigid and Flexible Joint Modelling in Multi-Body Dynamics Using Finite Elements," Comput. Methods Appl. Mech. Eng., **89**, pp. 395–418.
[54] Bauchau, O. A., and Rodriguez, L., 2002, "Modeling of Joints With Clearance in Flexible Multibody Systems," Int. J. Solids Struct., **39**, pp. 41–63.
[55] Cardona, A., and Géradin, M., 1992, "A Superelement Formulation for Mechanism Analysis," Comput. Methods Appl. Mech. Eng., **100**, pp. 1–29.
[56] Bauchau, O. A., and Hodges, D. H., 1999, "Analysis of Nonlinear Multi-Body Systems With Elastic Couplings," Multibody Syst. Dyn., **3**, pp. 168–188.
[57] Bauchau, O. A., 1999, "On the Modeling of Friction and Rolling in Flexible Multi-Body Systems," Multibody Syst. Dyn., **3**, pp. 209–239.
[58] Dmitrochenko, O. N., and Pogorelov, D. Y., 2003, "Generalization of Plate Finite Elements for Absolute Nodal Coordinate Formulation," Multibody Syst. Dyn., **10**(1), pp. 17–43.
[59] Garcia-Vallejo, D., Escalona, J. L., Mayo, J., and Dominguez, J., 2003, "Describing Rigid-Flexible Multibody Systems Using Absolute Coordinates," Nonlinear Dyn., **34**(1–2), pp. 75–94.
[60] Garcia-Vallejo, D., Valverde, J., and Dominguez, J., 2005, "An Internal Damping Model for the Absolute Nodal Coordinate Formulation," Nonlinear Dyn., **42**(4), pp. 347–369.
[61] Gerstmayr, J., and Shabana, A. A., 2006, "Analysis of Thin Beams and Cables Using the Absolute Nodal Coordinate Formulation," Nonlinear Dyn., **45**, pp. 109–130.
[62] Mikkola, M. A., and Matikainen, M. K., 2006, "Development of Elastic Forces for a Large Deformation Plate Element Based on the Absolute Nodal Coordinate Formulation," ASME J. Comput. Nonlinear Dyn., **1**(2), pp. 103–108.
[63] Schwab, A. L., and Meijaard, J. P., 2005, "Comparison of Three-Dimensional Beam Elements for Dynamic Analysis: Finite Element Method and Absolute Nodal Coordinate Formulation," *Proceedings of the ASME 2005 International Design Engineering Technical Conferences and Computer and Information in Engineering Conference* (DETC2005-85104), Long Beach, CA, Sept. 24–28, Paper. No. DETC2005-85104.
[64] Sopanen, J. T., and Mikkola, A. M., 2003, "Description of Elastic Forces in Absolute Nodal Coordinate Formulation," Nonlinear Dyn., **34**(1–2), pp. 53–74.
[65] Takahashi, Y., and Shimizu, N., 1999, "Study on Elastic Forces of the Absolute Nodal Coordinate Formulation for Deformable Beams," *Proceedings of ASME International Design Engineering Technical Conferences and Computer and Information in Engineering Conference*, Las Vegas, NV.
[66] Von Dombrowski, S., 2002, "Analysis of Large Flexible Body Deformation in Multibody Systems Using Absolute Coordinates," Multibody Syst. Dyn., **8**(4), pp. 409–432.
[67] Yoo, W. S., Lee, J. H., Park, S. J., Sohn, J. H., Pogorelov, D., and Dimitrochenko, O., 2004, "Large Deflection Analysis of a Thin Plate: Computer Simulation and Experiment," Multibody Syst. Dyn., **11**(2), pp. 185–208.
[68] Shabana, A. A., 1998, "A Computer Implementation of the Absolute Nodal Coordinate Formulation for Flexible Multibody Dynamics," Nonlinear Dyn., **16**(3), pp. 293–306.
[69] Sugiyama, H., Mikkola, A. M., and Shabana, A. A., 2003, "A Non-Incremental Nonlinear Finite Element Solution for Cable Problems," ASME J. Mech. Des., **125**, pp. 746–756.
[70] Romero, I., 2006, "A Study of Nonlinear Rod Models for Flexible Multibody Dynamics," *Proceedings of the Seventh World Congress on Computational Mechanics*, Los Angeles, CA, July 16–22.
[71] Huston, R. L., 1981, "Multibody Dynamics Including the Effect of Flexibility and Compliance," Comput. Struct., **14**, pp. 443–451.