Published in final edited form as:

Int J Non Linear Mech. 2013 December ; 57: 146–153. doi:10.1016/j.ijnonlinmec.2013.07.002.

A Logarithmic Complexity Floating Frame of Reference Formulation with Interpolating Splines for Articulated Multi-Flexible-Body Dynamics

I.M. Khan^a, W. Ahn^b, K.S. Anderson^a, and S. De^b

Department of Mechanical Aerospace and Nuclear Engineering, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180 USA

I.M. Khan: khani2@rpi.edu; W. Ahn: ahnw@rpi.edu; K.S. Anderson: anderk5@rpi.edu; S. De: des@rpi.edu

Abstract

An interpolating spline-based approach is presented for modeling multi-flexible-body systems in the divide-and-conquer (DCA) scheme. This algorithm uses the floating frame of reference formulation and piecewise spline functions to construct and solve the non-linear equations of motion of the multi-flexible-body system undergoing large rotations and translations. The new approach is compared with the flexible DCA (FDCA) that uses the assumed modes method [1]. The FDCA, in many cases, must resort to sub-structuring to accurately model the deformation of the system. We demonstrate, through numerical examples, that the interpolating spline-based approach is comparable in accuracy and superior in efficiency to the FDCA. The present approach is appropriate for modeling flexible mechanisms with thin 1D bodies undergoing large rotations and translations, including those with irregular shapes. As such, the present approach extends the current capability of the DCA to model deformable systems. The algorithm retains the theoretical logarithmic complexity inherent in the DCA when implemented in parallel.

Keywords

Multi-Flexible-Body Systems; Logarithmic Complexity; Divide-and-Conquer Algorithm; Interpolating Splines

1. Introduction

Modeling flexibility in the dynamic simulation of multibody systems often becomes unavoidable when its effects are significant, e.g, systems including articulated elastic objects such as bars, beams, shafts, and belts in application areas such as robotics and micro-electromechanical systems (MEMS). The goal of these simulations is to capture the essential dynamics of the system arising from the large overall motion, often regarded as rigid body motion associated with the kinematic joints between the flexible bodies, and that from the

© 2013 Elsevier Ltd. All rights reserved.

Correspondence to: I.M. Khan, khani2@rpi.edu.

^aComputational Dynamics Laboratory

^bThe Center for Modeling, Simulation, and Imaging in Medicine

URL: <http://www.seer.rpi.edu/cemsim/> (S.De)

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

elastic deformation of the individual components of the system. In such simulations, one often encounters situations where computational efficiency as well as model accuracy are important and a balance between the two is required. In many cases, it may be useful to employ multibody methods for computational cost savings, kinematic exactness, and ease in handling large rotations and translations. Using finite element based models may significantly increase computational cost.

A number of approaches have been developed for simulating flexible bodies that also incorporate the effects of gross rigid body motion. Some of the widely adopted approaches include the floating frame of reference formulation (FFR) [2, 3], absolute nodal coordinate formulation (for beam and plate type elements) [4, 5], and other finite element (FE) based techniques [6, 7, 8, 9]. For a brief review and an extensive literature survey on different computational strategies in flexible multibody systems, the reader is referred to [10]. Regardless of the method, computational cost increases with increasing complexity of the system. Therefore, the development of computationally efficient methods which are also accurate has always been an important topic of research in multibody dynamics. In this paper, we present a new multibody method that incorporate interpolating spline in a divide-and-conquer framework. The present algorithm provides an efficient approach for modeling dynamic systems employing articulated rigid and flexible bodies undergoing large rotations and translations.

The divide-and-conquer algorithm (DCA) was introduced by Roy Featherstone [11] as a massively parallel, truly time optimal multibody dynamics algorithm. The DCA is applicable to general multibody systems and achieves logarithmic complexity $O(\log(N_b))$, when implemented on $O(N_b)$ processors [12]. Thus, the DCA is a good candidate for situations where computational efficiency and cost are important. Several variants of the DCA were developed for multibody systems with closed loops [12, 13], flexible bodies [1], control problems [14, 15], and discontinuous changes arising during simulation [16]. In this paper, we present a DCA based efficient algorithm that utilizes the interpolating spline functions for modeling multi-flexible-body systems. A comparison between the spline-based approach (SDCA) and the assumed modes method scheme is presented. The assumed mode DCA or flexible DCA (FDCA [1]) is appropriate when dealing with bodies with regular geometry such as straight and uniform beams whose mode shapes have been thoroughly investigated in literature [17]. However, for irregularly shaped beams, the FDCA must resort to sub-structuring in order to adequately capture the deformation of the system [18]. The SDCA approach presented in this paper overcomes this limitation.

Various types of approximating and interpolating spline functions have found applications in a number of areas, ranging from computer graphics to statistics and mechanics. For example, B-splines have been extensively studied in finite element based methods [19, 20]. This is due to the computational advantages associated with B-splines including ease of implementation and smoothness. In multibody dynamics, the use of interpolating cubic splines has been studied in [21, 22], while others have hinted towards their potential applications in modeling multi-flexible body systems [23, 24]. However, a detailed analysis of interpolating spline-based algorithms is lacking, including a study of their performance compared to traditional modeling approaches. Spline functions are appropriate in multibody dynamics because of the relatively small number of nodes in such simulations, as well as their smooth, continuous and interpolating nature at these nodes. The new spline-based algorithm developed here is restricted to modeling multibody systems comprising thin beam like bodies such as ropes, tubes, beams, and polymer chains. Furthermore, the present algorithm can be used in conjunction with other DCA-based algorithms to model systems with different types of rigid and flexible bodies. In this work, we have provided examples corresponding to SDCA for planar mechanisms and 1D bodies. However, the technique may

be extended to problems in higher spatial dimensions. The results obtained with the SDCA are compared with the method of superposed assumed modes in FDCA and the numerical complexity of the method is studied. It is demonstrated that the DCA based on interpolating splines provides an alternate and computationally fast method for modeling articulated flexible bodies.

In section 2, a brief overview of the basic DCA and interpolating splines is presented. The derivation of DCA based spline method is presented in section 3. Computational complexity of FDCA and SDCA are compared in section 3. Finally, numerical examples and discussions are presented in section 4.

2. Theoretical Background

In this section, we present a brief overview of the basic DCA and interpolating splines. The computational complexity of the FDCA and SDCA is also presented in this section.

2.1. Basic Divide and Conquer Algorithm

Detailed derivation and analysis of the performance of the DCA can be found in [11, 12]. Here, we present the DCA in its basic form. Consider two representative bodies, k and $k+1$, connected with each other by a joint J^k . Let the points where each generic body interacts with other bodies and the environment, be termed as 'handles'. As an example, consider Fig. 1(a), where H_1^k and H_2^k are the two handles on body k . Similarly, for body $k+1$, the points H_1^{k+1} and H_2^{k+1} define the position of its handles. For convenience, these handles may correspond to the locations of the joints in a body, for example, the joint J^k can serve as the location for the outward and inward handles for body k and body $k+1$, respectively. The bodies and the joints along with the constraint forces acting on the handles are shown in Fig. 1(a).

There are two main processes in the DCA, the hierarchic assembly and the hierarchic disassembly. In the pre-assembly steps, the equations of motion for each body are formed at its handles. As such, for body k the two-handle equations of motion can be written as

$$A_1^k = \zeta_{11}^k F_{1c}^k + \zeta_{12}^k F_{2c}^k + \zeta_{13}^k, \quad (1)$$

$$A_2^k = \zeta_{21}^k F_{1c}^k + \zeta_{22}^k F_{2c}^k + \zeta_{23}^k, \quad (2)$$

In the above two equations A_1^k and A_2^k are the 6×1 spatial accelerations of body k at handles H_1^k and H_2^k , respectively. The terms $\zeta_{ij}^k (i=1, 2 \text{ and } j=1, 2,)$ the inverse inertia terms associated with the two handles, whereas $\zeta_{i3}^k (i=1, 2)$ terms contain all the state dependent accelerations as well as the effects of externally applied loads [12]. The two-handle equations for body $k+1$ can be given by

$$A_1^{k+1} = \zeta_{11}^{k+1} F_{1c}^{k+1} + \zeta_{12}^{k+1} F_{2c}^{k+1} + \zeta_{13}^{k+1}, \quad (3)$$

$$A_2^{k+1} = \zeta_{21}^{k+1} F_{1c}^{k+1} + \zeta_{22}^{k+1} F_{2c}^{k+1} + \zeta_{23}^{k+1}. \quad (4)$$

The goal of the assembly process is to combine the equations for the successive bodies to form the equations of the resulting assemblies. In case of body k and body $k+1$, the resulting assembly is illustrated in Fig. 1(b), and the two-handle equations of motion for the resulting assembly

$$A_1^k = \zeta_{11}^{k:k+1} F_{1c}^k + \zeta_{12}^{k:k+1} F_{2c}^{k+1} + \zeta_{13}^{k:k+1}, \quad (5)$$

$$A_2^{k+1} = \zeta_{21}^{k:k+1} F_{1c}^k + \zeta_{22}^{k:k+1} F_{2c}^{k+1} + \zeta_{23}^{k:k+1}. \quad (6)$$

Equations 5 and 6 provide the spatial accelerations of the outward handles of the sub-assembly $k:k+1$. The terms $\zeta_{ij}^{k:k+1}$ ($i=1, 2$ and $j=1, 2, 3$) have the same meaning as before, except they now represent the sub-assembly formed by bodies k and $k+1$. Note that the two-handle equations of the sub-assembly are in the same form as the equations of the constituent bodies. This process can be repeated in a hierarchic manner for all successive bodies in the multibody tree. This assembly process starts at the individual body or leaf nodes. The two-handle equations for the pairs of adjacent bodies are combined together to form the equations for the resulting sub-assemblies. This process continues in a hierarchic fashion until the process reaches the root or the primary system node. At this point, the assembly process stops and the two-handle equations of motion for the entire system are obtained. The hierarchic disassembly begins at the primary system node, where by using the boundary conditions, the equations of motion for the last assembly are solved. Using this information, the disassembly process solves the equations of the constituent sub-assemblies. This process continues until the process reaches the individual body nodes. At the end of the disassembly process all unknowns (e.g., spatial constraint forces, modal generalized accelerations, spatial constraint impulses, spatial accelerations, jumps in the spatial velocities) for the bodies at the individual sub-domain level of the binary tree are known. The assembly and disassembly processes are illustrated in Fig. 2.

2.2. Spline Interpolation

Spline functions are smooth piecewise interpolating curves that have applications in disciplines including computer graphics, numerical methods and mechanics. Various types of splines, their functional forms and error analysis are discussed in [25, 26, 27] among others. In this paper, we focus on using quadratic and cubic interpolating splines for modeling deformation fields for elastic bodies.

2.2.1. Interpolating Quadratic and Cubic Splines—On an interval $[a, b] := [a = q_1 < q_2 < \dots < q_{n_k-1} < q_{n_k} = b]$, a spline function, $\mathcal{S}$, is a piecewise real function which coincides at each subinterval knot, q_k ($k = 1 \dots n_k$) and is continuously differentiable for quadratic splines, and twice continuously differentiable for cubic splines. Thus, quadratic and cubic splines are $n_k - 1$ second and third order polynomials, respectively, that are joined together such that their values and the values of their respective derivatives coincide at the knots. However, such functions can not be uniquely determined and additional parameters are required to be defined on the interval. Consider the case of quadratic splines, the two requirements that

1. $\mathcal{S}_j(q_j) = y_j, \mathcal{S}_j(q_{j+1}) = y_{j+1} \quad (j = 1 \dots n_k - 1)$, and,
2. $\mathcal{S}'_j(q_{j+1}) = \mathcal{S}'_{j+1}(q_{j+1}) \quad (j = 1 \dots n_k - 2)$,

only provides $3n_k - 4$ equations to find $3(n_k - 1)$ coefficients for the $n_k - 1$ quadratic polynomials. Similarly, for cubic splines, two additional parameters are required to construct the polynomials. In the context of flexible body dynamics, these additional parameters may be derived from boundary conditions.

3. Spline Based Divide and Conquer Formulation

In this section, we present the theoretical development of the spline-based divide-and-conquer algorithm. The equations of motion for a flexible body are derived in the two-handle form, as described in section 2.1. In the present work, we develop the equations of motion for beam-like structures in a multibody system. The computational complexity of SDCA is also discussed in this section.

3.1. The Two-Handle Equations of Motion in the SDCA

Consider a 1D flexible body k in a multibody tree connected by two handles (H_1^k and H_2^k) at its extreme ends, see Fig. 3. Also consider that the generalized coordinates associated with the nodal displacement for the flexible body can be given by the vector $Y^k(t) = [y_1(t) \dots y_{n_k}(t)]$. The deformed configuration of this body can be represented through a spline function with a general form given as

$$\mathcal{S}(u) := \begin{cases} \mathcal{S}_1(u, y_1(t) \dots y_{n_k}(t)) & q_1 \leq u \leq q_2 \\ \vdots & \vdots \\ \mathcal{S}_{n_k-1}(u, y_1(t) \dots y_{n_k}(t)) & q_{n_k-1} \leq u \leq q_{n_k}, \end{cases}$$

where $q_1 \dots q_{n_k}$ define the knots. The equation for the i^{th} segment, $\mathcal{S}_i$ in the spline can be defined as

$$\mathcal{S}_i := \mathcal{S}_i(u, y_1(t) \dots y_{n_k}(t)) = \left[\frac{\partial \mathcal{S}_i}{\partial y_1(t)} \dots \frac{\partial \mathcal{S}_i}{\partial y_{n_k}(t)} \right] \begin{bmatrix} y_1(t) \\ \vdots \\ y_{n_k}(t) \end{bmatrix} \equiv \vec{\varphi}_i Y^k(t). \quad (7)$$

The SDCA uses the floating frame of reference formulation. This frame may be attached to handle-1 (H_1^k) on the body with which it is attached to the ground or other bodies in the system. Since the origin of the floating frame coincides with q_1 , therefore in this case the nodal displacement $y_1(t)$ is zero. Let the spatial velocity and spatial acceleration of the body fixed reference frame be defined as

$$V_1^k := \begin{bmatrix} \vec{\omega}_1 \\ \vec{v}_1 \end{bmatrix}, A_1^k := \begin{bmatrix} \vec{\alpha}_1 \\ \vec{a}_1 \end{bmatrix}, \quad (8)$$

where $\vec{\omega}_1$ and $\vec{v}_1$ are the angular velocity and acceleration vector of the body fixed reference frame, respectively, and $\vec{\alpha}_1$ and $\vec{a}_1$ are the linear velocity and acceleration of H_1^k , respectively. The position, velocity and acceleration of a generic point P , lying on the i^{th} segment of the spline can be given as

$$\vec{r}_P = \vec{r}_1 + \overbrace{\vec{r}_{ud} + \vec{\varphi}_i Y^k(t)}^{\vec{r}_B}, \quad (9)$$

$$\vec{v}_P = \vec{v}_1 + \vec{\omega}_1 \times \vec{r}_B + \vec{\varphi}_i \dot{Y}^k(t), \quad (10)$$

$$\vec{a}_P = \vec{a}_1 + \vec{\omega}_1 \times (\vec{\omega}_1 \times \vec{r}_B) + \vec{\alpha}_1 \times \vec{r}_B + 2\vec{\omega}_1 \times \vec{\varphi}_i \dot{Y}^k(t) + \vec{\varphi}_i \ddot{Y}^k(t). \quad (11)$$

In equation 9, $\vec{r}_{ud}$ is the undeformed position of point P on the i^{th} spline segment. For example, in case of transverse deformation of a beam in plane, this can be given as

$\vec{r}_{ud} = \begin{bmatrix} u \\ 0 \end{bmatrix}$. An illustration for the transverse deflection and the position of point P in the deformed configuration is shown in Fig. 3. Using Kane's formulation [28], the equations of motion for the i^{th} segment can be written as

$$F_i^{r*} + F_i^r = 0, r = 1 \cdots n_{dof} \quad (12)$$

where n_{dof} is the number of degrees of freedom of body k and is given by $n_{dof} = n_{rigid} + n_k - 1$. The rigid degrees of freedom (n_{rigid}) are associated with the body attached frame of reference. Since $y_1(t)$ is zero, therefore the nodal degrees of freedom is one less than n_k . The terms in equation 12 are given as

$$F_i^{r*} = \int_{q_i}^{q_{i+1}} (\vec{v}_P^r \cdot \vec{a}_P) dm, \quad (13)$$

$$F_i^r = -[\vec{\omega}_P^r \cdot \vec{\tau}_P^a + \vec{v}_P^r \cdot \vec{f}_P^a + \vec{\omega}_{q_i}^r \cdot \vec{\tau}_{q_i}^c + \vec{v}_{q_i}^r \cdot \vec{f}_{q_i}^c + \vec{\omega}_{q_{i+1}}^r \cdot \vec{\tau}_{q_{i+1}}^c + \vec{v}_{q_{i+1}}^r \cdot \vec{f}_{q_{i+1}}^c]. \quad (14)$$

In the above equation, $\vec{\tau}_P^a$ and $\vec{f}_P^a$ contains all the applied torques and forces, as well as the internal forces. The constraint forces and torques acting on the segment extremities are represented by $\vec{f}_{q_i}^c$, $\vec{f}_{q_{i+1}}^c$, $\vec{\tau}_{q_i}^c$ and $\vec{\tau}_{q_{i+1}}^c$. In equations 13 and 14, $\vec{v}_Q^r$ and $\vec{\omega}_Q^r$ ($Q=1, P, 2$) are the partial velocities and can be given as

$$\vec{v}_Q^r := \frac{\partial \vec{v}_Q}{\partial g_r}, \vec{\omega}_Q^r := \frac{\partial \vec{\omega}_Q}{\partial g_r}, Q=1, P, 2, r=1 \cdots n_{dof} \quad (15)$$

where g_r ($r = 1 \cdots n_{dof}$) represents the generalized coordinates of the system. For $r = 1 \cdots n_{rigid}$, g_r is the corresponding angular or linear velocity component of handle-1. Similarly, for $r = n_{rigid} + 1 \cdots n_{dof}$, g_r is the corresponding nodal displacement (y_i , $i = 2 \cdots n_k$). Expanding equation 12 and collecting terms, the equation for the i^{th} segment can be written in the matrix form as

$$\begin{bmatrix} \Gamma_{RR} & \Gamma_{RF} \\ \Gamma_{FR} & \Gamma_{FF} \end{bmatrix}_i^k \begin{bmatrix} A_1^k \\ \ddot{Y}^k(t) \end{bmatrix} - \begin{bmatrix} \lambda_R \\ \lambda_F \end{bmatrix}_{q_i}^k \begin{bmatrix} \vec{\tau}_{q_i}^c \\ \vec{f}_{q_i}^c \end{bmatrix}_i^k - \begin{bmatrix} \lambda_R \\ \lambda_F \end{bmatrix}_{q_{i+1}}^k \begin{bmatrix} \vec{\tau}_{q_{i+1}}^c \\ \vec{f}_{q_{i+1}}^c \end{bmatrix}_i^k - \begin{bmatrix} \beta_R \\ \beta_F \end{bmatrix}_i^k = 0, \quad (16)$$

where

$$\int_{q_i}^{q_{i+1}} (\vec{v}_P^r \cdot \vec{a}_P) dm \equiv \begin{bmatrix} \Gamma_{RR} & \Gamma_{RF} \\ \Gamma_{FR} & \Gamma_{FF} \end{bmatrix}_i^k \begin{bmatrix} A_1^k \\ \ddot{Y}^k(t) \end{bmatrix}, \quad (17)$$

$$\vec{\omega}_{q_i}^r \cdot \vec{\tau}_{q_i}^c + \vec{v}_{q_i}^r \cdot \vec{f}_{q_i}^c \equiv \begin{bmatrix} \lambda_R \\ \lambda_F \end{bmatrix}_{q_i}^k \begin{bmatrix} \vec{\tau}_{q_i}^c \\ \vec{f}_{q_i}^c \end{bmatrix}_i^k, \quad (18)$$

$$\vec{\omega}_{q_{i+1}}^r \cdot \vec{\tau}_{q_{i+1}}^c + \vec{v}_{q_{i+1}}^r \cdot \vec{f}_{q_{i+1}}^c \equiv \begin{bmatrix} \lambda_R \\ \lambda_F \end{bmatrix}_{q_{i+1}}^k \begin{bmatrix} \vec{\tau}_{q_{i+1}}^c \\ \vec{f}_{q_{i+1}}^c \end{bmatrix}, \quad (19)$$

$$\vec{\omega}_P^r \cdot \vec{\tau}_P^a + \vec{v}_P^r \cdot \vec{f}_P^a \equiv \begin{bmatrix} \beta_R \\ \beta_F \end{bmatrix}_{q_{i+1}}^k. \quad (20)$$

In equation 16, subscripts R and F represent the quantities associated with the gross rigid body motion and flexible degrees of freedom, respectively. Moreover, the terms represented by are related to the mass and inertia terms, similarly the terms are the coefficients of the constraint forces and the terms represented by represent all the state dependent, applied, and internal forces acting on the segment. Also note that at the internal nodes

$$\begin{bmatrix} \lambda_R \\ \lambda_F \end{bmatrix}_{q_i}^k \begin{bmatrix} \vec{\tau}_{q_i}^c \\ \vec{f}_{q_i}^c \end{bmatrix}_{i-1}^k = - \begin{bmatrix} \lambda_R \\ \lambda_F \end{bmatrix}_{q_i}^k \begin{bmatrix} \vec{\tau}_{q_i}^c \\ \vec{f}_{q_i}^c \end{bmatrix}_i^k, \quad 1 < i < n_k. \quad (21)$$

Summing equation 16 over all spline segments gives

$$\begin{bmatrix} \Gamma_{RR} & \Gamma_{RF} \\ \Gamma_{FR} & \Gamma_{FF} \end{bmatrix}^k \begin{bmatrix} A_1^k \\ \ddot{Y}^k(t) \end{bmatrix} = \begin{bmatrix} \lambda_R \\ \lambda_F \end{bmatrix}_{q_1}^k \begin{bmatrix} \vec{\tau}_{q_1}^c \\ \vec{f}_{q_1}^c \end{bmatrix}_1^k + \begin{bmatrix} \lambda_R \\ \lambda_F \end{bmatrix}_{q_{n_k}}^k \begin{bmatrix} \vec{\tau}_{q_{n_k}}^c \\ \vec{f}_{q_{n_k}}^c \end{bmatrix}_{n_k-1}^k + \begin{bmatrix} \beta_R \\ \beta_F \end{bmatrix}^k, \quad (22)$$

where

$$\begin{bmatrix} \Gamma_{RR} & \Gamma_{RF} \\ \Gamma_{FR} & \Gamma_{FF} \end{bmatrix}^k = \sum_{i=1}^{n_k-1} \begin{bmatrix} \Gamma_{RR} & \Gamma_{RF} \\ \Gamma_{FR} & \Gamma_{FF} \end{bmatrix}_i^k, \quad (23)$$

$$\begin{bmatrix} \beta_R \\ \beta_F \end{bmatrix}^k = \sum_{i=1}^{n_k-1} \begin{bmatrix} \beta_R \\ \beta_F \end{bmatrix}_i^k. \quad (24)$$

In equation 22 the constraint force at the intermediate nodes are cancelled out leaving only the constraint spatial forces at the two handles. The spatial and nodal accelerations can be separated from equation 22 by matrix partitioning and algebraic manipulation and can be written as

$$A_1^k = \gamma_{11}^k \begin{bmatrix} \vec{\tau}_{q_1}^c \\ \vec{f}_{q_1}^c \end{bmatrix}_1^k + \gamma_{12}^k \begin{bmatrix} \vec{\tau}_{q_{n_k}}^c \\ \vec{f}_{q_{n_k}}^c \end{bmatrix}_{n_k-1}^k + \gamma_{13}^k, \quad (25)$$

$$\ddot{Y}^k(t) = (-\Gamma_{FF}^k)^{-1} (\Gamma_{FR}^k A_1^k - [\lambda_F]_{q_1}^k \begin{bmatrix} \vec{\tau}_{q_1}^c \\ \vec{f}_{q_1}^c \end{bmatrix}_1^k - [\lambda_F]_{q_{n_k}}^k \begin{bmatrix} \vec{\tau}_{q_{n_k}}^c \\ \vec{f}_{q_{n_k}}^c \end{bmatrix}_{n_k-1}^k + \beta_F^k). \quad (26)$$

In equation 25, the terms γ_{ij}^k , ($i = 1, j = 1, 3$), are the inverse inertias of body k evaluated at H_1^k , and have the same form as obtained through FDCA [1]. Equation 25 and 26 gives the spatial acceleration of handle-1 and the nodal accelerations of the flexible body, respectively. The spatial acceleration of handle-2 (H_2^k) can be given as

$$A_2^k = (T^{H1H2})^T A_1^k + \vec{\varphi}_{n_k-1} \ddot{Y}(t) \Big|_2 + \vec{\omega}_1 \times (\vec{\omega}_1 \times \vec{r}_2) + 2\vec{\omega}_1 \times \vec{\varphi}_{n_k-1} \dot{Y}(t) \Big|_2, \quad (27)$$

where

$$T^{H1H2} := \begin{bmatrix} I & \vec{r}_2 \times \\ 0 & I \end{bmatrix}. \quad (28)$$

In equation 27, $\vec{r}_2$ is the position vector from H_1^k to H_2^k , where as, $\vec{\varphi}_{n_k-1} \dot{Y}(t)|_2$ and $\vec{\varphi}_{n_k-1} \ddot{Y}(t)|_2$ are the velocity and acceleration of $(n_k - 1)^{th}$ spline evaluated at H_2^k , respectively. In equation 28, I is a 3×3 identity matrix and $\vec{r}_2 \times$ is the 3×3 skew symmetric matrix of $\vec{r}_2$. Using equations 25, 26 and 27, the spatial acceleration of joint-2 can be written as

$$A_2^k = \gamma_{21}^k \begin{bmatrix} \vec{\tau}_{q_1}^c \\ \vec{f}_{q_1} \end{bmatrix}_1^k + \gamma_{22}^k \begin{bmatrix} \vec{\tau}_{q_n}^c \\ \vec{f}_{q_n} \end{bmatrix}_{n_k-1}^k + \gamma_{23}^k. \quad (29)$$

In equation 29, the terms γ_{ij}^k , ($i=2; j=1-3$), are the inverse inertias of body k evaluated at H_2^k . The equations thus formed for body k require the computation of the inertia terms () and force terms matrices () for each segment, and then their summation according to equations 23 and 24 respectively. These matrices can be calculated independently for each segment on separate processors and do not require any matrix multiplication or inversion. Moreover, it can be noticed that equations 25 and 29 are now in the same form as the two-handle equation in the DCA and the assembly and disassembly process in the multibody tree can be done in the usual manner. At the end of the assembly and disassembly process for the multibody system the spatial accelerations and constraint forces for all bodies in the system are known and the nodal accelerations for each body can be obtained using equation 26.

3.2. Computational Complexity

In the parallel implementation of the DCA, bodies in the binary tree are mapped onto individual processors, thus yielding an overall complexity of $O(\log(N_b))$, where N_b is the total number of bodies in the system [11]. For the FDCA, with a non-constant modal mass matrix the computational complexity can be calculated as $O(\log(N_b)) + O(\max(n_k)^3)$, where $\max(n_k)$ is maximum number of modal generalized coordinates, and the cubic term is associated with the inversion cost [29]. However, in case of sub-structuring the computational complexity naturally increases. Additional calculations are performed in the FDCA to find the accelerations and constraint forces at the zero degree of freedom joints between the sub-structures. In other words, each sub-structure in the system acts as an individual body and the cost associated with it must be included in the overall cost. If the dominant number of sub-structures in a multibody tree is m , then the overall computational cost for the FDCA can be given as $O(m \log(N_b)) + O(\max(n_k)^3)$. On the other hand, the SDCA forms the equations of motion through the addition of corresponding matrices of the segments. Therefore, the computational complexity for the SDCA essentially remains as $O(\log(N_b)) + O(\max(n_k)^3)$, where $\max(n_k)$ is the dominant number of nodes for any flexible body in the system. The difference in the computation time is demonstrated in the next section by a numerical example where the FDCA uses a sub-structuring approach.

4. Numerical Examples

In order to test the formulation, two numerical examples are presented in this section. For both examples, the results are compared with a reference solution as well as the assumed

mode shapes method. The computer code for each test case is written in Matlab[®] and care is taken to maintain uniformity in programming. A fourth-order Runge-Kutta time integrator is employed for all numerical integrations.

4.1. Axial Vibration of a Fixed-Free Beam

The first numerical example is the axial vibration of a fixed-free beam. The beam is modeled using quadratic splines with three equal sub-intervals. The piecewise quadratic polynomials for the spline function are given as

$$\mathcal{S}_1 = -9 \frac{(3y_2 - 4y_3 + 2y_4)u^2}{L^2} + 6 \frac{(2y_2 - 2y_3 + y_4)u}{L}, \quad (30)$$

$$\mathcal{S}_2 = 9 \frac{(y_2 - 3y_3 + 2y_4)u^2}{L^2} - 6 \frac{(2y_2 - 5y_3 + 3y_4)u}{L} + 4y_2 - 7y_3 + 4y_4, \quad (31)$$

$$\mathcal{S}_3 = 9 \frac{(y_3 - y_4)u^2}{L^2} - 18 \frac{(y_3 - y_4)u}{L} + 9y_3 - 8y_4. \quad (32)$$

Note that the quadratic spline function satisfies the boundary conditions for a fixed-free

beam, i.e. $y_1 = 0$ and $\left. \frac{d(\mathcal{S}_3)}{du} \right|_{u=L} = 0$. The beam is initially at rest and under a compressive strain of 0.1. The length, area, density and elastic modulus of the beam are assumed to be unity in appropriate units and the model is simulated for 10 secs. The plot of the tip deflection and the absolute error compared to the analytical solution is shown in Fig. 5. It may be noticed that results conforms well to the analytical solution and the error remains small during the course of the simulation. The error and its growth may be attributed to the approximating nature of the splines and drift caused by the numerical integration of the equations of motion.

In case of sub-structuring, it is assumed that the beam is divided into three smaller components connected to each other by zero *dof* joints. For each beam, the deformation is approximated by its first natural mode shape. The resulting system is simulated using the FDCA. In both cases, i.e. with the SDCA and the FDCA, the number of flexible *dofs* is three. As expected, the simulation time for the FDCA is found to be higher than SDCA by a factor of three. This is because at each step the FDCA has to invert additional matrices to compute forces and accelerations at the joints. The response of the system and the error plots are shown in Fig. 6. Since the beam is discretized and only a truncated set of modes is used, the sub-structuring approach with the FDCA cannot accurately capture the correct beam dynamics. This behavior is also reported in the original FDCA paper [1]. Additional modes such as static correction modes or attachment modes could provide more accurate solutions, but with a higher computational cost [30, 31].

4.2. Elastic Two Arm Robot

The second numerical example is a well know problem that was first described in [32] and used by several researchers as a benchmark problem in flexible multibody dynamics [1, 24, 33, 34]. The system consists of two elastic beams connected to each other and the ground by revolute joints. The location of the joints on beam serves as the origin of the body fixed reference frame. Moreover, each beam has a point mass rigidly attached to its tip. The system, initially in static equilibrium, follows a prescribed angular motion under the

influence of gravity. Figure. 7 shows the physical parameters of the system and an illustration of the problem. The angular motion constraints for the two joints are given by

$$\phi_1(t) = \begin{cases} -\frac{\pi}{4} & t < 0 \\ +\frac{\pi}{4}(-1+72t^3) & 0 \leq t < \frac{1}{6} \\ +\frac{\pi}{4}(-18t+108t^2-144t^3) & \frac{1}{6} \leq t < \frac{1}{3} \\ +\frac{\pi}{4}(-8+54t-108t^2+72t^3) & \frac{1}{3} \leq t < \frac{1}{2} \\ +\frac{\pi}{4} & \frac{1}{2} \leq t, \end{cases} \quad (33)$$

$$\phi_2(t) = -\phi_1(t). \quad (34)$$

In this example, the cross-sections of both beams remain undeformed and a linear elastic material is considered. Furthermore, the nature of the problem requires that the geometric stiffening effect must be taken into consideration when modeling the system. The internal forces are calculated using a linear elastic model and the foreshortening effects are induced explicitly in the solution. The strain energy and internal forces for the i^{th} spline segment are given as

$$\frac{1}{2} \int_{q_i}^{q_{i+1}} EI \left(\frac{\partial^2 \vec{\phi}_i Y^k(t)}{\partial u^2} \right)^2 du, F_{elastic} = -\frac{\partial V_i}{\partial y_{i+1}} \cdot i=1 \cdots n_k - 1 \quad (35)$$

The geometric stiffening effects for this problem are calculated following the strategy defined in [35]. Accounting for foreshortening effects, the position of a generic point P , lying on the i^{th} spline segment on the body can be given as

$$\vec{r}_P = \vec{r}_1 + \vec{r}_{ud} + \vec{\varphi}_i Y^k(t) - \frac{1}{2} (Y^k(t))^T H_i(Y^k(t)) - \frac{1}{2} (Y^k(t))^T H_{i-1}(Y^k(t)) \Big|_{u=q_i}, \quad (36)$$

where

$$H_i = \int_{q_i}^u \left(\frac{\partial \vec{\phi}_i}{\partial u} \right)^T \left(\frac{\partial \vec{\phi}_i}{\partial u} \right) du. \quad (37)$$

The fourth term on the right hand side of equation 36 is due to the foreshortening in the i^{th} spline segment, whereas the fifth term is due to the foreshortening in $(i-1)^{th}$ segment, evaluated at q_i .

In the SDCA, the deformation field of the system is modeled using two cubic spline segments and the plots of the position are shown in Fig. 8. The results appear to be in agreement with the reported solutions [1, 24, 36].

Three instance in time, as shown in Fig. 7, are chosen to evaluate the accuracy of the algorithm. These time instances are important because several published results are available that utilize distinct modeling strategies [33, 34]. Furthermore, we used a commercial FE program ABAQUS™ to simulate the system. In ABAQUS, each beam is modeled using 32 beam elements with non-linear effects taken into account. The results obtained from ABAQUS and other reported formulations are given in Table. 1. It may be noticed that the results obtained from ABAQUS are quite accurate and can be used as the reference solution. In the SDCA, we used cubic splines with two, three and four segments to model each beam. The spline functions are chosen to satisfy fixed-free beam boundary conditions. The values

of the position of the tip at the three time instances are given in Table. 2. Moreover, the problem is also modeled using the FDCA with one axial and one transverse shape functions, and also using two transverse shape functions. The mode shapes are given by

$$\psi_{axial} = \left(\frac{u}{L}\right)^2, \quad (38)$$

$$\psi_{trans} = \frac{(2+i)}{2} \left(\frac{u}{L}\right)^{i+1} - \frac{i}{2} \left(\frac{u}{L}\right)^{i+2}, \quad i=1, 2. \quad (39)$$

Figure. 9(a) shows the L^2 norm of the error in the position of the tip using the SDCA and the FDCA, where the solution provided by ABAQUS is used as the reference solution. It may be observed that the error is of similar magnitude and depends on the number of spline segments or assumed mode shapes. Unlike the first example in section 4.1, no sub-structuring is used in this example for the FDCA, therefore the simulation times for both the SDCA and the FDCA are similar. Figure. 9(b) shows the plot of the trajectory of the tip using the SDCA and the zoomed in view of the tip position at its highest point. Both figures clearly demonstrate that the results obtained using the SDCA conform well to the reference solution and the error can be reduced by increasing the number of spline segments.

5. Conclusion and Future Work

In this paper, we investigated the use of quadratic and cubic splines in the DCA environment for modeling multi-flexible body systems. A theoretical background of the interpolating spline-based DCA is established and numerical examples are also presented. It is demonstrated that the SDCA is comparable to the FDCA in terms of accuracy and efficiency. Moreover, the SDCA is much faster and accurate in problems where FDCA resort to sub-structuring. This paper is limited to the application of the SDCA to planar 1D problems. We will investigate the use of the SDCA in spatial mechanisms and problems involving large deformations as part of our future work.

Acknowledgments

The authors would like to acknowledge the financial support of the National Institute for Biomedical Imaging and Bioengineering of the National Institute of Health grant # 1R01EB009362, 2R01EB005807, and 5R01EB0110037.

References

1. Mukherjee R, Anderson KS. A Logarithmic Complexity Divide-and-Conquer Algorithm for Multi-Flexible Articulated Body Systems. *Computational and Nonlinear Dynamics*. 2007; 2(1):10–21.
2. Likins PW. Modal Method for Analysis of Free Rotations of Spacecraft. *AIAA Journal*. 1967; 5(7): 1304–1308.
3. Likins PW, Baddeley V, Barbera FJ. Mathematical Modeling of Spinning Elastic Bodies for Modal Analysis. *AIAA Journal*. 1973; 11(9):1251–1258.
4. Shabana AA. Definition of the Slopes and the Finite Element Absolute Nodal Coordinate Formulation. *Multibody System Dynamics*. 1997; 1(3):339–348.
5. Berzeri M, Campanelli M, Shabana AA. Definition of the Elastic Forces in the Finite-Element Absolute Nodal Coordinate Formulation and the Floating Frame of Reference Formulation. *Multibody System Dynamics*. 2001; 5:21–54. ISSN 1384-5640.
6. Oden, J. *Finite Elements of Nonlinear Continua*. New York: McGrawHill; 1972.
7. Bathe KJ, Ramm E, Wilson EL. Finite element Formulations for Large Deformation Dynamic Analysis. *International Journal for Numerical Methods in Engineering*. 1975; 9(2):353–386.

8. Simo JC, Vu-Quoc L. On the Dynamics in Space of Rods Undergoing Large Motionsa Geometrically Exact Approach. *Computer Methods in Applied Mechanics and Engineering*. 1988; 66(2):125–161.
9. Behdinan K, Stylianou M, Tabarrok B. Co-rotational Dynamic Analysis of Flexible Beams. *Computer Methods in Applied Mechanics and Engineering*. 1998; 154(3):151–161.
10. Wasfy TM, Noor AK. Computational Strategies for Flexible Multibody Systems. *Applied Mechanics Reviews*. 2003; 56(6):553–613.
11. Featherstone R. A Divide-and-Conquer Articulated Body Algorithm for Parallel $O(\log(n))$ Calculation of Rigid Body Dynamics. Part 1: Basic Algorithm. *International Journal of Robotics Research*. 1999; 18(9):867–875.
12. Mukherjee R, Anderson KS. An Orthogonal Complement Based Divide-and-Conquer Algorithm for Constrained Multibody Systems. *Nonlinear Dynamics*. 2007; 48(1–2):199–215.
13. Khan IM, Anderson K. Performance investigation and constraint stabilization approach for the orthogonal complement-based divide-and-conquer algorithm. *Mechanism and Machine Theory*. 2013; 67(0):111–121.
14. Poursina M, Anderson K. Canonical ensemble simulation of biopolymers using a coarse-grained articulated generalized divide-and-conquer scheme. *Computer Physics Communications*. 2013; 184(3):652–660.
15. Poursina M, Anderson K. An extended divide-and-conquer algorithm for a generalized class of multibody constraints. *Multibody System Dynamics*. 2013; 29:235–254.
16. Khan IM, Poursina M, Anderson K. Model Transitions and Optimization Problem in Multi-Flexible-Body Systems: Application to Modeling Molecular Systems. *Computer Physics Communications*. 2013; 184(7):1717–1728.
17. Dietz, S.; Wallrapp, O.; Wiedemann, S. *Proceedings of Multibody Dynamics 2003*. Lisbon, Portugal: IDMEC/IST; 2003. Nodal vs. Modal Representation in Flexible Multibody System Dynamics.
18. Craig, R.; Kurdila, A. *Fundamentals of Structural Dynamics*. Hoboken, NJ: 2011.
19. Höllig K. *Finite Element Methods With B-Splines*, *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics. 2003
20. Hughes T, Cottrell J, Bazilevs Y. Isogeometric Analysis: CAD, Finite Elements, NURBS, Exact Geometry and Mesh Refinement. *Computer Methods in Applied Mechanics and Engineering*. 2005; 194(3941):4135–4195.
21. Cho K, Hori N, Angeles J. On the Controllability and Observability of Flexible Beams Under Rigid-Body Motion. *International Conference on Industrial Electronics, Control and Instrumentation, IECON '91*. 1991; vol. 1:455–460.
22. Saad M, Piedbœuf JC, Akhrif O, Saydy L. Modal Analysis of Assumed-Mode Models of a Flexible Slewing Beam. *International Journal of Modelling, Identification and Control*. 2006; 1(4): 325–337.
23. Piedbœuf JC, Moore B. On the Foreshortening Effects of a Rotating Flexible Beam Using Different Modeling Methods. *Mechanics of Structures and Machines*. 2002; 30(1):83–102.
24. Botz M, Hagedorn P. Dynamic Simulation of Multibody Systems Including Planar Elastic Beams Using Autolev. *Engineering Computations*. 1997; 14(4):456–470.
25. Greville, TNE. *Theory and Applications of Spline Functions*. Academic Press; 1969.
26. Schultz, MH. *Spline Analysis*. Englewood Cliffs, NJ: Prentice-Hall; 1973.
27. Stoer, J.; Bulirsch, R. *Introduction to Numerical Analysis*. Springer: Texts in Applied Mathematics; 2002.
28. Kane, TR.; Levinson, DA. *Dynamics: Theory and Application*. NY: Mcgraw-Hill; 1985.
29. Mukherjee, R. Ph.D. thesis. Troy, NY: Rensselaer Polytechnic Institute; 2007. Multibody Dynamics Algorithm Development and Multiscale Modelling of Materials and Biomolecular Systems.
30. Kim SS, Haug EJ. Selection of Deformation Modes for Flexible Multibody dynamics. *Mechanics of Structures and Machines*. 1990; 18(4):565–586.
31. Craig, R.; Kurdila, A. *Fundamentals of Structural Dynamics*. Wiley; 2011.

32. Sorge, K.; Ingenieure, VD. Mehrkörpersysteme mit starr-elastischen Subsystemen. Fortschritt-Berichte VDI.: Schwingungstechnik, VDI-Verlag; 1994.
33. Melzer F. Symbolic Computations in Flexible Multibody Systems. Nonlinear Dynamics. 1996; 9:147–163.
34. Claus H. A Deformation Approach to Stress Distribution in Flexible Multibody Systems. Multibody System Dynamics. 2001; 6:143–161.
35. Botz M, Hagedorn P. Multiple Body Systems with Flexible Members. Nonlinear Dynamics. 1990; 1:433–447.
36. Bhalerao, K. Ph.D. thesis. Troy, NY: Rensselaer Polytechnic Institute; 2010. On Methods for Efficient and Accurate Design and Simulation of Multibody Systems.

- We investigate interpolating spline-based divide-and-conquer algorithm (SDCA)
- The algorithm is applicable to multi-exible-body systems
- The performance of the SDCA is compared with the assumed modes-based DCA (FDCA)
- The algorithm is tested using benchmark numerical problems

Figure 1.

Assembling of two bodies to form a subassembly, a) consecutive bodies k and $k + 1$, b) a fictitious subassembly formed by coupling bodies k and $k + 1$.

Figure 2.
The hierarchic assembly-disassembly process in DCA.

Figure 3.
Position of a generic point P in a flexible beam.

Figure 4.
Cantilever beam divided in three segments.

(a) Tip displacement

(b) Percent relative error

Figure 5. Longitudinal vibration of a fixed free beam with the SDCA, a) tip deflection, and b) percent relative error.

(a) Tip displacement

(b) Percent relative error

Figure 6.

Longitudinal vibration of a fixed free beam with the FDCA, a) tip deflection, b) and percent relative error.

Figure 7. Illustration of elastic two arm robot problem, important time instances, and physical parameters of the beams.

(a) X-coordinate

(b) Y-coordinate

Figure 8.
Elastic two arm robot using two cubic spline segments.

(a) L^2 norm of error

(b) End point trajectory

Figure 9.

Plots of, a) L^2 norm of error using SDCA with multiple spline segments, and b) end point trajectory of the mechanism.

Table 1Position of the tip of the mechanism, $[x, y]$ (m).

Sorge [34]	Melzer [33]	ABAQUS
$t = 0.00$ s, [1.057, -0.411]	$t = 0.00$ s, [1.057, -0.410]	$t = 0.00$ s, [1.057, -0.411]
$t = 0.41$ s, [1.019, 0.599]	$t = 0.42$ s, [1.019, 0.592]	$t = 0.41$ s, [1.015, 0.599]
$t = 0.57$ s, [1.043, 0.123]	$t = 0.57$ s, [1.046, 0.123]	$t = 0.57$ s, [1.042, 0.122]

Table 2

Position of the tip of the mechanism using cubic splines, $[x, y]$ (m).

2 segments	3 segments	4 segments
$t = 0.00$ s, [1.057,-0.411]	$t = 0.00$ s, [1.057,-0.411]	$t = 0.00$ s, [1.057,-0.411]
$t = 0.41$ s, [1.018,0.589]	$t = 0.41$ s, [1.017,0.595]	$t = 0.41$ s, [1.017,0.596]
$t = 0.57$ s, [1.045,0.134]	$t = 0.57$ s, [1.042,0.126]	$t = 0.57$ s, [1.042,0.124]