ARTICLE IN PRESS

Available online at www.sciencedirect.com
SCIENCE DIRECT®

Comput. Methods Appl. Mech. Engrg. xxx (2005) xxx–xxx

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

A new time-finite-element implicit integration scheme for multibody system dynamics simulation

Mojtaba Oghbaei, Kurt S. Anderson *

Department of Mechanical, Aerospace, and Nuclear Engineering, Rensselaer Polytechnic Institute, Troy, New York 12180, United States

Received 22 June 2004; received in revised form 22 April 2005; accepted 22 April 2005

Abstract

When performing the dynamic simulation of stiff mechanical systems, implicit type integration schemes are usually required to preserve stability. This article presents a new implicit time integrator, which is a particular application of a novel *state-time* formulation recently developed by the authors in a more general scope. The proposed scheme is constructed with the intent of benefitting from the accuracy and apparent robustness thus far achieved with this algorithm in an integration context. This is realized by first setting up the weighted residual equations for the system in a form associated with the application of a time marching integration scheme. The resulting algebraic equations are then solved, minimizing the error of integration time step in a *generalized energy* sense, allowing one to capture the stiff behavior of solution in an efficient manner. Examples are provided to show the proposed method performance when dealing with a stiff system.

Keywords: Multibody dynamics; State-time formulation; Implicit integration scheme; Stiff mechanical systems

1. Introduction

Multibody systems (MBS) can be described as a collection of rigid and/or flexible bodies that are connected through different types of joints. Such systems can be found in various applications ranging widely from terrestrial vehicles, air and space applications to MEMS devices and biomechanical applications. Multibody dynamics, as a discipline describing the dynamic behavior of such systems, plays a major role in the design and analysis of such systems. The benefits of corresponding simulations and associated analysis are realized by increased reliability of theses systems through identification of design defects, optimized design parameters, increased innovation, and/or reduction in the overall cost and time of the development cycle.

The state of MBS changes with time under the effect of applied forces and moments. When dynamic analysis of such constrained mechanical systems is desired, a set of often differential algebraic equations (DAE) describing the system equations of motion must be set up and subsequently solved. Because time and accuracy are often both crucial in the simulation of the dynamic systems under consideration, an efficient time integration of these equations is of importance in the simulation of complex MBS. The traits of most interest in an underlying dynamic formulation and associated integration scheme are the speed, accuracy, and robustness of the numerical algorithm. A variety of formulations and dynamic simulation techniques have been developed which are sufficiently general to handle a wide range of MBS. Here we will limit our attention to the work which has most directly influenced the development of the procedure presented here.

* Corresponding author.
E-mail addresses: oghbam@rpi.edu (M. Oghbaei), anderk5@rpi.edu (K.S. Anderson).

0045-7825/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2005.04.016

The world is replete with systems where the stiffness or time scales associated with its components are sufficiently great so that the resulting equations are termed "stiff" and exhibit stiff behavior. Stiffness may be produced by the physical properties of a system or by the numerical techniques utilized to perform the simulation. For instance, MBSs that contain flexible parts, when discretized in the spatial domain, can lead to stiff modes with additional difficulties arising from time-dependence and nonlinearity. Very stiff springs, damping, or contact phenomena may also result in such behavior which is often characterized by a combination of large overall motions and often localized high frequency components in the time domain. In this case, the ratio of the maximum to minimum eigenvalues becomes large, and the effectiveness of explicit integration methods is diminished due to the step size requirements needed to maintain integrator stability [1–4].

Negrut et al. [5] presented a state-space DAE solution framework that uses an associated implicit ODE code for numerical integration of a resulting reduced set of state-space ordinary differential equations. This concept was demonstrated by using an implicit Runge–Kutta method in their methodology which has resulted in an L-stable, stiffly accurate implicit algorithm. Additionally, a computational implementation of the implicit Park method, which is a linear combination of the second and the third order BDF (backward difference formula) method, is proposed in [6] for the simulation of stiff mechanical systems.

Multirate integration is another technique for the simulation of many physical systems where the solution components can be separated into two or more groups (fast and slow subsystems) with widely varying rates of change in temporal scale. A summary of the work done in this area may be found in [7–10]. Computational savings are realized because the Nordsieck form of the Adams integrator does not have to evaluate and integrate the slower portions of the system with the same frequency of the fast portions.

The focus of most of the algorithms developed so far has been based on either the direct use of traditional numerical methods or a combination and/or modification thereof. Such methods solve for the system equations of motion in a point-wise fashion, namely solving for unknown function values at the integration step boundary, and possibly at predetermined interior points. Even though polynomial interpolation functions are often used to approximate the solution curve, the error control criteria as a rule can only be applied on the set of function approximated values on the boundaries of the integration steps. However, there are other integration schemes proposed by some researchers which are based on a weighted residual development of a time marching algorithm. Some of these efforts are cited below.

Borri and Bottasso have discussed several features of finite elements in time domain with application on dynamic systems. In [11], they presented two different weak formulations, i.e., Primal form (pure displacement formulation) and Mixed form (a two-field mixed formulation) to represent the dynamics of such systems. These formulations were based on the Hamilton's law of varying action, where the latter was a modification of the first by introducing the modified Hamiltonian function. The problem of rigid body dynamics was also considered in the context of Petrov–Galerkin mixed method formulation in [12]. In this work, the weak form for rigid body dynamics was derived using the linear and angular momentum balance which was shown to be correspondent to the well-known Hamilton's weak principle. This work was presented with the idea of introducing an alternate integration scheme for the solution of governing dynamic equations. Recently it has been shown by the same investigators that there is an intimate link existing between the finite element in time methods and the popular implicit Runge–Kutta methods, with the only difference being in the choice of unknown variables [13,14].

Hughes and Hulbert [15,16] developed a space–time finite-element method for classical elastodynamics which employs the discontinuous Galerkin method in time. Betsch and Steinmann have also developed several time-stepping finite-element schemes which possess algorithmic energy conservation for the simulation of applications such as *N*-body problem, nonlinear elastodynamics and mechanical systems with holonomic constraints [17–19]. In the first case, the time-stepping scheme is based on a Petrov–Galerkin finite-element method applied to the Hamiltonian formulation of the *N*-body problem. For the second application, the discretization process utilizes the continuous Galerkin method applied to the Hamiltonian formulation of semidiscrete nonlinear elastodynamics problem. A mixed Galerkin discretization method for index 3 DAEs pertaining to finite dimensional mechanical systems with holonomic constraints is proposed in [19]. The method leads in a natural way to time-stepping algorithms that inherit major conservation properties of the underlying constrained Hamiltonian system, namely total energy and angular momentum.

These works are mostly based on Hamilton's Principle which uses an energy-type approach in which the constraint forces interacting between adjacent bodies of a MBS will not appear explicitly in the equations of motion. This is an advantage of using indirect methods such as *Lagrangian* formulation over the other direct methods like *Newton–Euler*. However, on the negative side, they have tended to result in a dense and highly coupled system of nonlinear equations (for complex systems) which generally do not lend themselves well to time and resource-effective computation.

One of the advantages of the proposed formulation relative to the conventional approaches is its ability to effectively treat and consider multiple time scales. Like multirate integration schemes, there exists the potential of simultaneously accommodating multiple, grossly different time scales within a single simulation with this formalism. This can be accomplished by allowing different suitable temporal interval sizes for each generalized coordinate or generalized force. It should be noted that the main goal of this research [20] is to improve the state of the art in computationally efficient algorithms by circumventing the time-marching nature of the traditional simulation techniques. This would provide the means to more

fully and effectively exploit anticipated *massively parallel* computing resources. However, in the case of current article, it is shown that the associated concept *when utilized as a sequential algorithm* can function well as an efficient and general implicit integration scheme.

The remainder of this article is organized as follows. The next section provides an outline of the proposed formulation, which is followed by the characteristics of the method. In Section 4, detailed simulation results of a stiff double pendulum is given along with comparison with other traditional implicit methods. At the end, some concluding remarks are made on the use of this methodology for general multibody dynamics problems.

2. The proposed method

The new formulation is achieved by writing the weighted residual form of the full descriptor form of the governing equations of a MBS [21] and then applying the *Galerkin* approximation in time domain over the current integration step. The underlying equations to be treated via this approach are the *Newton–Euler* equations, the *Poisson's* kinematical equations [22], and the kinematic constraint equations for each body. Consider the MBS shown in Fig. 1(a) where the naming conventions for the variables associated with the free body diagram of a typical body B within this system are illustrated in part (b) of the same figure, and all are defined in the following.

It is worth mentioning here that as a result of such a temporal discretization, the system of differential equations is converted into a set of nonlinear algebraic equations. The use of the Newton–Euler formulation along with the parameterization of orientation using Poisson's kinematical equations result in a set of nonlinear algebraic equations, though of large dimension, is least coupled and at worst *quadratic* in system unknown variables.

Let us start with the weighted residual representation of the Euler's equation associated with a representative body B in the system. Euler's equation can be written in general form as

$$\sum \vec{M}^B = \sum_k \vec{r}_{kA} \times \vec{F}_{kA} + \sum_m \vec{r}_{B^* j_m} \times \vec{f}_{mC} + \sum_l \vec{T}_l = \vec{I}^{B/B^*} \cdot {}^N\vec{\alpha}^B + {}^N\vec{\omega}^B \times \vec{I}^{B/B^*} \cdot {}^N\vec{\omega}^B, \tag{1}$$

where $\sum \vec{M}^B$ represents all the moments applied on the body B, $\vec{F}_{kA}$ is the kth applied force, $\vec{r}_{kA}$ is the kth position vector from B^* (the mass center of body B) to the application point of $\vec{F}_{kA}$, $\vec{f}_{mC}$ is the mth unknown constraint force, $\vec{r}_{B^* j_m}$ is the mth position vector from B^* to application point (joint) of $\vec{f}_{mC}$, $\vec{T}_l$ is the lth applied concentrated moment, $\vec{I}^{B/B^*}$ is the inertia dyadic of body B with respect to B^*, ${}^N\vec{\alpha}^B$ is the absolute angular acceleration of mass center of B, and ${}^N\vec{\omega}^B$ is the absolute angular velocity. The above equation can be written in indicial form as

$$\varepsilon_{ijh} r_{kj} C_{ph} F_{kp} + \varepsilon_{ijs} r_{mj} C_{ns} f_{mn} + C_{qi} T_q = I_{ij} \alpha_j + \varepsilon_{ijt} \omega_j I_{tl} \omega_l, \tag{2}$$

where ε_{ijh} is the cyclic permutation operator, C_{ph} are the elements of the direction cosine matrix, ${}^N\underline{C}^B$ from the local frame B to the Newtonian reference frame N, and the other symbols are the scalar components of the vector quantities described earlier. All indices except k and m vary from 1 to 3, while these two will depend on the number of applied and constraint forces acting on the body. Performing a weighted residual approach on the above ordinary differential equation results in the following algebraic equation:

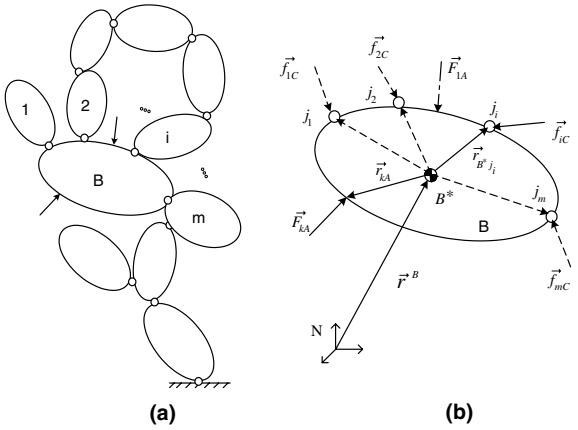

Fig. 1. Schematic of a multibody system.

$$\overline{C}_{ph_x}\cdot\left\{\varepsilon_{ijh}r_{kj}F_{kp}\int_0^1\psi_x(t)\cdot\psi_r(t)\,\mathrm{d}t\right\}+\overline{C}_{qi_y}\cdot\left\{T_q\int_0^1\psi_y(t)\cdot\psi_r(t)\,\mathrm{d}t\right\}$$
$$+\overline{C}_{ns_w}\cdot\overline{f}_{mnt}\cdot\left\{\varepsilon_{ijs}r_{mj}\int_0^1\psi_w(t)\cdot\varphi_t(t)\cdot\psi_r(t)\,\mathrm{d}t\right\}$$
$$-I_{ij}\left\{\left[\overline{q}_{ju}\dot{\psi}_u(1)\cdot\psi_r(1)-\omega_j(0)\cdot\psi_r(0)\right]-\overline{q}_{ju}\int_0^1\dot{\psi}_u(t)\cdot\dot{\psi}_r(t)\,\mathrm{d}t\right\}$$
$$-\overline{q}_{ju}\cdot\overline{q}_{lv}\cdot\left\{\varepsilon_{ijt}I_{tl}\int_0^1\dot{\psi}_u(t)\cdot\dot{\psi}_v(t)\cdot\psi_r(t)\,\mathrm{d}t\right\}=0 \tag{3}$$

in which the barred quantities represent the unknown nodal variables to be solved for, and $\psi(t)$ and $\varphi(t)$ denote the temporal trial functions associated with the system state and force variables, respectively. The order of these functions are in general different so that the *Babuška–Brezzi condition* [23] is satisfied. Variation of all the new sub indices in the above equation such as x, y, u and so forth depend on the order of trial functions.

Note that the equality $\alpha_i=\dot{\omega}_i=\ddot{q}_i$ is used where q exists mathematically, but not necessarily physically. Also, Eq. (3) is expressed for a unit size temporal element (integration step) where in case of using *Lagrange* family of interpolating functions, it constitutes $3k$ quadratic algebraic equations and $9k+3m(p+1)+3k$ unknown variables, with k and p being the order of the trial functions associated with spatial and force variables respectively, and m is the number of geometric constraints.

The next equation is the *Newton's* second law of motion that is given below

$$\vec{F}^B+\sum_m\vec{f}_{mC}=m_B\ddot{\vec{r}}^B \tag{4}$$

in which $\vec{F}^B=\sum_k\vec{F}_{kA}$, m_B is the mass of body B, and $\vec{r}^B$ denotes the absolute displacement vector of the mass center in the inertial reference frame. Applying the similar approach as described above on Eq. (4) results in the following algebraic equation where for simplicity the sub or superscript B is eliminated.

$$\left\{\overline{r}_{ij}\dot{\psi}_j(1)\cdot\psi_r(1)-\dot{r}_i(0)\cdot\psi_r(0)\right\}-\overline{r}_{ij}\int_0^1\dot{\psi}_j(t)\cdot\dot{\psi}_r(t)\,\mathrm{d}t$$
$$-\frac{1}{m}\left\{F_i\int_0^1\psi_r(t)\,\mathrm{d}t+\sum_m\left[\overline{f}_{mk}\cdot\int_0^1\varphi_k(t)\cdot\psi_k(t)\,\mathrm{d}t\right]\right\}=0\quad i=1,2,3. \tag{5}$$

Eq. (5) constitutes $3k$ linear algebraic equations and introduces $3k$ additional unknowns per temporal element. All symbols and indices reserve their meaning from before.

As the full descriptor formulation is used in the proposed scheme, another set of equation is needed to take the large rotational motion of MBS into consideration. There are several parameterizations available for describing the reference frame orientations, such as the orientation angles (of which *Euler's angles* are a subset), *Euler's parameters*, and the direct use of *direction cosines* as rotation parameters. For the proposed methodology, it is realized that the last form of parameterization results in the set of algebraic equations that are in the most desirable form. If the direction cosines are used directly as state variables, then the kinematical differential equations which relate the time derivative of the transformation matrix to the angular velocity of the body are the *Poisson's* kinematic equations given by

$${}^N\dot{\underline{C}}^B={}^N\underline{C}^B\cdot\left[{}^N\underline{\omega}^B_{\times}\right]_B, \tag{6}$$

where $\left[{}^N\underline{\omega}^B_{\times}\right]_B$ is the angular velocity cross-product matrix corresponding to the local body B reference frame and associated body-fixed basis vectors. This equation can be written in indicial form as

$$\dot{C}_{il}=-\varepsilon_{mlk}\omega_kC_{im} \tag{7}$$

in which, all symbols retain their definition from before. Eq. (8) shows the weighted residual representation of the Poisson's equations over the current integration step which constitutes $9k$ quadratic algebraic equations and there are no additional unknown variables introduced by them.

$$\overline{C}_{il_n}\cdot\int_0^1\dot{\psi}_n(t)\psi_r(t)\,\mathrm{d}t+\overline{C}_{im_j}\cdot\left\{\varepsilon_{mlk}\overline{q}_{kp}\int_0^1\dot{\psi}_p(t)\cdot\psi_j(t)\cdot\psi_r(t)\,\mathrm{d}t\right\}=0. \tag{8}$$

Note that in this equation each element of the direction cosine matrix is treated as an independent variable and approximated in much the same manner as the other state variables. Also, in all the equations derived so far (i.e., Eqs. (3), (5), and (8)), the weighting functions are considered the same as the trial functions associated with the system state variables.

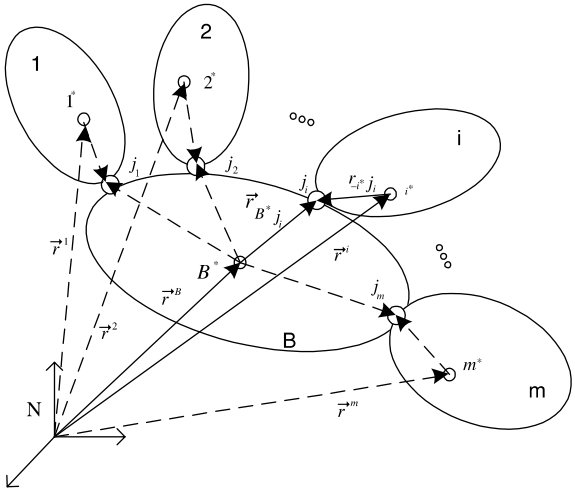

Fig. 2. Kinematic constraint relationships for body B and its neighbors.

The last set of equations to be considered are the kinematic constraint equations. Consider again the body B with m neighboring bodies as illustrated in Fig. 2. For each joint J_i, the velocity level of kinematic constraint equations can be written as

$$\dot{\vec{r}}^{B} + {}^{N}\dot{\underline{C}}^{B}\,\underline{r}_{B^*j_i} = \dot{\vec{r}}^{j} + {}^{N}\dot{\underline{C}}^{i}\,\underline{r}_{i^*j_i} \quad i = 1, 2, \ldots, m \tag{9}$$

with their weighted residual representation given in Eq. (10) that form a linear set of equations.

$$\begin{aligned}&\overline{r}^{B}_{tj}\int_0^1 \dot{\psi}_j(t)\cdot\varphi_z(t)\,\mathrm{d}t + {}^{N}\overline{C}^{B}_{tkm}\int_0^1 r_{B^*j_ik}\dot{\psi}_m(t)\cdot\varphi_z(t)\,\mathrm{d}t - \overline{r}^{i}_{ts}\int_0^1 \dot{\psi}_s(t)\cdot\varphi_z(t)\,\mathrm{d}t\\ &\quad + {}^{N}\overline{C}^{i}_{tln}\int_0^1 r_{i^*j_il}\dot{\psi}_n(t)\cdot\varphi_z(t)\,\mathrm{d}t = 0 \quad i = 1, 2, \ldots, m.\end{aligned} \tag{10}$$

This results in an additional $3m(p+1)$ equations per current temporal element that if considered along with all the other bodies' discretized equations will form a consistent number of equations and unknowns. As indicated earlier, the combination of all discretized equations exhibits only quadratic nonlinearity in the nodal variables with a significant linear part as reiterated in compact form in Eq. (11).

$$\begin{aligned}&\underline{A}_1\overline{q}\cdot\overline{q} + \underline{B}_1\overline{C}\cdot\overline{f} + \underline{D}_1\overline{q} + \underline{E}_1\overline{C} = \underline{R}_1,\\ &\underline{A}_2\overline{r} + \underline{B}_2\overline{f} = \underline{R}_2,\\ &\underline{A}_3\overline{C} + \underline{B}_3\overline{C}\cdot\overline{q} = \underline{R}_3,\\ &\underline{A}_4\overline{r} + \underline{B}_4\overline{C} = \underline{R}_4,\end{aligned} \tag{11}$$

which are sorted based on Eqs. (3), (5), (8) and (10), respectively. The coefficient matrices as well as the right-hand residuals $\underline{R}_i$ are known as soon as the system parameters, initial states, and initial guess are provided with these equations. Note that the underlined barred quantities are the matrix representation of each set of unknown nodal variables described earlier. This matrix of unknown variables may then be solved for by an appropriate nonlinear system solution scheme.

3. Characteristics of the method

When solving a matrix of initial value problems, one common approach is to convert the governing equations into the state-space form as

$$\underline{y}' = \underline{f}(t, \underline{y}) \quad \underline{y}(t_0) = \underline{y}_0 \tag{12}$$

in which depending on whether the unknown function values at the terminal boundary of integration step are used, the method is called explicit or implicit. Implicit methods tend to result in a set of nonlinear algebraic equations that need a Newton type method to solve for the unknown variables. In this sense, the proposed method is regarded as an implicit type integrator since at each integration step the evaluation requires the use of system state information at the terminal boundary of the temporal element. To achieve this a Newton-type method is applied on Eq. (11) to solve for unknown nodal variables.

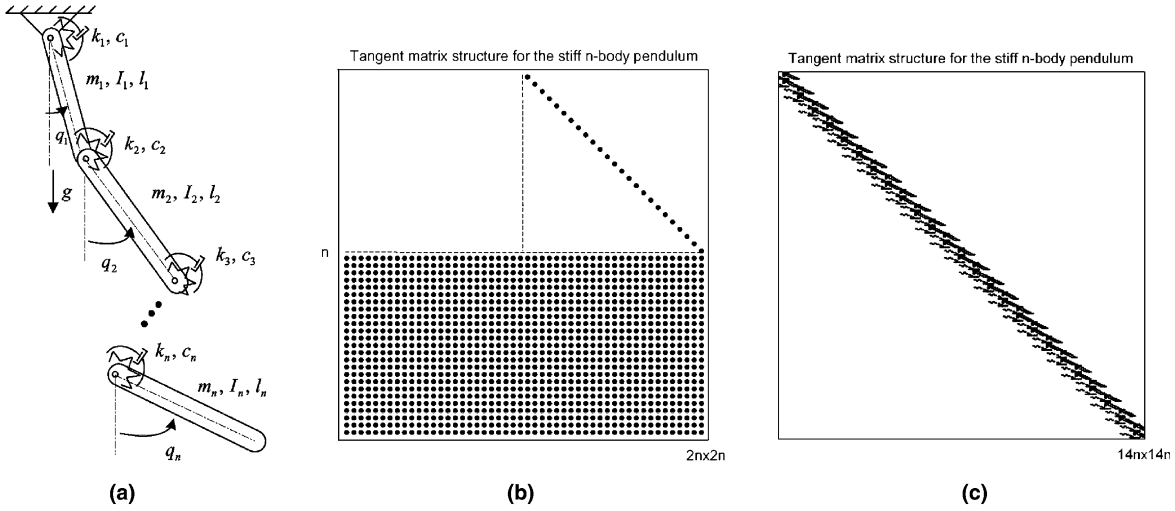

Fig. 3. Tangent matrix structure for the stiff n-body pendulum. (a) The stiff n-body pendulum, (b) the traditional state-space method and (c) the proposed method.

If the standard quadratic and linear Lagrange shape functions are used for spatial variables and constraint forces, respectively then there are two unknown barred quantities per each spatial variable namely the middle and the final nodes and two unknown barred quantities per each constraint force that are the initial and final nodes of the integration step. Once the Newton iteration converges at the current step, the boundary nodes will be updated and used as initial values for the next integration step and the simulation marches forward temporally.

Due to the sparse structure and minimum degree of nonlinearity of the equations resulting from this formulation, the method should perform well relative to traditional methods when applied to systems possessing a large number of generalized coordinates and or kinematic constraints. Specifically, more traditional state-space type formulations yield systems of substantially smaller dimension, but the associate equations are highly, often fully coupled, and extremely nonlinear. In such cases, the associated tangent matrix is heavily populated, expensive to produce, and even more expensive to decompose. Also, given the great level of nonlinearity associated with these approaches when applied to large complex systems, the number of iteration required for convergence of the method at each time step may be large. By comparison, the proposed formulation, though resulting in a larger set of equations, is in the least coupling form and produces the minimum degree of nonlinearity. Generation of the associated tangent matrix is trivial and corresponding decomposition and solution of the resulting system of equations at each iteration is relatively inexpensive.

For instance, let us consider a stiff n-body serial pendulum problem as shown in Fig. 3(a), where stiffness is introduced through the rotational springs and dampers mounted on the connecting joints. As the problem involves stiff modes, an implicit integrator is required for a stable numerical integration and as such it requires solving a system of nonlinear algebraic equations. Fig. 3 demonstrates a comparison between the structure of the tangent matrix associated with the proposed formulation using linear-quadratic Lagrange element and a traditional state-space type formulation for this problem. The latter approach will produce a smaller set of equations while being highly nonlinear (nth order transcendental nonlinearity) and heavily coupled. Hence, the lower part of the tangent matrix will be fully populated after linearization as shown in Fig. 3(b). Thus, for relatively large values of n the cost of decomposing and solving such system will be in the order of $\mathcal{O}(n^3)$ for each iteration, while using the proposed method will result in $\mathcal{O}((bandwidth)^2 \cdot n)$ operations that is much smaller than the other. Moreover, as the number of bodies are increased, there will be additional diagonal blocks introduced to the sparse block diagonal structure of the system tangent matrix and the bandwidth remains constant.

Given the at-worst quadratic nonlinearity of the resulting equations in this method, convergence has been thus far shown to be rapid. Additionally, there has been a great deal of literature presenting efficient (and parallelizable) solution methods for the system of large scale linear and quadratic equations which can also be exploited within this formulation [24,25]. Hence, the proposed method can potentially serve as an efficient and robust time integration scheme particularly when dealing with large complex multibody systems.

3.1. Stability analysis

Ideally, one would desire that a general argument be made on the properties of a numerical method employed for the solution of a class of differential equations. This will be too complicated in general, as for instance in the case of nonlinear problems the behavior of the system is dependent on the particular solution trajectory considered (i.e., the initial state). As

such, usually the characteristics of a method is tested to capture the essential properties of a class of differential equations, termed as the test equation. Stability analysis is performed by checking the variation of the spectral radius ρ of the amplification or transition matrix (the matrix A in Eq. (13)) derived for the test equation. The spectral radius of this matrix, which maps the initial (known) state of the system into the unknown state vector, is the largest absolute value of its eigenvalues. To ensure unconditional stability, this quantity must not exceed unity. Indeed, this is taken from the following fact. The absolute stability requirement suggests that since for all stable test equations $|y(t_n)| \leqslant |y(t_{n-1})|$, an ($A$-stable) discretization method should behave in the same fashion, as well.

$$\underline{y}_n = A\underline{y}_{n-1} \quad \rho = \max|\lambda_i|. \tag{13}$$

Let us consider the matrix form of the test equation, i.e., $\underline{y}' = A_{2\times 2}\underline{y}$ which is the state-space representation of a second order ODE. To check for the stability behavior of the proposed formulation, we apply the suggested discretization scheme on the following standard linear 1DOF oscillator (spring–mass–damper) and analyze the variation of its spectral radius.

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 x = f(t). \tag{14}$$

Utilizing the family of quadratic Lagrange shape functions on an integration step $[t_{n-1}, t_n]$, with h being the length of the domain $h = t_n - t_{n-1}$, results in a set of two linear equations and two unknowns. These variables are $\bar{x}_2$ and $\bar{x}_3$ which denote the unknown nodal variables associated with the middle node $t = \frac{t_{n-1}+t_n}{2}$, and boundary node $t = t_n$. It should be noted that in general once these nodal variables are known, they are substituted in the approximate relations and some post-processing calculations provide the velocity information throughout the domain yielding the initial condition for the next increment. Manipulating these equations and writing them in matrix form result in

$$\begin{bmatrix} f_{11}(h,\zeta,\omega_n) & f_{12}(h,\zeta,\omega_n) \\ f_{21}(h,\zeta,\omega_n) & f_{22}(h,\zeta,\omega_n) \end{bmatrix} \begin{Bmatrix} \bar{x}_2 \\ \bar{x}_3 \end{Bmatrix} = \begin{Bmatrix} x_0 \\ v_0 \end{Bmatrix} \tag{15}$$

$f(t)$ is eliminated as it does not affect the analysis. The inverse of the coefficient matrix in Eq. (15) forms the amplification matrix for the proposed method which obviously is a function of h, ζ and ω_n.

A simple script was written to check the variation of the absolute values of the eigenvalues of the amplification matrix. This was made possible by changing each parameter present in the matrix A, namely, h, ζ, ω_n for quite a large range. It was realized that one of the absolute eigenvalues of this matrix, $|\lambda_1|$ always lies between 0 and 0.75 and the other, $|\lambda_2|$ is bounded by $\sim$0.25. Therefore, the spectral radius of this system remains always smaller than unity for all combinations of mass, spring and damper coefficients and the interval size of h. To reassure about these results, the two equations $|\lambda_i| = a$ $i = 1,2$ were also solved in Maple© for h given a large range of variation on ζ and ω_n, and for different values of $0 \leqslant a \leqslant 1$. There was no solution found for h if a was chosen greater than the above values (0.75 and 0.25, respectively) for each of the cases $i = 1,2$, which confirms the results mentioned above.

As an example, an underdamped system with the physical parameters $m = 1$, $c = 0.4$, $k = 0.16$ (all units are SI) was considered which produces the natural frequency of $\omega_n = \sqrt{k/m} = 0.4$ and the damping coefficient $\zeta = c/c_{cr} = c/2\sqrt{mk} = 0.5$. The eigenvalues of the amplification matrix (using 3-noded Lagrange element) were plotted as a function of h which can be seen in Fig. 4. In this plot, the upper curve (plotted as solid line) determines the variations of the spectral radius.

Similar analysis was repeated for the case of cubic and quartic families of Lagrange shape functions, where it was noticed that as the order of shape functions is increased, the upper bound of spectral radii decreases. For the above

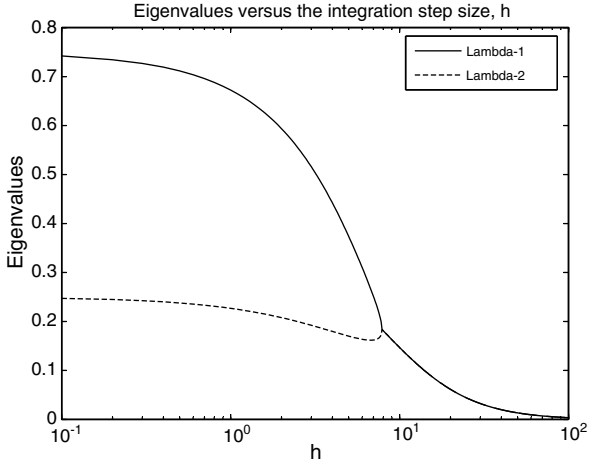

Fig. 4. Eigenvalues of the amplification matrix (quadratic Lagrange element).

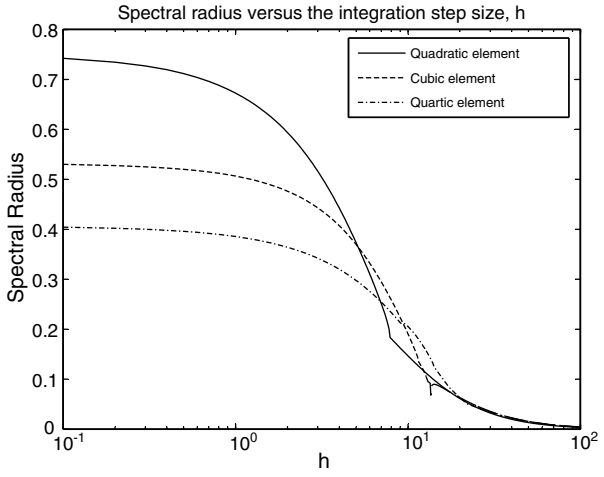

Fig. 5. Spectral radii of the amplification matrices.

mentioned example, the variation of spectral radii for all these cases, i.e., using quadratic, cubic and quartic shape functions are shown in Fig. 5.

As a final note, the stability analysis of a simple nonlinear system was also considered for a number of initial conditions and initial guesses. Nonlinearity was introduced to the above 1DOF oscillator by adding a cubic stiffness term k_2x^3 which converts the linear system into the well-known Duffing nonlinear oscillator

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 x + \omega_{\mathrm{NL}}^2 x^3 = f(t), \tag{16}$$

where $\omega_{\mathrm{NL}} = \sqrt{k_2/m}$. Discretizing equation (16) using the proposed method and via the 3-noded Lagrange element results in a nonlinear algebraic equation which after linearization about the current iteration point takes the form

$$\begin{bmatrix} g_{11}(h,\zeta,\omega_n,x_0,v_0,\bar{x}_2^{i-1},\bar{x}_3^{i-1}) & g_{12}(h,\zeta,\omega_n,x_0,v_0,\bar{x}_2^{i-1},\bar{x}_3^{i-1}) \\ g_{21}(h,\zeta,\omega_n,x_0,v_0,\bar{x}_2^{i-1},\bar{x}_3^{i-1}) & g_{22}(h,\zeta,\omega_n,x_0,v_0,\bar{x}_2^{i-1},\bar{x}_3^{i-1}) \end{bmatrix} \begin{Bmatrix} \delta\bar{x}_2^i \\ \delta\bar{x}_3^i \end{Bmatrix} = \{\underline{\text{Residual}}\}. \tag{17}$$

In this equation, x_0 and v_0 are the initial state, $\bar{x}_2^{i-1}$ and $\bar{x}_3^{i-1}$ are the updated initial guess from the $(i-1)$th iteration, and $\delta\bar{x}_2^i$ and $\delta\bar{x}_3^i$ are the updated differential change in unknown nodal variables at the current ith iteration. Thus, as indicated above the behavior of the system is dependent on the particular solution trajectory considered. Similar analysis was carried out for this system where by altering the values of h, ζ, ω_n and w_{NL} the range of variation of the amplification matrix eigenvalues were examined. In this case, it was also noticed that the spectral radius of the amplification matrix and associated with the quadratic Lagrange element is bounded by 0.75. This result was verified by altering the values of initial states and guesses in several cases.

In the following section, an example is presented to indicate how the proposed method can be used as an implicit type integration scheme for a stiff dynamic system simulation.

4. Example: Stiff double pendulum

As a simple nonlinear example, consider the double pendulum depicted in Fig. 6. This system was selected because it is easy to understand, demonstrates the accuracy and other characteristics of the method, and is an accepted test case available in the literature [5]. Being that as it may, this problem will not perform well in assessing simulation speed and computational cost relative to state-space methods because as pointed out in Section 3, these methods will in this instance yield a heavily populated tangent matrix of dimension 4×4, whereas the formulation presented here will produce a banded matrix of dimension 28×28. Dynamic simulation results for this system are given based on different families of shape functions that are selected to interpolate the time history of state variables and constraint forces for the simulation period. Through simulation results, it is shown how this scheme can capture stiff behavior in the solution, with discussion provided on the characteristics and selection of the preferred family of shape functions.

4.1. System parameters

Stiffness is induced by properly selecting the constants of rotational spring/damper actuator acting on the joints. The same system parameters as given in [5] are chosen here which are: $m_1 = 3$ and $m_2 = 0.3$ masses of pendulums, $l_1 = 2$

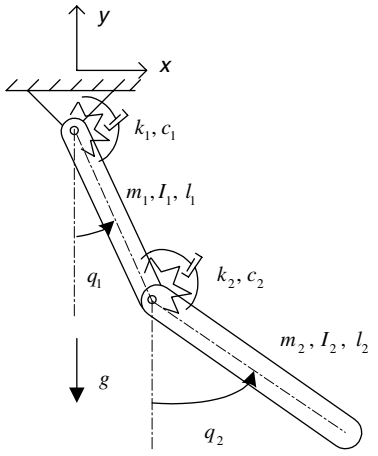

Fig. 6. Stiff double pendulum.

and $l_2 = 3$ lengths of pendulums, $k_1 = 400$, $k_2 = 3 \times 10^5$ and $c_1 = 15$, $c_2 = 5 \times 10^4$ rotational springs and dampers coefficients, respectively. All units are in SI. The zero tension angles for the two springs are $q_1^0 = 0$ and $q_2^0 = \pi/2$ and the initial conditions are $q_{1_0} = \pi/2, q_{2_0} = 5\pi/12$ and $\dot{q}_{1_0} = 0$, $\dot{q}_{2_0} = 10$. What makes this 2DOF system stiff is the dominant eigenvalue at the initial state which has a very small real part of the order -10^5.

Plots presented over the next pages provide representative results associated with the 2 s simulation of this pendulum via the proposed formulation (hereafter referred to as ST), the BDF method, as well as the MATLAB© ODE15s solver that is specifically written for the temporal integration of stiff systems. ODE15s is a variable order and a variable step size solver based on the numerical differentiation formulas (NDFs). Optionally, the backward differentiation formulas (BDFs, also known as Gear's method) may also be utilized. It can use each of these formulas with a maximum order of accuracy of 5.

4.2. Lagrange family of shape functions

As indicated previously, the order of the polynomial interpolating functions associated with spatial variables and constraint forces can and should be different from each other to satisfy the Babuška–Brezzi condition. This in many cases requires the use of lower order interpolation functions for the force-type variables than for the spatial variables. The plots, which are presented in Fig. 7 show the time history of the angle q_1 and angular velocity ω_1 using linear quadratic (LQ) (i.e. linear interpolation functions are used for the constraint force variables while quadratic interpolation functions for the spatial variables) and quadratic–cubic (QC) Lagrange shape functions and 75 integration steps. Note that there is a small subfigure in each of the plots associated with angular velocity via the methods used throughout this article. This is to magnify the region showing that there is an initial stiff behavior in the angular velocity as well as to indicate how well each method is able to capture it. The plots correspondent to each of the ST and ODE15s results are superimposed onto the same set of axes for better comparison.

In these plots, the ST solutions are obtained using 75 equally spaced intervals in the simulation period, while the results from Matlab ODE15s are generated using a variable step-size scheme. This is due to the fact that if this equal step-size interval is prescribed to the ODE15s integrator, then it will just pass over the stiff behavior without capturing the initial rapid change in the angular velocity. Additionally, the proposed method was also tested with a variable step-size scheme which demonstrated that it did capture the initial stiff part of the curve as well as maintaining the overall macro-behavior of the solution curve.

Comparison of the plots given in Fig. 7 shows graphically how the solution accuracy is improved through a p-type refinement both in position and velocity level. Simulations are also run using different number of integration steps to provide a measure/indicator of how error improves for both the LQ and QC cases. Fig. 8 illustrates the error analysis of the two above classes of shape functions used in the simulation of this problem. In this study, the same error assessment method as presented in [5], i.e. the maximum error $\Delta^{(n)}$ and the average trajectory error $\overline{\Delta}^{(n)}$ are adopted, specifically

$$\Delta^{(n)} = \max_{1 \leqslant i \leqslant n} \Delta_i \quad \overline{\Delta}^{(n)} = \frac{1}{n} \|\Delta_i\|_2. \tag{18}$$

The error Δ_i at time t_i is defined as $\Delta_i = |E_i - e_i|$, where E_i is the reference solution and e_i is the simulation results obtained using selected number of integration steps, n, both at time t_i. The grid points of the simulation interval are denoted by $t_{\text{init}} = t_1 < t_2 < \cdots < t_n = t_{\text{end}}$. So, the reference solution is obtained first by applying a tight tolerance on the Matlab

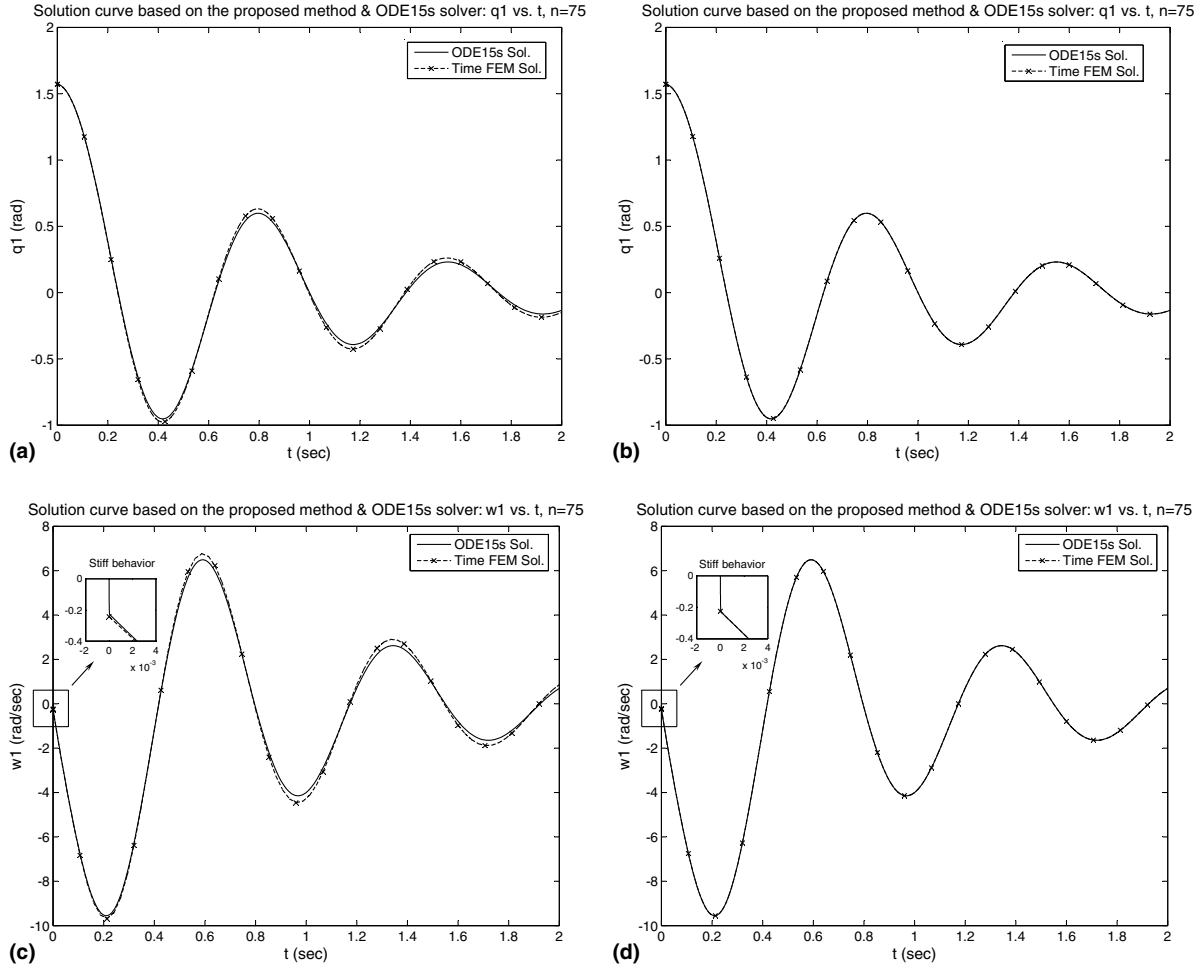

Fig. 7. Simulation results of the stiff double pendulum, $n = 75$. (a) Angle q_1 vs. t LQ interpolation, (b) angle q_1 vs. t QC interpolation, (c) angular velocity w_1 vs. t LQ interpolation and (d) angular velocity, w_1 vs. t QC interpolation.

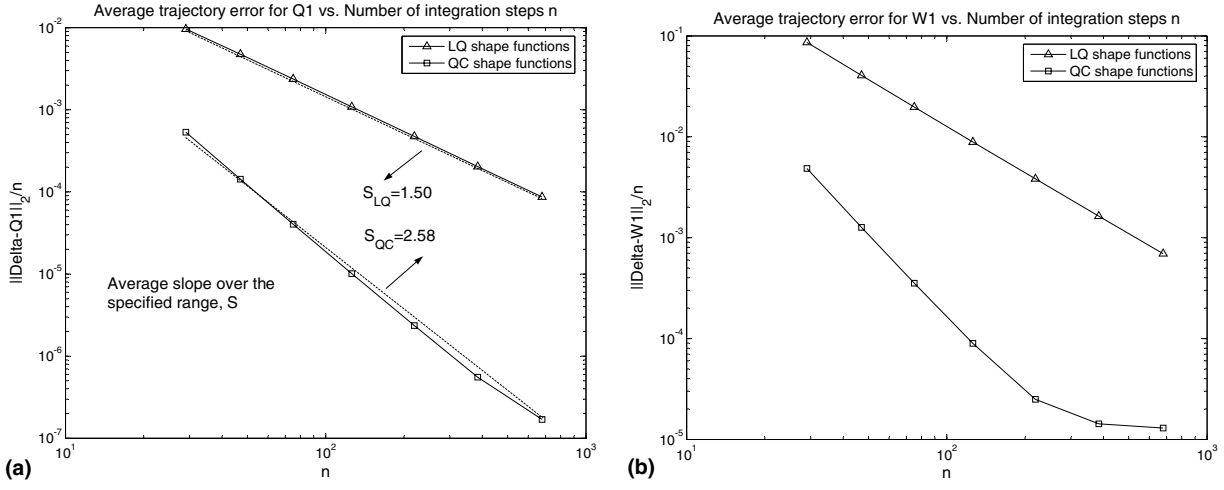

Fig. 8. Average trajectory error in q_1 and w_1 for both LQ and QC cases. (a) Average trajectory error in the angle q_1 and (b) average trajectory error in the angular velocity w_1.

ODE15s solver and then all the results are compared against them to find the maximum grid-point error and the average trajectory error per simulation period. The absolute and relative tolerances for the reference solution is set to 10^{-8}.

These results are also tabulated in Table 1 for more convenient reference. Inspection of the numerics presented here indicates that the method exhibits a super-linear performance of 1.50 (that is the error decreases as $(1/n)^{1.50}$) when using the

Table 1
Average trajectory error as a function of number of integration steps

# of integration steps (n)	29	47	75	126	219	384	678
$\overline{\Delta}_{q_1}^{(n)}$ (LQ)	0.0096	0.0048	0.0024	0.0011	4.74e−4	2.04e−4	8.68e−5
$\overline{\Delta}_{q_1}^{(n)}$ (QC)	5.35e−4	1.43e−4	4.04e−5	1.01e−5	2.35e−6	5.53e−7	1.70e−7
$\overline{\Delta}_{\omega_1}^{(n)}$ (LQ)	0.0862	0.0406	0.0197	0.0089	0.0039	0.0016	6.94e−4
$\overline{\Delta}_{\omega_1}^{(n)}$ (QC)	0.0048	0.0013	3.54e−4	8.96e−5	2.50e−5	1.43e−5	1.30e−5

linear-quadratic interpolation set, and super-quadratic performance of 2.58 for the quadratic–cubic interpolation set. The accuracy of the approach thus appears to be influenced by the lower order of the force-associated interpolating functions.

4.3. Combination of Lagrange–Hermite family of shape functions

Hermite family of interpolating functions along with the Lagrange shape functions are also used for the parametrization of equations. In Hermite interpolation both function values and its derivatives of various order can be interpolated. Since the approximation is smooth in the element interior, interpolation of derivatives is considered for the nodes on the boundaries of elements to ensure that the derivative of the interpolant is continuous across the interface. Since both position and velocity level of the system state variables are present in the ST formulation, Hermite interpolating function would seem to be a proper choice for continuous approximation of velocity type variables.

The idea here is to interpolate the spatial variables using Hermite shape functions and the constraint forces using Lagrange interpolation. Note that the former parametrization cannot be used for constraint forces since there is no information available on their derivative data. It is realized that to avoid the spurious modes in the system, a Lagrange shape function of at least two order in degree less than the Hermite shape function should be used, otherwise the system tangent matrix becomes singular. Fig. 9 demonstrates the same simulation period for the stiff double pendulum using the combination of the Lagrange and Hermite shape functions. In an effort to conserve space, only two representative results are given where the left plot is using the linear Lagrange–cubic Hermite element and the right plot is using quadratic Lagrange–quartic Hermite element.

It is observed that the velocity level information of system state variables is not interpolated so well as the similar case where only the Lagrange shape functions were used. This is due to the fact that the system of equations are now more constrained and these constraints are not only on the system state variables but on their derivatives as well. Indeed, in this case at the end of each integration step, both the state variables and their derivatives are dictated to the system and considered as *unchangeable* initial conditions for the next integration step. So, the Newton iterate does its best to minimize the error over the current integration step in an average sense and this is what is exactly observed from the results if any portion of the solution is zoomed in. However, since the initial values are not allowed to change (due to the use of Hermite shape function on spatial variables), the system has to balance the unknown nodal variables at the terminal boundary of each integration step in order to minimize the energy error norm. Thus, any offset in the initial position and velocity (that is due to the error from the previous step) gets boosted in the current step and this accumulates as the simulation moves forward in time.

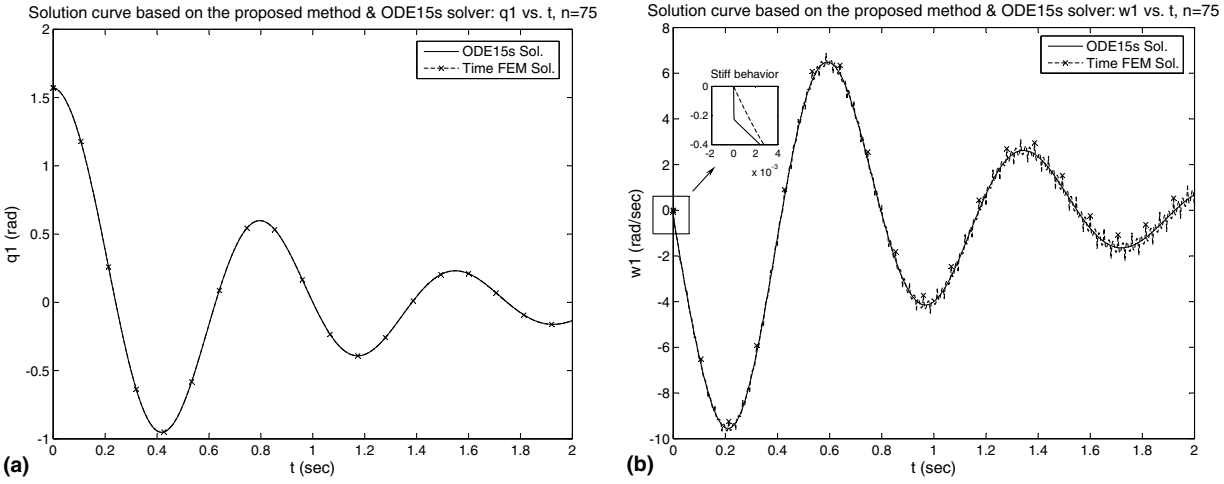

Fig. 9. Simulation results of the stiff double pendulum, $n = 75$. (a) angle q_1 vs. t L1H3 interpolation and (b) angular velocity, w_1 vs. t L2H4 interpolation.

It is concluded through some test cases that overall the use of Lagrange interpolation is a better choice than the combination of Lagrange–Hermite shape functions. In the subsequent sections, comparisons are made between the results from Lagrange family of shape functions and the results presented in [5], as well as the results from a second order traditional BDF method.

4.4. Comparison of results

An implicit Runge–Kutta method for dynamic simulation of MBS is presented in [5]. The article presents a DAE reduction process into the state-space ordinary differential equations (SSODE) through the generalized coordinate partitioning method [26]. In Table II of this article [5], the results of error analysis for the 2 s simulation of the problem at hand are tabulated and in Table III a comparison between the number of integration steps required to fulfill some tolerances is given. As shown above, the same number of integration steps are also taken for this problem using the described formulation. Fig. 10 shows a comparison between the results from ST QC-Lagrange shape functions and the results given in Table II of the above article.

As one would expect, the results in Fig. 10 clearly demonstrate that once the integration step size has decreased beyond a certain level, the SSODE solver offers better accuracy given that it is a fourth order accurate method, whereas the QC-Lagrange element shows a super-quadratic order of accuracy. In Fig. 11, the simulation results using the second order BDF method are presented to provide a better comparison base with the results from the ST simulation.

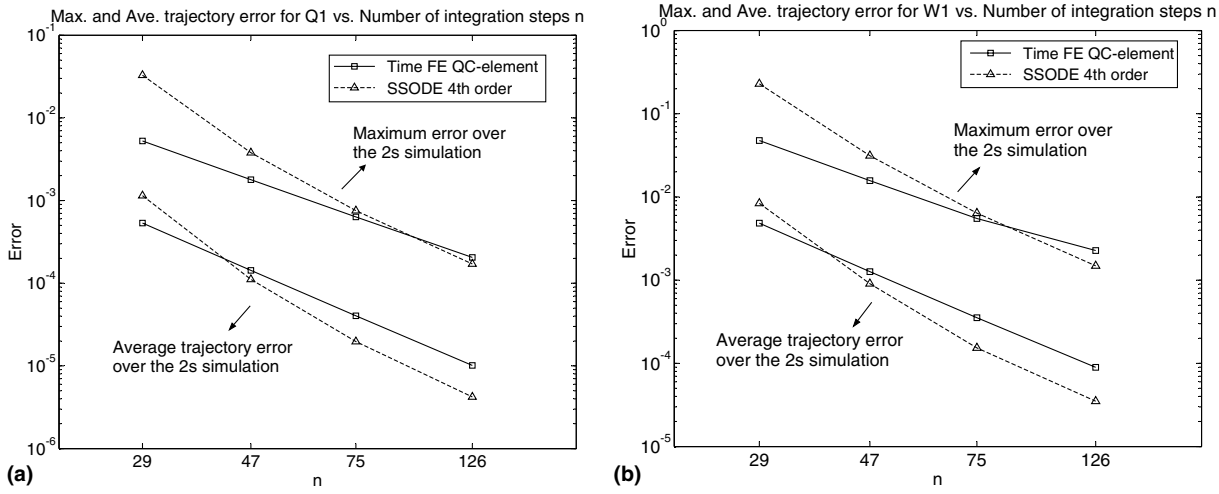

Fig. 10. Error comparison between the ST QC-Lagrange and fourth order SSODE results. (a) Error results for the angle q_1 vs. n and (b) error results for the angular velocity, w_1 vs. n.

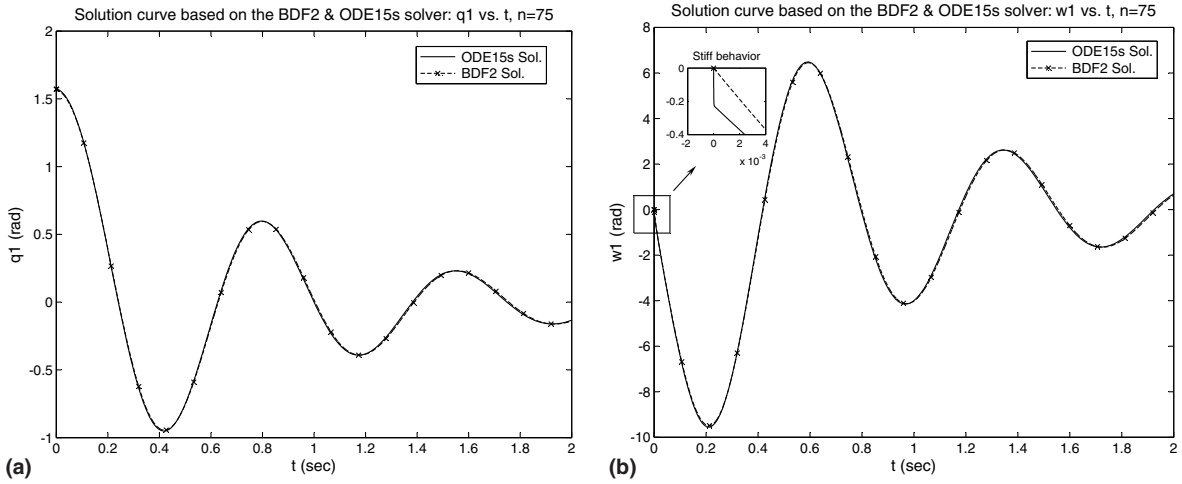

Fig. 11. Simulation results of the stiff double pendulum, $n = 75$. (a) Results for angle q_1 and (b) results for angular velocity, w_1.

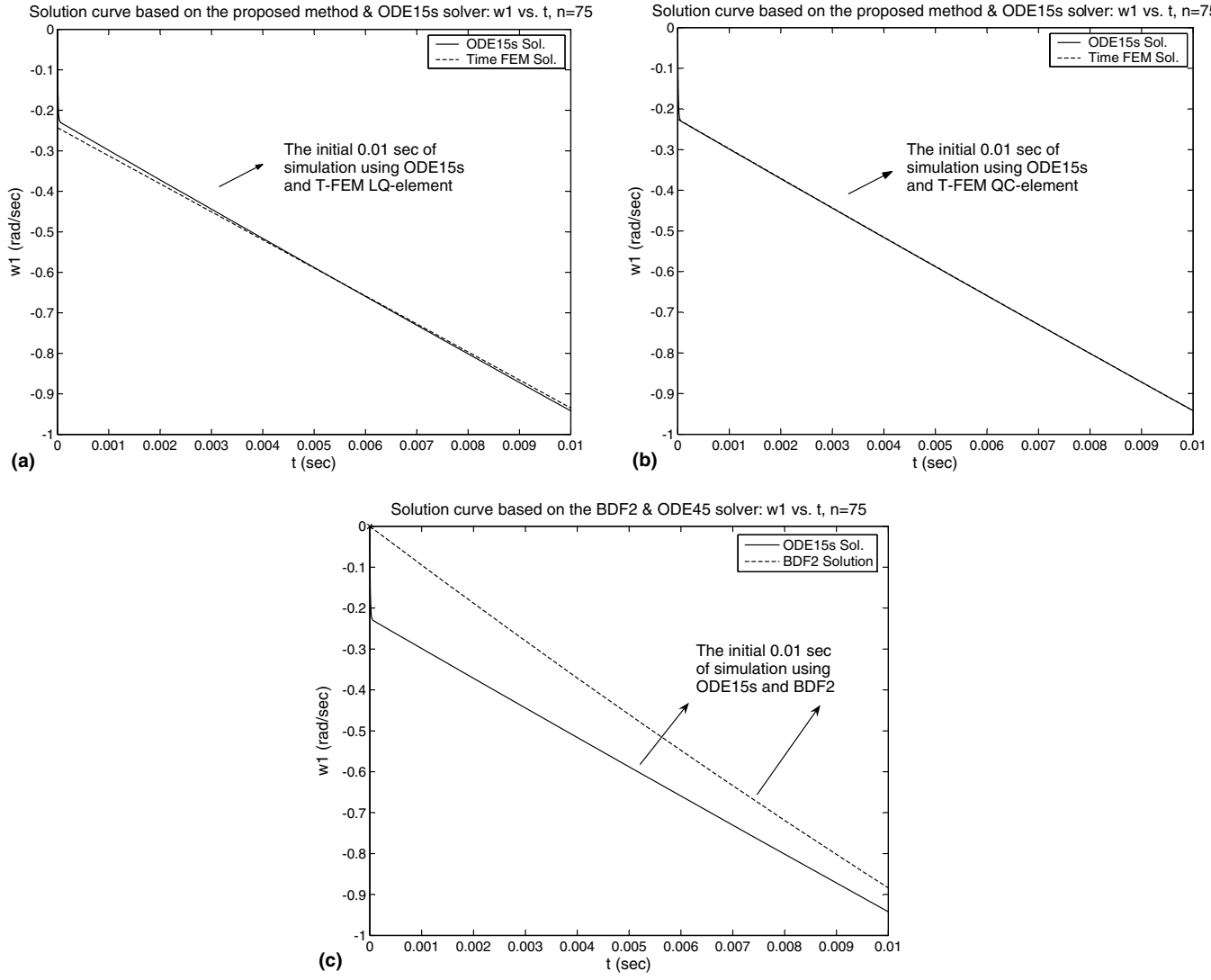

Fig. 12. Capturing the stiff behavior via ST formulation. (a) LQ interpolation, (b) QC interpolation and (c) BDF2.

Comparison of these plots with those presented in Fig. 7 shows that the results from BDF2 method are more accurate than the ones from LQ-Lagrange element and slightly less accurate than the ones with QC-Lagrange element. This is expected since the BDF2 is a second order method while the other two have super-linear and super-quadratic accuracy, respectively. Also, as indicated above the linear and angular velocity of the two bodies are approximated using Lagrange shape functions which are of one degree less than the shape functions used for position coordinates and rotation angles. This is another source of reduced accuracy in the calculations per each integration step. Being that as it may, it is observed that integration scheme based on the presented method shows better performance in capturing the region of solution around the stiff behavior. This is illustrated in Fig. 12, where plots are obtained by zooming in on the initial stiff behavior of angular velocity curve of the simulations considered so far. Note that all methods perform the simulation with the same constant step size for integration.

5. Conclusion

When performing dynamic simulation of mechanical systems, a system of often differential algebraic equations must be solved at the current time step for the system state derivatives, which should be temporally integrated. The results from current integration step are then used to update the system state and the process repeats as the simulation marches sequentially forward in time.

By applying the proposed time-finite element (a.k.a. state-time [20]) methodology on the current temporal integration step and running the simulation in a sequential manner, it is shown that the method has the potential of acting as an effi-

cient implicit integration scheme relative to more traditional state-space schemes for multibody systems with large numbers of degrees of freedom. This formulation will not be advantageous when dealing with simple mechanical systems, since it results in a large set of equations and increased numerical effort. However, for sufficiently general and complex systems where there is a large set of equations and constraints, this method should offer particular advantage relative to traditional methods as the resulting system of equations exhibit at worst quadratic nonlinearity, and minimum degree of coupling. This results in a large but sparse system tangent matrix which is trivial to construct, and inexpensive to decompose and solve. Additionally, the method becomes significantly beneficial when some or all constraint forces are explicitly needed for the purpose of design/optimization and/or modeling of contact friction.

In this investigation, only the polynomial interpolating functions are utilized in the test cases presented. However, there are other classes of approximating techniques such as compactly supported *wavelet* bases that have been shown to be a powerful numerical tool for the fast and accurate solution of partial differential equations occurring in various branches of science and engineering. Their well-known multiresolution capability and localization properties can also be potentially used within this formulation which shows great promise when dealing with dynamic simulation of complex multibody systems, particularly as it relates to capturing stiff behavior.

Acknowledgments

Support for this work has been provided by National Science Foundation (NSF) under the Award No. CMS-0219734 and is gratefully appreciated. Also, the authors appreciate Mr. John Evans, one of URP students in the research group, who has contributed for some test cases.

References

[1] M. Schaub, B. Simeon, Automatic *h*-scaling for the efficient time integration of stiff mechanical systems, Multibody Syst. Dyn. J. 8 (2002) 329–345.
[2] M. Schaub, B. Simeon, Blended Lobatto methods in multibody dynamics, ZAMM J. Appl. Math. Mech. 83 (10) (2003) 720–728.
[3] J. Meijaard, Application of Runge–Kutta–Rosenbrock methods to the analysis of flexible multibody systems, Multibody Syst. Dyn. J. 10 (2003) 263–288.
[4] S. Chen, D.A. Tortorelli, An energy-conserving and filtering method for stiff nonlinear multibody dynamics, Multibody Syst. Dyn. J. 10 (2003) 341–362.
[5] D. Negrut, E.J. Haug, H.C. German, An implicit Runge–Kutta method for integration of differential algebraic equations of multibody dynamics, Numer. Alg. 9 (2003) 121–142.
[6] D. Pogorelov, Differential-algebraic equations in multibody system modelling, Multibody Syst. Dyn. J. 19 (1998) 183–194.
[7] S. Kim, E.J. Haug, Dual rate integration methods for flexible mechanical system dynamics, Technical Report 86-12, Center for Computer-Aided Design, The University of Iowa, 1986.
[8] S.S. Kim, J.S. Freeman, Multirate integration for mutibody dynamic analysis with decomposed subsystems, in: Proceedings of the ASME Design Engineering Technical Conferences (ASME DETC99), number DETC99/VIB-8252, Las Vegas, NV, September 12–15, 1999.
[9] S. Skelboe, Multirate integration methods, theory and implementation, Proc.—IEEE Int. Symp. Circuits Syst. 3 (1984) 1089–1092.
[10] O.A. Palusinski, Simulation of dynamic systems using multirate integration techniques, Trans. Soc. Computer Simul. 2 (4) (1985) 257–273.
[11] M. Borri, C. Bottaso, Basic features of the time finite element approach for dynamics, Meccanica 27 (1992) 119–130.
[12] M. Borri, C. Bottasso, Petrov–Galerkin finite elements in time for rigid-body dynamics, J. Guidance Control Dyn. 17 (5) (1994) 1061–1067.
[13] C. Bottaso, A general framework for interpreting time finite element formulations, Appl. Numer. Math. 25 (1997) 355–368.
[14] C. Bottaso, M. Borri, Some recent development in the theory of finite elements in time, Comput. Model. Simul. Engrg. 4 (3) (1999) 201–205.
[15] T.J.R. Hughes, G.M. Hulbert, Space–time finite element methods for elastodynamics: formulations and error estimates, Comput. Methods Appl. Mech. Engrg. 66 (3) (1988) 339–363.
[16] T.J.R. Hughes, G.M. Hulbert, Space–time finite element methods for second-order hyperbolic equations, Comput. Methods Appl. Mech. Eng. 84 (3) (1990) 327–348.
[17] P. Betsch, P. Steinmann, Conservation properties of a time FE method. Part I. Time-stepping schemes for *N*-body problems, Int. J. Numer. Methods Engrg. 49 (2000) 599–638.
[18] P. Betsch, P. Steinmann, Conservation properties of a time FE method. Part II. Time-stepping schemes for non-linear elastodynamics, Int. J. Numer. Methods Engrg. 50 (2001) 1931–1955.
[19] P. Betsch, P. Steinmann, Conservation properties of a time FE method. Part III. Mechanical systems with holonomic constraints, Int. J. Numer. Methods Engrg. 53 (2002) 2271–2304.
[20] K.S. Anderson, M. Oghbaei, A State-time formulation for dynamic systems simulation using massively parallel computing resources, Nonlinear Dyn. J. 39 (3) (2005) 305–318.
[21] R.E. Roberson, R. Schwertassek, Dynamics of Multibody Systems, Springer-Verlag, Berlin, 1988.
[22] T.R. Kane, P.W. Linkins, D.A. Levinson, Spacecraft Dynamics, McGraw-Hill, 1983.
[23] T.J.R. Hughes, The Finite Element Method, Linear Static and Dynamic Finite Element Analysis, Dover Publications Inc., New York, 1987.
[24] P. Benner, E.S. Quintana-Ort, G. Quintana-Ort, Solving linear and quadratic matrix equations on distributed memory parallel computers, in: Proceedings of the 1999 IEEE International Symposium on Computer Aided Control System Design, Hawaii, USA, August 22–27, 1999, pp. 46–51. ISBN 0-7803-5449-4.
[25] P. Benner, Efficient algorithms for large-scale quadratic matrix equations, Proc. Appl. Math. Mech. 1 (1) (2002) 492–495.
[26] R.A. Wehage, E.J. Haug, Generalized coordinate partitioning for dimension reduction in analysis of constrained systems, J. Mech. Des. 104 (1982) 247–255.