MULTIBODY DYNAMICS 2003
Jorge A.C. Ambrósio(Ed)
IDMEC/IST, Lisbon,Portugal, July 1-4 2003

IMPLICIT TIME-STEPPING PROCEDURES IN CONSERVATIVE NON-LINEAR ELASTODYNAMICS: IMPROVED ACCURACY AND APPLICATION TO SYSTEMS WITH 3D ROTATIONS AND CONSTRAINED KINEMATICS

Gordan Jelenić, Edward Graham and José Muñoz

Imperial College London
Aeronautics Department
Prince Consort Road
London SW7 2BY
e-mail: g.jelenic@imperial.ac.uk, e.graham@imperial.ac.uk, jose.munoz@imperial.ac.uk

Keywords: conservative time-integration, higher-order accuracy, bar and beam finite elements, kinematic constraints

Abstract. *In this work, an approach is made towards improving the global accuracy and also the order of accuracy in implicit time-stepping procedures for stiff mechanical problems. In principle, higher-order accuracy can be achieved without compromising any of the conservation properties of an underlying (second-order) algorithm, and without introduction of additional integration stages or splitting of the original time step. Alternatively, it may be preferable to keep the order of accuracy of the underlying algorithm and focus on reducing the global error of the solution. The key to this is in the formulation of time-stepping procedure that utilise non-constant integration parameters. For 3D beams with standard joint types, a somewhat related technique may be used to address some of the complexities typical of beam elements. For more complex joint types, however, this approach may not be immediately applicable. As an example, the sliding joint is analysed in detail.*

1 INTRODUCTION

Numerical time integration of dynamical systems may be performed using either explicit or implicit time-stepping procedures. The latter are particularly suitable for problems with a wide spectrum of natural frequencies in the cases when it is reasonable to assume that only the amplitudes associated with the lower part of the frequency spectrum contribute significantly to the time response of the system. In order to improve the numerical stability properties of implicit

time-stepping procedures in the non-linear regime, it is important to design these procedures in such a way as to make the integrals of motion (total energy for Hamiltonian systems and the total translational and angular momentum for the systems with spatial symmetries) algorithmically conserved [1−11]. Additionally, the conserved quantities in mechanical systems with symmetries imply the existence of families of *relative equilibrium states*. It is becoming increasingly accepted that competitive time-integrators should be capable of 192.168.0.42preserving these states [1, 3, 11].

In Section 2 of this paper we will show how to improve the accuracy properties of some existing conserving time-stepping procedures, by introducing a set of configuration-dependent scalar integration parameters. This is done whilst maintaining all of the conservation properties of the underlying algorithm. We will build on the ideas outlined in [11] (which were presented for a central-force problem) and analyse to what extent they may be applied to (i) a single bar element and (ii) an assembly of bar elements. The use of configuration-dependent integration parameters allows for improved accuracy to be achieved within a *single-step time-advancing framework*, which is in contrast with more costly multi-stage and time-splitting techniques [12]. The additional computational overhead will thus be almost completely contained in the formation of the Jacobian matrix within the Newton-Raphson iterative solution procedure. The approach we use here is motivated by [6], although we demand not only conservation of the integrals of motion, but also preservation of the physical states of relative equilibria. The improved accuracy properties will be shown to reduce the absolute position and momentum errors when the time step is large, or improve the order of convergence towards the exact solution when the time step is sufficiently small.

In Section 3 we turn our attention to time integration in more practical problems of motion of flexible multibody systems. In these structures, we need to consider a variety of additional complexities, which are mainly related to the existence of 3D rotations as additional unknowns of the problem. Firstly, the rotational degrees of freedom in 3D space make the configuration space of the problem a non-linear manifold rather than a (linear) vector space; this more elaborate algebraic structure requires a more sophisticated process of time integration [7, 9, 10, 13]. Additionally, the mathematical model of the structure also requires the spatial discretisation of the rotational degrees of freedom; unless handled properly, this discretisation is known to be detrimental to objectivity of the algorithmic strain measures [14, 15, 16, 17]. Finally, multibody systems may contain different types of kinematic releases, translational as well as rotational, which require a specially designed procedure within a finite-element setting [18, 19, 20, 21, 22]. In this paper, we will embed the kinematic releases within a minimum-set method [2, 20, 21, 22] and address some of the difficulties related to constraint violation and conservative integration in the case of sliding joints [23].

2 IMPLICIT INTEGRATION USING NON-CONSTANT PARAMETERS

In dynamic numerical analyses of elastic continua, it is standard to perform the spatial discretisation of the continuum using a finite-element-based interpolation of the displacement and

the velocity (and sometimes also the acceleration) fields, while simultaneously employing a finite-difference based time integration of these fields. In order to do so, we normally define the time-integration process by specifying a set of *constant* integration parameters [24].

In this section, we will present a generalisation of this approach, whereby the integration parameters are allowed to be configuration-dependent, and apply the new method to problems of one-dimensional elastic continua – bar elements and truss structures.

2.1 Application to a single bar element

Let us consider a bar element of unstrained length $\bar{l}$ and constant cross-sectional area A, and define its position with respect to the origin of an inertial system $\mathbf{0}$ by specifying the position vectors of its end-points $\boldsymbol{r}_1, \boldsymbol{r}_2 \in \mathbb{R}^3$.

For an elastic, homogeneous bar, with a constant strain, the strain energy function ϕ depends only on the length $l = \|\boldsymbol{r}_2 - \boldsymbol{r}_1\|$ of the bar, and, for a constant load $\boldsymbol{F}$ acting at the two end-nodes, the potential of the applied loads U is a function of the vector of nodal positions $\boldsymbol{R}^t = \langle \boldsymbol{r}_1^t \; \boldsymbol{r}_2^t \rangle$ such that $U = \boldsymbol{R}\cdot\boldsymbol{F}$. The total potential energy and the total kinetic energy of the bar are then

$$\Phi(\boldsymbol{R}) = \phi(l) - \boldsymbol{R}\cdot\boldsymbol{F} \quad \text{and} \quad T(\dot{\boldsymbol{R}}) = \frac{1}{2}\int_0^{\bar{l}} A\rho\dot{\boldsymbol{r}}(x)\cdot\dot{\boldsymbol{r}}(x)dx, \tag{1}$$

where $\boldsymbol{r}(0) = \boldsymbol{r}_1$, $\boldsymbol{r}(\bar{l}) = \boldsymbol{r}_2$ and ρ is the density of the bar material in its undeformed state.

2.1.1 Spatially- and fully discrete equations of motion

By assuming a linear interpolation of the displacement vector along the bar via $\boldsymbol{r}(x) = \boldsymbol{N}(x)\boldsymbol{R}$ with $\boldsymbol{N}(x) = \frac{1}{l}\left[(\bar{l}-x)\boldsymbol{I} \;\; x\boldsymbol{I}\right]$, where $\boldsymbol{I}$ is the identity matrix in $\mathbb{R}^3 \times \mathbb{R}^3$, the Lagrangian function becomes

$$L(\boldsymbol{R},\dot{\boldsymbol{R}}) \equiv T(\dot{\boldsymbol{R}}) - \Phi(\boldsymbol{R}) = \frac{1}{2}\dot{\boldsymbol{R}}\cdot\boldsymbol{M}\dot{\boldsymbol{R}} - \phi(l) + \boldsymbol{R}\cdot\boldsymbol{F} \quad \text{where} \quad \boldsymbol{M} = \frac{A\rho\bar{l}}{6}\begin{bmatrix} 2\boldsymbol{I} & \boldsymbol{I} \\ \boldsymbol{I} & 2\boldsymbol{I} \end{bmatrix}. \tag{2}$$

The equation of motion for a semi-discrete dynamical system containing a single bar finite element then follows from $\frac{d}{dt}\nabla_{\dot{\boldsymbol{R}}}L - \nabla_{\boldsymbol{R}}L = \mathbf{0}$, where

$$\nabla_{\boldsymbol{R}}L = -\phi'(l)\nabla_{\boldsymbol{R}}\, l + \boldsymbol{F} = -\frac{\phi'(l)}{l}\tilde{\boldsymbol{I}}\boldsymbol{R} + \boldsymbol{F} \quad \text{with} \quad \tilde{\boldsymbol{I}} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{I} \\ -\boldsymbol{I} & \boldsymbol{I} \end{bmatrix}. \tag{3}$$

By writing $\boldsymbol{V} = \dot{\boldsymbol{R}}$ we arrive at the coupled first-order system

$$\boldsymbol{M}\dot{\boldsymbol{V}} = -\frac{\phi'(l)}{l}\tilde{\boldsymbol{I}}\boldsymbol{R} + \boldsymbol{F} \quad \text{and} \quad \dot{\boldsymbol{R}} = \boldsymbol{V}. \tag{4}$$

We now propose a family of algorithms that mimic the above differential equations by satisfying the following discrete equations of motion

$$\begin{aligned} \frac{1}{\Delta t}(\boldsymbol{M}\boldsymbol{V}_d + \gamma\boldsymbol{B}^{-1}\tilde{\boldsymbol{I}}\boldsymbol{M}\boldsymbol{V}_{1/2}) &= -\xi\boldsymbol{B}^{-1}\tilde{\boldsymbol{I}}\boldsymbol{R}_{1/2} + \boldsymbol{F} \\ \frac{1}{\Delta t}(\boldsymbol{R}_d - \gamma\boldsymbol{B}^{-1}\tilde{\boldsymbol{I}}\boldsymbol{R}_{1/2}) &= \boldsymbol{B}^{-1}\boldsymbol{V}_{1/2}, \end{aligned} \tag{5}$$

where for any given quantity $(\cdot)$ we define $(\cdot)_d \equiv (\cdot)_{n+1} - (\cdot)_n$ and $(\cdot)_{1/2} \equiv \frac{1}{2}[(\cdot)_n + (\cdot)_{n+1}]$; $\boldsymbol{R}_k$ and $\boldsymbol{V}_k$ are the discrete approximations to the displacements $\boldsymbol{R}(t_k)$ and velocities $\boldsymbol{V}(t_k)$ at time $t_k \geq 0$; and $\Delta t = t_{n+1} - t_n$ is the time-step length. The parameters γ and ξ and matrix $\boldsymbol{B}$ are real-valued functions of $\boldsymbol{R}_n$ and $\boldsymbol{R}_{n+1}$, and $\boldsymbol{B}$ is assumed to have the block structure

$$\boldsymbol{B} = \begin{bmatrix} \beta_1\boldsymbol{I} & \beta_2\boldsymbol{I} \\ \beta_2\boldsymbol{I} & \beta_1\boldsymbol{I} \end{bmatrix}. \tag{6}$$

2.1.2 Conservation criteria and local accuracy analysis

We now state without proof the conditions for algorithm (5) to conserve energy and momenta, and also to preserve the relative equilibria of the underlying physical system.

Proposition *Algorithm (5) conserves translational and angular momentum when* $\boldsymbol{F} = \boldsymbol{0}$ *for any* $\beta_1,\ \beta_2,\ \gamma$ *and* $\xi \in \mathbb{R}$ *such that* $\det(\boldsymbol{B}) \neq 0$ *and also conserves energy provided that*

$$\xi = \frac{\phi_d - \gamma(\boldsymbol{B}^{-1}\tilde{\boldsymbol{I}}\boldsymbol{R}_{1/2}\cdot\boldsymbol{F} + \frac{1}{\Delta t^2}\boldsymbol{R}_d^*\cdot\boldsymbol{B}^{-1}\tilde{\boldsymbol{I}}\boldsymbol{M}\boldsymbol{R}_d^*)}{\boldsymbol{R}_d^*\cdot\boldsymbol{B}^{-1}\tilde{\boldsymbol{I}}\boldsymbol{R}_{1/2}}, \tag{7}$$

where $\boldsymbol{R}_d^* = \boldsymbol{B}\boldsymbol{R}_d - \gamma\tilde{\boldsymbol{I}}\boldsymbol{R}_{1/2}$*. Algorithm (5) additionally recovers the exact relative equilibrium solution for a free-flying bar provided that*

$$\gamma = 0,\ \ \xi = \frac{\phi'(l_0)}{l_0},\ \ \beta_1 - \beta_2 = \frac{\frac{1}{2}w_0\Delta t}{\tan(\frac{1}{2}\theta)},\ \ \text{and}\ \ \beta_1 + \beta_2 = 1 \tag{8}$$

where l_0 *denotes the initial length of the bar,* w_0 *its initial angular velocity about one of the principal axes of inertia and* $\theta = \arccos\frac{(\boldsymbol{r}_{2,n}-\boldsymbol{r}_{1,n})\cdot(\boldsymbol{r}_{2,n+1}-\boldsymbol{r}_{1,n+1})}{l_n l_{n+1}}$ *is the incremental angle of rotation, whenever the appropriate initial conditions for such a solution exist. It also recovers the exact trajectory rotational relative equilibrium solution for a bar pinned at one of its ends provided that*

$$\gamma = 0,\ \ \xi = \frac{\phi'(l_0)}{l_0},\ \ \text{and}\ \ \beta_1 = \frac{\frac{1}{2}w_0\Delta t}{\tan(\frac{1}{2}\theta)}, \tag{9}$$

whenever the appropriate starting conditions for such a solution exist.

Thus incorporating conditions (7)-(9) into algorithm (5) results in an energy-momentum algorithm that exactly recovers the solutions along trajectories of relative equilibria, while still allowing for different choices of $\beta_1,\ \beta_2$ and γ (which then determine ξ). The algorithm of Simo and Tarnow [8], as applied to a single bar element, is recovered when $\beta_1 = 1, \beta_2 = 0$ and $\gamma = 0$.

We now investigate the local accuracy of algorithm (5). We first define the *local error vector* as

$$\boldsymbol{\epsilon} := \boldsymbol{Z}_{n+1} - \boldsymbol{Z}(t_{n+1}) \quad \text{when} \quad \boldsymbol{Z}_n = \boldsymbol{Z}(t_n) \tag{10}$$

where $\boldsymbol{Z}^t = \langle \boldsymbol{R}^t \;\; \boldsymbol{V}^t \rangle$. The local error represents the departure from the exact solution at a given time after one time-step assuming the solution at time-step n to be exact, i.e. $\boldsymbol{Z}_n = \boldsymbol{Z}(t_n)$. We next define the *residual vector* $\boldsymbol{g}^t(\boldsymbol{X}) = \langle \boldsymbol{g}^{1^t}(\boldsymbol{X}) \;\; \boldsymbol{g}^{2^t}(\boldsymbol{X}) \rangle$ where $\boldsymbol{X}^t = \langle \boldsymbol{X}_R^t \;\; \boldsymbol{X}_V^t \rangle$ with

$$\begin{aligned} \boldsymbol{g}^1(\boldsymbol{X}) &:= \frac{1}{\Delta t}\Big[\boldsymbol{B}\boldsymbol{M}(\boldsymbol{X}_V - \boldsymbol{V}_n) + \frac{\gamma}{2}\tilde{\boldsymbol{I}}\boldsymbol{M}(\boldsymbol{X}_V + \boldsymbol{V}_n)\Big] + \frac{\xi}{2}\tilde{\boldsymbol{I}}(\boldsymbol{X}_R + \boldsymbol{R}_n) - \boldsymbol{B}\boldsymbol{F}, \\ \boldsymbol{g}^2(\boldsymbol{X}) &:= \frac{1}{\Delta t}\Big[\boldsymbol{B}(\boldsymbol{X}_R - \boldsymbol{R}_n) - \frac{\gamma}{2}\tilde{\boldsymbol{I}}(\boldsymbol{X}_R + \boldsymbol{R}_n)\Big] - \frac{1}{2}(\boldsymbol{X}_V + \boldsymbol{V}_n), \end{aligned} \tag{11}$$

and make use of the fact that $\boldsymbol{g}^1$ and $\boldsymbol{g}^2$ are related to the local error vector in such a way as to make it sufficient to analyse the errors in $\boldsymbol{g}^1$ and $\boldsymbol{g}^2$ in order to analyse the local error [25]. We proceed by inserting the exact solution $\boldsymbol{X} = \boldsymbol{Z}(t_{n+1})$ into the residual expression (11) and examining the coefficients of the powers of Δt that arise. We assume that $\boldsymbol{Z}(t)$ is analytic in a neighbourhood of t_n, and that β_1, β_2, γ and ξ are also analytic functions of t within the same neighbourhood. This enables us to write

$$\beta_j[\boldsymbol{Z}(t_{n+1})] = \sum_{i=0}^{\infty} \beta_{j,i}\Delta t^i, \quad \gamma[\boldsymbol{Z}(t_{n+1})] = \sum_{i=0}^{\infty} \gamma_i \Delta t^i \quad \text{and} \quad \xi[\boldsymbol{Z}(t_{n+1})] = \sum_{i=0}^{\infty} \xi_i \Delta t^i \tag{12}$$

where the coefficients $\beta_{1,i}, \beta_{2,i}, \gamma_i$ and ξ_i are fully defined at time t_n. By further substituting (12) into (11), we can derive conditions for obtaining a given order of accuracy in terms of the $\beta_{1,i}$, $\beta_{2,i}$, γ_i and ξ_i. Table 1 lists the criteria up to and including 4^{th}-order accuracy.

It can be shown that the conservation criteria (7)-(9) do not conflict with the conditions given in Table 1. Thus the new family contains 4^{th}-order accurate energy-momentum algorithms that exactly recover the trajectories of relative equilibria.

2.1.3 Numerical example – motion in a central force field

Let us analyse the pendulum example shown in Figure 1, as used by many others [1, 3, 5, 11], with the potential function representative of the St. Venant-Kirchhoff material [3]

$$\tilde{V}(l) = \frac{1}{2}k\left(\frac{l^2 - \bar{l}^2}{2\bar{l}}\right)^2 \tag{13}$$

for constants k (stiffness) and $\bar{l}$.

The top of the pendulum is fixed at the origin $(0, 0)$ with the mass starting at position $\boldsymbol{q}_0^t = \langle 0 \;\; l_0 \rangle$ with initial length $l_0 = \bar{l} = 1$. The mass used is $m = \frac{A\rho\bar{l}}{3} = 1$, and the pendulum is fired with an initial horizontal velocity $\boldsymbol{v}_0^t = \langle 10 \;\; 0 \rangle$. The stiffness is chosen to be $k = 10^8$,

k	conditions needed
0	$\boldsymbol{B}_0 \neq \boldsymbol{0},\ \gamma_0\tilde{\boldsymbol{I}} = \boldsymbol{0}$
1	$\boldsymbol{B}_0 = \boldsymbol{I}_6,\ \xi_0 = f_n,\ \gamma_1\tilde{\boldsymbol{I}} = \boldsymbol{0}$
2	$\boldsymbol{B}_1 = \boldsymbol{0},\ \xi_1 = \frac{1}{2}\dot{f}_n,\ \gamma_2\tilde{\boldsymbol{I}} = \boldsymbol{0}$
3	$\boldsymbol{B}_2 = -\frac{1}{12}f_n\boldsymbol{M}^{-1}\tilde{\boldsymbol{I}},\ \xi_2 = \frac{1}{6}\ddot{f}_n,\ \gamma_3\tilde{\boldsymbol{I}} = \frac{1}{12}\dot{f}_n\boldsymbol{M}^{-1}\tilde{\boldsymbol{I}}$
4	$\boldsymbol{B}_3 = -\frac{1}{24}\dot{f}_n\boldsymbol{M}^{-1}\tilde{\boldsymbol{I}},\ \xi_3 = \frac{1}{24}f_n^{(3)},\ \gamma_4\tilde{\boldsymbol{I}} = \frac{1}{24}\ddot{f}_n\boldsymbol{M}^{-1}\tilde{\boldsymbol{I}}$

Table 1: Cumulative conditions for k^{th}-order accuracy, where $f_n^{(s)} \equiv \frac{\mathrm{d}^s}{\mathrm{d}t^s}\left\{\frac{\phi'[l(t)]}{l(t)}\right\}_{t=t_n}$.

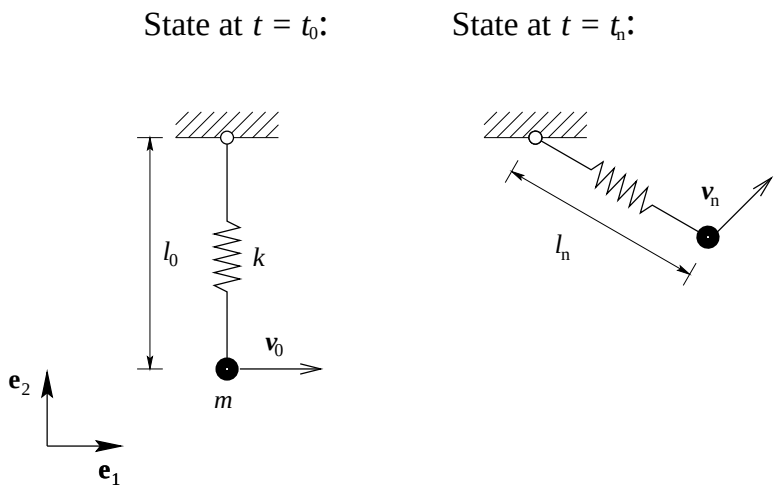


Figure 1: Pendulum of mass m and stiffness k with initial length l_0 and velocity $\boldsymbol{v}_0$.

which creates a large disparity between the two natural frequencies of motion, thus making the problem *stiff*.

Each of the algorithms under consideration will be used with a range of time-step sizes, and the relative errors in the positions $\boldsymbol{q}$ and momenta $\boldsymbol{p}$ will be taken at 0.6 seconds. The relative errors are calculated as

$$\frac{||\boldsymbol{q}_n - \boldsymbol{q}(t_n)||}{||\boldsymbol{q}(t_n)||} \quad \text{and} \quad \frac{||\boldsymbol{p}_n - \boldsymbol{p}(t_n)||}{||\boldsymbol{p}(t_n)||}$$

respectively, where $\{\boldsymbol{q}_n,\, \boldsymbol{p}_n\}$ denotes the approximate solution and $\{\boldsymbol{q}(t_n),\, \boldsymbol{p}(t_n)\}$ the ref-

erence solution – the solution obtained by running a scheme several times with decreasing time-step sizes, until the two consecutive solutions match up to 15 digits. With the aid of quadruple precision arithmetic, we are able to use an extremely tight convergence tolerance for the Newton-Raphson iteration: 10^{-26}.

The simplest type of fourth-order algorithm to test is that which has β_1 and γ defined as truncated versions of the power series in Δt given in (12), with coefficients matching the requirements for accuracy given in Table 1. Note that due to the boundary conditions used in this example, β_2 vanishes. Thus β_1 and γ are defined entirely at time-step n, which means that algorithms of this type *cannot* be time-reversible. The remaining parameter ξ is then obtained from (7) to ensure energy conservation. We call this algorithm EM4. A more sophisticated approach to designing higher-order algorithms involves finding closed-form expressions for the parameters β_1 and γ which, when expanded in a power series in Δt, match the criteria given in Table 1. These would now involve quantities at time-step $n+1$, thus requiring a more complicated linearisation for the Newton-Raphson procedure. In this way, however, one can design algorithms which are time-reversible; by way of example, we take

$$\beta_1 \;=\; \frac{\sqrt{\frac{f_{1/2}}{m}}\frac{\Delta t}{2}}{\tan\left(\sqrt{\frac{f_{1/2}}{m}}\frac{\Delta t}{2}\right)}, \quad \gamma \;=\; \frac{\Delta t^2}{12m} f_\Delta \quad \text{and} \quad \xi \text{ given by (7).} \tag{14}$$

Expanding β_1 and γ about t_n (with $f(t_{n+1})$ in place of f_{n+1}) and comparing terms with Table 1 shows that the scheme is fourth-order accurate and preservation of relative equilibria implies recovery of the *exact* solutions at all time-step sizes for steady-state problems, with β_1 satisfying (9). We call this algorithm EMTR4. Given that the trajectory of the stiff pendulum closely resembles that of a steady-state example, we would expect such an algorithm to provide solutions to the stiff problem with minimal error in the angle of rotation. For a bench-mark against which to assess the performance of the new algorithms, we will also test the algorithm of Simo and Tarnow [8], referred to here as EMM, whereby $\beta_1 = 1$ and $\gamma = 0$ with ξ obtained from (7). Finally, we propose a second-order algorithm designed to solve stiff problems at large time-steps. Such an algorithm corresponds to (5) with

$$\beta_1 \;=\; \frac{\frac{1}{2}\theta}{\tan(\frac{1}{2}\theta)}, \quad \gamma = 0 \quad \text{and} \quad \xi \;\text{ given by (7)} \tag{15}$$

where θ is the incremental angle between $\boldsymbol{q}_n$ and $\boldsymbol{q}_{n+1}$. We call this algorithm EM2β.

Figure 2 contrasts the performance of EM4, EMTR4, EM2β and EMM at time $t_n = 0.6$ seconds. We see immediately that EM4 provides a converged solution for only the two smallest time-step sizes. In contrast, the time-reversible fourth-order accurate algorithm EMTR4 is notably more robust, but still not as robust as EMM. It should be noted, however, that whenever EMTR4 converges, it outperforms EMM. When EMTR4 does not converge, the results of EMM can still be improved by using EM2β instead. This algorithm gives much more accurate results than EMM at larger time-steps, despite the order of accuracy being the same for each.

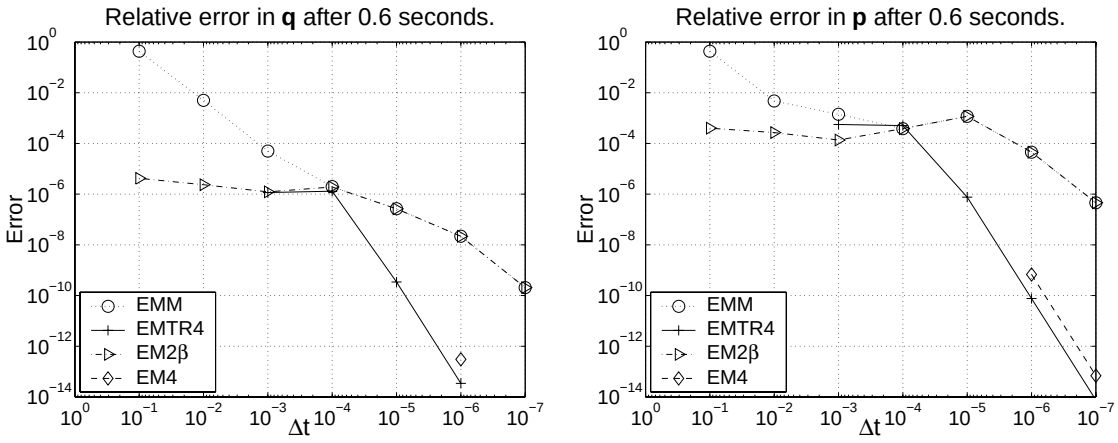


Figure 2: Relative errors in position and momentum.

This is because β_1 in (15) has been specifically designed to eliminate the error in the period of rotation, hence the errors for EM2β should always be smaller than those for EMM.

2.2 Multi-element formulation

There are hidden subtleties when extending Algorithm (5) to cater for more than one element. For such an algorithm to conserve both energy and momenta, the velocities at the nodes must be common across all elements, as one might expect. However, when pursuing accuracy higher than second order, the forces from all elements adjacent to a given node contribute to the velocity at that node [25]. This means that for higher-order accuracy with multi-element formulations, the assembly process cannot be done solely on an element by element basis, as is conventional for finite-element methods.

For a structure assembled of a number of bars joining in N nodal points, the analogue of (4) becomes

$$\boldsymbol{M}\dot{\boldsymbol{V}} = -\boldsymbol{H}\boldsymbol{R} + \boldsymbol{F}, \quad \text{and} \quad \dot{\boldsymbol{R}} = \boldsymbol{V}, \tag{16}$$

where now $\boldsymbol{R}, \boldsymbol{V}, \boldsymbol{F} \in \mathbb{R}^{3N}$, $\boldsymbol{M} \in \mathbb{R}^{3N\times 3N}$ is a global mass matrix and $\boldsymbol{H} \in \mathbb{R}^{3N\times 3N}$ is defined as

$$\boldsymbol{H} = \begin{bmatrix} \sum_{j\neq 1} h_{1j}\boldsymbol{I} & -h_{12}\boldsymbol{I} & \cdots & -h_{1N}\boldsymbol{I} \\ -h_{21}\boldsymbol{I} & \sum_{j\neq 2} h_{2j}\boldsymbol{I} & \cdots & -h_{2N}\boldsymbol{I} \\ \vdots & \vdots & \ddots & \vdots \\ -h_{N1}\boldsymbol{I} & -h_{N2}\boldsymbol{I} & \cdots & \sum_{j\neq N} h_{Nj}\boldsymbol{I} \end{bmatrix} \quad \text{with} \quad h_{ij} = \frac{\phi'_{ij}}{l_{ij}}. \tag{17}$$

where $\phi_{ij} \neq 0$ only if there exists a bar between nodes i and j. The multi-element equivalent of (5) can now be written as

$$\begin{aligned} \frac{1}{\Delta t}(\boldsymbol{M}\boldsymbol{V}_d + \boldsymbol{G}\boldsymbol{M}\boldsymbol{V}_{1/2}) &= -\alpha \boldsymbol{X}\boldsymbol{R}_{1/2} + \boldsymbol{F}, \\ \frac{1}{\Delta t}(\boldsymbol{R}_d - \boldsymbol{G}^t\boldsymbol{R}_{1/2}) &= \boldsymbol{Y}\boldsymbol{M}\boldsymbol{V}_{1/2}. \end{aligned} \tag{18}$$

Let us mention without proof that this algorithm conserves the total translational momentum for $\boldsymbol{F} = \boldsymbol{0}$ provided the sum of all the entries in any column of both $\boldsymbol{G}$ and $\boldsymbol{X}$ is equal to zero. Furthermore, the algorithm conserves the total angular momentum for $\boldsymbol{F} = \boldsymbol{0}$ provided matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ are symmetric. The conditions necessary for the preservation of relative equilibria are currently under investigation.

The conditions necessary for the algorithm to provide fourth-order accuracy are summarised in Table 2 for any $\alpha = 1 + \mathcal{O}(\Delta t^4)$. Note that such an algorithm must also be energy-conserving up to the same order of accuracy. The full energy conservation in the remaining powers of Δt can then be added in a projection-like manner described in [9]. Details are given in [25].

k	conditions needed
0	$\boldsymbol{G}_0 = \boldsymbol{0}$
1	$\boldsymbol{G}_1 = \boldsymbol{0},\ \boldsymbol{X}_0 = \boldsymbol{H}_n,\ \boldsymbol{Y}_0 = \boldsymbol{M}^{-1}$
2	$\boldsymbol{G}_2 = \boldsymbol{0},\ \boldsymbol{X}_1 = \frac{1}{2}\dot{\boldsymbol{H}}_n,\ \boldsymbol{Y}_1 = \boldsymbol{0}$
3	$\boldsymbol{G}_3 = \frac{1}{12}\dot{\boldsymbol{H}}_n\boldsymbol{M}^{-1},\ \boldsymbol{X}_2 = \frac{1}{12}\boldsymbol{H}_n\boldsymbol{M}^{-1}\boldsymbol{H}_n + \frac{1}{6}\ddot{\boldsymbol{H}}_n,\ \boldsymbol{Y}_2 = \frac{1}{12}\boldsymbol{M}^{-1}\boldsymbol{H}_n\boldsymbol{M}^{-1}$
4	$\boldsymbol{G}_4 = \frac{1}{24}\ddot{\boldsymbol{H}}_n\boldsymbol{M}^{-1},\ \boldsymbol{X}_3 = \frac{1}{24}\left(\dot{\boldsymbol{H}}_n\boldsymbol{M}^{-1}\boldsymbol{H}_n + \boldsymbol{H}_n\boldsymbol{M}^{-1}\dot{\boldsymbol{H}}_n\right) + \frac{1}{24}\boldsymbol{H}_n^{(3)}$
	$\boldsymbol{Y}_3 = \frac{1}{24}\boldsymbol{M}^{-1}\dot{\boldsymbol{H}}_n\boldsymbol{M}^{-1}$

Table 2: Cumulative conditions for k^{th}-order accuracy.

3 IMPLICIT INTEGRATION FOR 3D BEAMS WITH END RELEASES

Although the extension of the previous technique for beams has not yet been fully developed, a somewhat related idea may be employed [2, 22] in order to address some of the issues involved in designing fully conserving time-integration schemes for 3D beams [10, 26]. Briefly, to provide full conservation of the momenta and energy for a beam structure it has been found that one needs to perform the update of rotational degrees of freedom using the Cayley transform rather than the standard exponential mapping [10]. This process involves interpolating the *tangent-scaled* rotations [26], which exacerbates the already existing problem of non-invariance

of the strain measures. This problem may be solved by abandoning the interpolation of the rotational degrees of freedom altogether [17]. An alternative idea, which retains the rotational degrees of freedom in the formulation, has been presented in [2] and extended to cater for beams with end releases (including those with dependent degrees of freedom) in [22]. For a simple beam element with no external loads and N nodes we have the following dynamic equilibrium satisfied at each node I:

$$\boldsymbol{g}^I \equiv \boldsymbol{q}_d^I(\mathbf{N}_{1/2}, \mathbf{M}_{1/2}) + \boldsymbol{q}_k^I = \mathbf{0}, \tag{19}$$

where $\mathbf{N}$ and $\mathbf{M}$ are the stress and stress-couple resultants. By denoting the derivatives with respect to the arc-length parameter X of the beam and time t as $(\bullet)'$ and $(\dot{\bullet})$, the internal force vector $\boldsymbol{q}_s^I$ and the kinetic part $\boldsymbol{q}_k^I$ of $\boldsymbol{g}^I$ are then defined as follows,

$$\begin{aligned}
\boldsymbol{q}_s^I(\mathbf{N}_{1/2}, \mathbf{M}_{1/2}) &= \int_L \begin{bmatrix} I'^i\mathbf{0} & \mathbf{0} \\ -\widehat{\boldsymbol{r}}'_{1/2} I^i & I'^i\mathbf{I} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\Lambda}_{1/2}\mathbf{N}_{1/2} \\ \mathbf{S}(\boldsymbol{\alpha})^{-1}\boldsymbol{\Lambda}_n\mathbf{M}_{1/2} \end{Bmatrix} \mathrm{d}X \\
\boldsymbol{q}_k^I &= \frac{1}{\Delta t}\int_L \begin{Bmatrix} A\rho(\dot{\boldsymbol{r}}_{n+1} - \dot{\boldsymbol{r}}_n)I^i \\ (\boldsymbol{\pi}_{n+1} - \boldsymbol{\pi}_n)I^i \end{Bmatrix} \mathrm{d}X,
\end{aligned}$$

where $\boldsymbol{\Lambda}$ is the rotation matrix, $\boldsymbol{\pi} = \rho\boldsymbol{\Lambda}\mathbf{J}\mathbf{W}$ is the specific angular momentum, $\mathbf{J}$ is the tensor of mass moments of inertia, $\mathbf{W}$ is the (material) angular velocity and $\mathbf{S}(\boldsymbol{\alpha})^{-1} = \frac{1}{1+|\boldsymbol{\alpha}|^2/4}\left(\mathbf{I} + \frac{1}{2}\widehat{\boldsymbol{\alpha}}\right)$, where $\boldsymbol{\alpha}$ is the incremental tangent-scaled rotation, such that $\boldsymbol{\Lambda}_{n+1} = cay(\boldsymbol{\alpha})\boldsymbol{\Lambda}_n$ with $cay(\alpha) = \mathbf{I} + \frac{1}{1+|\boldsymbol{\alpha}|^2/4}\left(\widehat{\boldsymbol{\alpha}} + \frac{1}{2}\widehat{\boldsymbol{\alpha}}^2\right)$. The symbol $\widehat{\bullet}$ denotes a skew-symmetric matrix such that $\widehat{\boldsymbol{w}}\boldsymbol{v} = \boldsymbol{w} \times \boldsymbol{v}$ for $\boldsymbol{w}, \boldsymbol{v} \in \mathbb{R}^3$ and I^i are the standard Lagrangian interpolation polynomials. The above definition for the internal and the kinetic force vectors gives the algorithm from [10], provided the approximation of the rotational field is obtained by interpolating the *incremental tangent-scaled* rotations. While indeed fully conserving, this interpolation results in a severe loss of strain invariance [26]. Using the same definition of the nodal residual along with interpolating *incremental unscaled* rotations reduces the non-invariance problems, but such an algorithm ceases to be energy-conserving (though it is still momentum-conserving). The algorithm can be made fully invariant and momentum-conserving by applying the generalised interpolation of rotations [16]. Although such a definition of the residual does not provide an energy-conserving algorithm, this property may be recovered by adding an extra term to the weak form [2]. This extended version basically uses an alternative residual which, in the absence of external loads, is given by

$$\boldsymbol{g}_\beta^I = \boldsymbol{q}_d^I(\mathbf{N}_{1/2}, \mathbf{M}_{1/2}) + \boldsymbol{q}_k^I + \beta\boldsymbol{q}_d^I((\mathbf{N}_{n+1} - \mathbf{N}_n, \mathbf{0}).$$

In this algorithm, the free parameter β is derived in such a way to make the energy conserved. Extension of this idea to systems with a variety of different joint types, including those with dependent degrees of freedom (e.g. screw joint, rack-and-pinion joint or worm gear), is straightforward [22]. In the following sections, particular attention is paid to the formulations involving *sliding joints*, which require special treatment.

3.1 Master–slave approach

In the present work, the kinematic constraints are dealt with using the master-slave technique [20, 21, 22] which uses the minimum set of degrees of freedom (dof). The kinematics of the detached node of the joint (*slave node*) is computed from the reference kinematics (*master node/s*) plus the released displacements and/or rotations that are added to the total set of degrees of freedom of the structure.

Before writing the equations of the kinematic relationship, two main types of joints will be described: prismatic joints (non-sliding) and sliding joints. In the first case, the released displacement slides along *one of the directions of the body attached frame of the master node* (the *initial* contact node), whereas in the second case the sliding node (slave node) always slides in the direction *tangent to the centroidal axis of the beam* at the *current* contact point. Figure 3 depicts both situations, where node N (slave node) of element A slides along element B. It is worth noting that the two models use different definitions of the slave and master nodes. The prismatic joint has a permanent single master node (the node at which the initial contact is made) while the sliding joint has all the M nodes of the element as the master nodes. We also point out the fact that the prismatic joint is unrealistic (see Figure 3), but it will be detailed here for the sake of completeness.

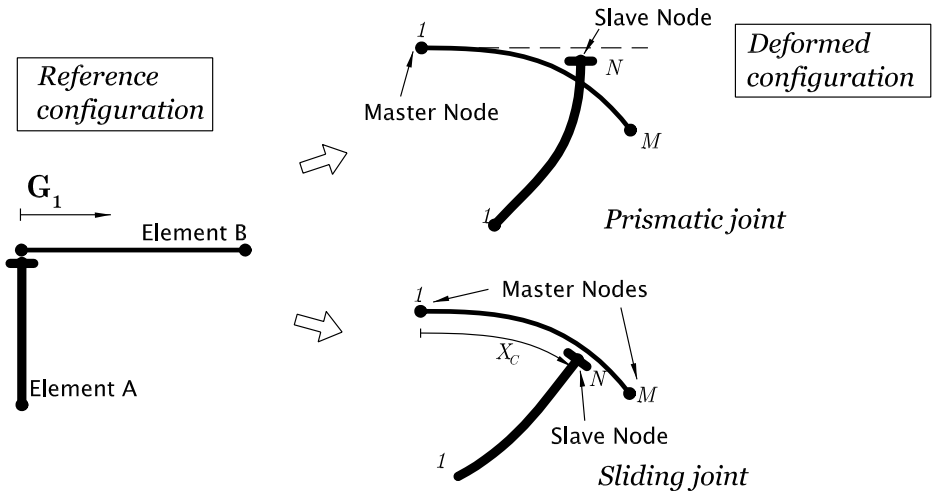


Figure 3: Master-slave approach for the prismatic and the sliding joints

Denoting by $\delta\boldsymbol{p}_s$ the kinematic variation of the slave node (node N of element A), by $\delta\boldsymbol{p}_m$ the kinematic variation of the master node (node 1 of element B in the prismatic joint, or all the nodes of element B for the sliding joint), and by $\delta\boldsymbol{p}_R$ the variations of the released displacement and rotations, a general transformation rule may be written as follows,

$$\delta\boldsymbol{p}_s = \delta\boldsymbol{p}^{A,N} = [\mathbf{R}\ \mathbf{Q}] \left\{ \begin{array}{c} \delta\boldsymbol{p}_R \\ \delta\boldsymbol{p}_m \end{array} \right\} = \mathbf{N}_\delta \delta\boldsymbol{p}_{Rm}. \tag{20}$$

This equation involves *infinitesimal* changes, i.e. $\delta \boldsymbol{p}^t =< \delta \boldsymbol{r} \ \delta \boldsymbol{\vartheta} >$ where $\delta \boldsymbol{\vartheta}$ stems from the variation of the rotation matrix $\delta \boldsymbol{\Lambda} = \widehat{\delta \boldsymbol{\vartheta}} \boldsymbol{\Lambda}$. However, an equivalent expression may be written using *incremental* changes, i.e.

$$\Delta \boldsymbol{p}_s = \mathbf{N}_\Delta \Delta \boldsymbol{p}_{Rm} \tag{21}$$

where $\Delta \boldsymbol{p}^t =< \boldsymbol{r}_{n+1} - \boldsymbol{r}_n \ \boldsymbol{\alpha} >$. Some particular forms of matrix $\mathbf{N}_\Delta$ for non-sliding joints have been derived in references [21, 22] that keep the conserving properties of the algorithm into which they are embedded.

Equations (20) and (21) allow us to rewrite the discretised virtual work of the form $\delta \mathcal{H}^I = \delta \boldsymbol{p}_s^I \cdot \boldsymbol{g}_\delta^I$ into

$$\delta \mathcal{H}^I = \delta \boldsymbol{p}_{Rm}^I \cdot \mathbf{N}_\delta^{I\,t} \boldsymbol{g}_\delta^I \tag{22}$$

or in the case of conserving algorithms, to transform the energy increment of the form $\Delta \mathcal{H}^I = \Delta \boldsymbol{p}_s^I \cdot \boldsymbol{g}_\Delta^I$ into

$$\Delta \mathcal{H}^I = \Delta \boldsymbol{p}_{Rm}^I \cdot \mathbf{N}_\Delta^{I\,t} \boldsymbol{g}_\Delta^I. \tag{23}$$

The matrices $\mathbf{N}_\delta^I$ and $\mathbf{N}_\Delta^I$ in (22) and (23) are the transformation matrices of element I. The arbitrariness of the variations $\delta \boldsymbol{p}_{Rm}$ or $\Delta \boldsymbol{p}_{Rm}$ lead to the following system of N_T equations, after assembling the elemental residual vectors $\boldsymbol{g}_\delta^I$ or $\boldsymbol{g}_\Delta^I$ for all N_T nodes of the model:

$$\sum_I \mathbf{N}_\delta^{I,i\,t} \boldsymbol{g}_\delta^{I,i} = \mathbf{0} \quad \text{for nodes} \quad i = 1, \ldots, N_T,$$

or in incremental form

$$\sum_I \mathbf{N}_\Delta^{I,i\,t} \boldsymbol{g}_\Delta^{I,i} = \mathbf{0} \quad \text{for nodes} \quad i = 1, \ldots, N_T,$$

where $\mathbf{N}_\delta^{I,i}$ or $\mathbf{N}_\Delta^{I,i}$ are the part of the transformation matrix of element I that is multiplying the dof of node i.

We emphasise that since the master node of the prismatic joint belongs to element A, the master-slave formulation of this joint can be modelled as an elemental relationship within element A and thus elemental condensation may be applied [27]. In fact, the rotation of the master node is equivalent to the orientation of the cross-section of element A at node N. For the sliding joint, however, the *master* rotation is the rotation of the cross-section of element B at the contact point, computed from the rotations of the master nodes. The master-slave relationship of the sliding joint requires both elements, A and B, and for this reason a new coupling element will be defined in Section 3.1.2.

3.1.1 Explicit form of transformation matrix $\mathbf{N}_\delta$

The matrices $\mathbf{R}$ and $\mathbf{Q}$ arising from the infinitesimal variation of the kinematic relationship between master and slave nodes are given in Table 3 for both types of joints. The part of the table concerning the prismatic joints has been already reported in the literature [27] and thus no details will be given. With regard to the sliding joint, the vectors $\boldsymbol{r}_C^B = \boldsymbol{r}^B(X_C)$ and the matrix $\boldsymbol{\Lambda}_C^B = \boldsymbol{\Lambda}^B(X_C)$ are the position vector and the rotation matrix of element B evaluated at point X_C. The constant matrix $\boldsymbol{\Lambda}_{rel}$ denotes the initial relative rotation between the two elements, i.e. $\boldsymbol{\Lambda}_0^{A,N} = \boldsymbol{\Lambda}_0^{B,1}\boldsymbol{\Lambda}_{rel}$. In computing the infinitesimal variation of the kinematics of the sliding joint, the variation of the contact point coordinate X_C must also be taken into account, i.e.

$$\begin{aligned}
\delta\boldsymbol{\Lambda}^{A,N} &= \widehat{\delta\boldsymbol{\vartheta}}^{A,N}\boldsymbol{\Lambda}^{A,N} = \boldsymbol{\Lambda}_C'^B\boldsymbol{\Lambda}_R\boldsymbol{\Lambda}_{rel}\delta X + \widehat{\delta\boldsymbol{\vartheta}}_C^B\boldsymbol{\Lambda}^{A,N} + \widehat{\boldsymbol{\Lambda}_C^B\delta\boldsymbol{\vartheta}}_R\boldsymbol{\Lambda}^{A,N} \\
&= \left[(\mathbf{G}_1\cdot\boldsymbol{r}_R)\widehat{\boldsymbol{k}_C^B} + \widehat{\delta\boldsymbol{\vartheta}_C^B} + \widehat{\boldsymbol{\Lambda}_C^B\delta\boldsymbol{\vartheta}}_R\right]\boldsymbol{\Lambda}^{A,N} \\
\delta\boldsymbol{r}^{A,N} &= \boldsymbol{r}_C'^B\delta X + \delta\boldsymbol{r}_C^B = (\mathbf{G}_1\cdot\boldsymbol{r}_R)\boldsymbol{r}_C'^B + \delta\boldsymbol{r}_C^B,
\end{aligned}$$

where $\boldsymbol{k}_C^B = \boldsymbol{k}^B(X_C)$ is the spatial curvature of the beam at the current contact point and $\mathbf{G}_1 =< 1\ 0\ 0 >^t$ is the unit vector indicating the direction along which the released displacement takes place. The relations $\delta\boldsymbol{r}_C^B = I_C^{B,j}\delta\boldsymbol{r}^{B,j}$ and $\delta\boldsymbol{\vartheta}_C^B = I_C^{B,j}\delta\boldsymbol{\vartheta}^{B,j}$ have been used in order to derive matrix $\mathbf{Q}$, where $I_C^{B,j}$ are the interpolation functions for each node j of element B evaluated at point X_C.

	Prismatic joints	Sliding joints
Displacements	$\boldsymbol{r}^{A,N} = \boldsymbol{r}_m + \boldsymbol{\Lambda}_m\boldsymbol{r}_R$	$\boldsymbol{r}^{A,N} = \boldsymbol{r}_C^B$
Rotations	$\boldsymbol{\Lambda}^{A,N} = \boldsymbol{\Lambda}_m\boldsymbol{\Lambda}_R$	$\boldsymbol{\Lambda}^{A,N} = \boldsymbol{\Lambda}_C^B\boldsymbol{\Lambda}_R\boldsymbol{\Lambda}_{rel}$
$\mathbf{R}$	$\begin{bmatrix} \boldsymbol{\Lambda}_m & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_m \end{bmatrix}$	$\begin{bmatrix} \boldsymbol{r}_C'^B\otimes\mathbf{G}_1 & \mathbf{0} \\ \boldsymbol{k}_C^B\otimes\mathbf{G}_1 & \mathbf{0} \end{bmatrix}$
$\mathbf{Q}$	$\begin{bmatrix} \mathbf{I} & -\widehat{\boldsymbol{\Lambda}_m\boldsymbol{r}}_R \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$	$\begin{bmatrix} I_C^{B,1}\mathbf{I} & \mathbf{0} & \dots & I_C^{B,M}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & I_C^{B,1}\mathbf{I} & \dots & \mathbf{0} & I_C^{B,M}\mathbf{I} \end{bmatrix}$

Table 3: Master–slave infinitesimal relationships for displacements and rotations

3.1.2 Coupling element definition of the sliding joint

As has been pointed out, the master–slave relationship for the sliding joint includes the kinematics of both elements, A and B. In order to permit the transition of the contact point through a slide-line formed by a set of elements, a generic coupling element will be defined by the slave element A and the element in which the contact point is currently situated. Although the initial common node to both elements is no longer active once the contact node has jumped to another element, it will be also be included in the coupling element as Figure 4 shows.

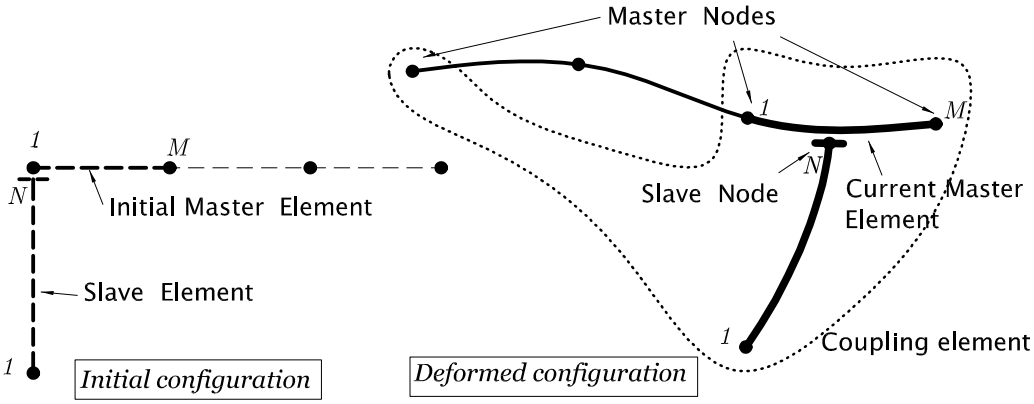


Figure 4: Coupling element definition

Using the matrices $\mathbf{R}$ and $\mathbf{Q}$ defined in Table 3 for the sliding joint, and bearing in mind that the sliding node is the N node of element A, the transformation matrix $\mathbf{N}$ for the coupling element is given by

$$\mathbf{N}_{cp} = \left[\begin{array}{c|ccc|c} \mathbf{0} & \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0}_{6\times 6M} \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} & \mathbf{0} & \mathbf{0}_{6\times 6M} \\ \mathbf{R} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{Q}_{6\times 6M} \end{array}\right]_{6N\times 6(1+N+M)} \tag{24}$$

where the matrices $\mathbf{I}$ and $\mathbf{0}$ are the unit and zero 6×6 matrices. Further details of the linearisation and the joints with dependent degrees of freedom will be given in [23].

3.2 Momentum-conserving algorithm for flexible mechanisms in the presence of sliding joints

When extending the momentum-conserving algorithm (19) to systems with sliding joints we need to preserve the conserving properties for any change of the contact position within one element and also when element jumping occurs. An algorithm that satisfies these conditions will be briefly described. The proof of the conservation of the angular momentum and details about the linearisation can be found in [23].

In order to establish the conditions for the conservation of the angular momentum, the kinematic relations of the sliding joint in Table 3 must be rewritten in an incremental form. Furthermore, the condition $\boldsymbol{r}^{A,N} = \boldsymbol{r}^{B}_{C}$ will be relaxed, so $\boldsymbol{r}^{B}_{C}$ will be seen as an approximation of the position of the sliding node $\boldsymbol{r}^{A,N}$. Considering that at times t_{n+1} and t_n the contact point coordinates are X_{C2} and X_{C1}, on elements B and D respectively (see Figure 5) , the incremental displacement of the sliding node may be obtained as follows,

$$\boldsymbol{u}^{A,N} = \boldsymbol{r}^{A,N}_{n+1} - \boldsymbol{r}^{A,N}_{n} \approx \boldsymbol{r}^{B}_{C,n+1} - \boldsymbol{r}^{B}_{C,n} = I^{D,j}_{C2}\boldsymbol{r}^{D,j} - I^{B,j}_{C1}\boldsymbol{r}^{B,j}, \tag{25}$$

where $I_{C2}^{D,j}$ and $I_{C1}^{B,j}$ stand for the interpolation functions of elements D and B evaluated at contact points X_{C2} and X_{C1}. It is shown in [23] that the expression obtained on the right of (25) may be approximated in two different ways. The first one will be termed *Appx1* and assumes that the contact point remains within one element. The second will be called *Appx2* and involves element jumping (see Table 4). Without going into the details of the algebraic manipulations, the resulting relations between the incremental degrees of freedom give rise to two different matrices $\mathbf{N}_{\Delta} = [\mathbf{R}\ \mathbf{Q}]$, which are denoted by $\mathbf{N}_{\Delta 1}$ and $\mathbf{N}_{\Delta 2}$. Table 5 gives both matrices for the two approximations, where $I_{C\frac{1}{2}}^{B,j} = \frac{1}{2}(I_{C2}^{B,j} + I_{C1}^{B,j})$, $I_{C\frac{1}{2}}^{\prime B,j} = \frac{1}{2}(I_{C2}^{\prime B,j} + I_{C1}^{\prime B,j})$, $I_{C\gamma}^{D,j} = I_{C2}^{D,j} - \gamma \Delta X I_{C2}^{\prime D,j}$, and γ is a constant parameter. The incremental rotation $\boldsymbol{\alpha}^{BD}$ is the tangent-scaled rotation that transforms $\boldsymbol{\Lambda}_{C1,n}^{B}$ (rotation of element B at time t_n at the contact point X_{C1}) into $\boldsymbol{\Lambda}_{C2,n+1}^{D}$ (rotation of element D at time t_{n+1} at the contact point X_{C2}), i.e. $\boldsymbol{\Lambda}_{C2,n+1}^{D} = cay(\boldsymbol{\alpha}^{BD})\boldsymbol{\Lambda}_{C1,n}^{B}$.

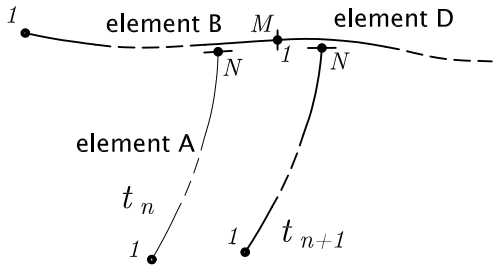


Figure 5: Simplified mesh for the case of element jumping

In fact, the two transformation matrices give rise to different conditions for the conservation of angular momentum when used in conjunction with the residual given in equation (19). The matrix $\mathbf{N}_{\Delta 1}$ leads to the condition

$$\boldsymbol{r}_{1/2}^{A,N} = I_{C\frac{1}{2}}^{B,j} \boldsymbol{r}_{1/2}^{B,j}, \tag{26}$$

whereas matrix $\mathbf{N}_{\Delta 2}$ implies the kinematic condition,

$$\boldsymbol{r}_{1/2}^{A,N} = I_{C\gamma}^{D,j} \boldsymbol{r}_{1/2}^{D,j}. \tag{27}$$

In (26), the sliding condition is satisfied in the middle of the time-step. Because of $I_{C\gamma}^{D,j}$, condition (27) is only an approximation to the sliding condition at time $t_{1/2}$; the error in this approximation is minimal for $\gamma = \frac{1}{2}$. Consequently, the time-integration strategy suggested is to use matrix $\mathbf{N}_{\Delta 1}$ whenever the contact point remains within an element at times t_n and t_{n+1}, and to apply $\mathbf{N}_{\Delta 2}$ when the contact point jumps from one element to another. This strategy is used in the the following numerical example.

It is worth noting that approximations *Appx1* and *Appx2* imply $\Delta \boldsymbol{p}_s \approx \mathbf{N}_{\Delta} \Delta \boldsymbol{p}_{Rm}$ which in turn transforms the construction of the conserving algorithms as follows

	Appx1: $\mathbf{N}_{\Delta 1}$	*Appx2*: $\mathbf{N}_{\Delta 2}$
$\mathbf{R}$	$\begin{bmatrix} I'^{B,j}_{C\frac{1}{2}} \boldsymbol{r}^{B,j}_{1/2} \otimes \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2}\left(\boldsymbol{r}'^{B}_{C1,n} + \boldsymbol{r}'^{D}_{C2,n}\right) \otimes \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix}$
$\mathbf{Q}$	$\begin{bmatrix} I^{B,1}_{C\frac{1}{2}}\mathbf{I} & \mathbf{0} & \dots & I^{B,M}_{C\gamma}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & I^{B,1}_{C\frac{1}{2}}\mathbf{I} & \dots & \mathbf{0} & I^{D,M}_{C\frac{1}{2}}\mathbf{I} \end{bmatrix}$	$\begin{bmatrix} I^{D,1}_{C\gamma}\mathbf{I} & \mathbf{0} & \dots & I^{D,M}_{C\gamma}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & I^{D,1}_{C\gamma}\mathbf{I} & \dots & \mathbf{0} & I^{D,M}_{C\gamma}\mathbf{I} \end{bmatrix}$
with $\mathbf{B} = \frac{1}{1-\frac{1}{4}\boldsymbol{\alpha}^B \cdot \boldsymbol{\Lambda}^B_{C1,n}\boldsymbol{\alpha}_R} \boldsymbol{\Lambda}^B_{C1,n}\left(\mathbf{I} + \frac{1}{2}\widehat{\boldsymbol{\alpha}}^{BD}\right)$		

Table 4: Matrices $\mathbf{R}$ and $\mathbf{Q}$ using approximations *Appx1* and *Appx2*

$$\Delta\mathcal{H} = \Delta\boldsymbol{p}_s \cdot \boldsymbol{g}_\Delta \approx \Delta\boldsymbol{p}_{Rm} \cdot \mathbf{N}^t_\Delta \boldsymbol{g}_\Delta = 0.$$

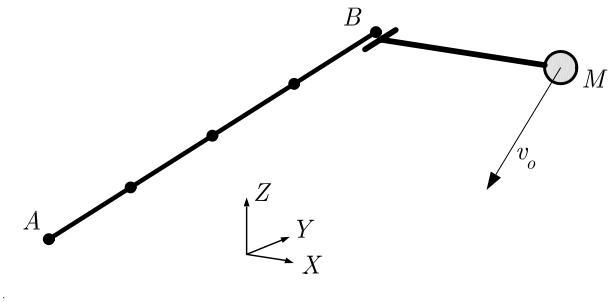
In this sense, the more accurate the approximations are, the smaller the energy increment obtained. Some attempts in making the algorithm energy conserving by avoiding the approximations or correcting the energy increment have not yet given satisfactory answers.

3.3 Numerical example

The following example involves two flexible beams connected through a sliding joint with no released rotations. The initial configuration of beams AB and BM is depicted in Figure 6, where the geometrical and material properties (the same for both beams) are also shown. Beams AB and BM are modelled using four and one quadratic elements respectively. A mass of 1 kg is attached to beam BM at point M where an initial velocity $\boldsymbol{v}_0$ is applied. The analysis terminates when the sliding node reaches point A. Two time-integration schemes are used, the Newmark trapezoidal rule [24] and the momentum conserving algorithm described in Section 3.2. The first scheme uses an initial time-step of $\Delta t = 0.0005$. During the course of the analysis this time-step had to be reduced twice in order to provide a converged solution. Eventually, however, this scheme failed to converge. Figure 7 shows that this occurred when the angular momentum blew up. The second scheme uses a constant time-step of $\Delta t = 0.0025$. The released displacement of the sliding node is shown in Figure 7. It can be seen that the results produced by the two schemes are very similar before the occurrence of the momenutm blow-up in the Newmark algorithm, in spite of the fact that in the momentum conserving algorithm the contact condition was somewhat relaxed. Figure 7 also confirms the momentum-conserving properties of the second scheme.

4 CONCLUSIONS

In the first part of this paper, we have presented a method of formulating higher-order accurate and conservative single-step algorithms for the problems of one-dimensional elastic continua. For a single bar element it has been shown that the method is capable of increasing



$$A = \left\{ \begin{array}{c} 0.0 \\ 0.0 \\ 0.0 \end{array} \right\}; B = \left\{ \begin{array}{c} 0.0 \\ 3.0 \\ 1.0 \end{array} \right\}$$
$$M = \left\{ \begin{array}{c} 1.0 \\ 3.0 \\ 1.0 \end{array} \right\}; \boldsymbol{v}_0 = \left\{ \begin{array}{c} 0.0 \\ -10.0 \\ -10.0 \end{array} \right\}$$
$$A = 1.0 \times 10^{-4} \; ; \; I_{yy} = I_{zz} = 2.0 \times 10^{-5}$$
$$E = 1.0 \times 10^{6} \; ; \; \nu = 0.3 \; ; \; \rho = 800$$

Figure 6: Free sliding mass example

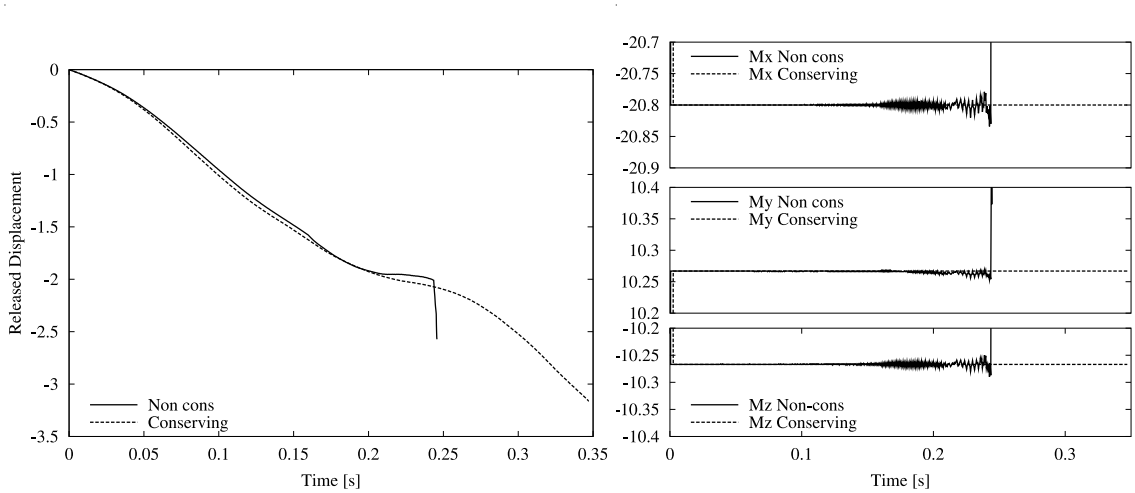


Figure 7: Released displacements and components of the angular momentum

the order of accuracy for sufficiently small time steps or reducing the global error for larger time steps. Extension of the method to cater for multi-bar truss structures brings additional complexities, which are largely related to assembly procedure and higher-order accuracy with energy conservation. Some of these issues are currently under investigation, but we believe that there exists a scope for applying the present methodology to more complicated problems of non-linear elastodynamics.

In the second part, we have briefly described a related idea, in which some of the well-known complexities of invariant and conserving time-integration of beam structures are addressed by introducing a configuration-dependent parameter. In this way, a fully conserving and invariant algorithm may be provided for most of the standard joint types including revolute, spherical, universal, prismatic, cylindrical as well as joints with dependent degrees of freedom (e.g. screw joints and rack-and-pinion joints). The problem becomes much more involved in the presence of sliding joints, in particular when the sliding can take place along a series of elements. These joints can be successfully implemented within the adopted master–slave technique for statics and non-conserving dynamics, but in order to maintain the conservation of some of the constants of motion it becomes necessary to relax some of the constraint conditions. A momentum

conserving algorithm for sliding joints has been presented and compared to the standard Newmark trapezoidal rule.

5 ACKNOWLEDGEMENTS

This work has been financed by the Engineering and Physical Science Research Council (EPSRC) with contract/grant numbers GR/R 04171/01 and AF/100089. This support is gratefully acknowledged.

REFERENCES

[1] F Armero and I Romero. On the formulation of high-frequency dissipative time-stepping algorithms for nonlinear dynamics. Part I: Low order methods for two model problems and nonlinear elastodynamics. *Comp. Meth. Appl. Mech. Engng.*, 190:2603–2649, 2001.

[2] MA Crisfield and G Jelenić. Energy/momentum conserving time integration procedures with finite elements and large rotations. In JAC Ambrósio and M Kleiber, editors, *Computational Aspects of Nonlinear Structural Systems with Large Rigid Body Motion*, pages 181–200, Pultusk, Poland, 2-7 July 2000. IOS Press.

[3] O Gonzalez and JC Simo. On the stability of symplectic and energy-momentum algorithms for non-linear hamiltonian systems with symmetry. *Comp. Meth. Appl. Mech. Engng.*, 134:197–222, 1996.

[4] D Greenspan. Completely conservative, covariant numerical methodology. *Computers and Mathematics with Applications*, 29(4):37–43, 1995.

[5] D Kuhl and MA Crisfield. Energy-conserving and decaying algorithms in non-linear structural dynamics. *Int. J. Num. Meth. Engng.*, 45:569–599, 1999.

[6] RA LaBudde and D Greenspan. Energy and momentum conserving methods of arbitrary order for the numerical integration of equations of motion I. Motion of a single particle. *Numer. Math.*, 25:323–346, 1976.

[7] D Lewis and JC Simo. Conserving algorithms for dynamics of Hamiltonian systems on Lie groups. *J. Nonlin. Sc.*, 4:253–299, 1994.

[8] JC Simo and N Tarnow. The discrete energy-momentum method. Conserving algorithms for nonlinear elastodynamics. *J. Appl. Math. Phys. (ZAMP)*, 43:757–792, 1992.

[9] JC Simo, N Tarnow, and KK Wong. Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics. *Comp. Meth. Appl. Mech. Engng.*, 100:63–116, 1992.

[10] JC Simo, N Tarnow, and M Doblare. Non-linear dynamics of three-dimensional rods: exact energy and momentum conserving algorithms. *Int. J. Num. Meth. Engng.*, 38:1431–1473, 1995.

[11] E Graham and G Jelenić. A general framework for conservative single-step time-integration schemes with higher-order accuracy for a central-force system. *CMAME*, 2003. Accepted for publication.

[12] H Yoshida. Construction of higher order symplectic integrators. *Physics Letters A*, 150(5):262–268, 1990.

[13] A Cardona and MA Géradin. A beam finite element non-linear theory with finite rotations. *Int. J. Num. Meth. Engng.*, 26:2403–2438, 1988.

[14] P Betsch and P Steinmann. Frame-indifferent beam finite elements based upon the geometrically exact beam theory. *Int. J. Num. Meth. Engng.*, 54(12):1775–1788, 2002.

[15] A Ibrahimbegović and RL Taylor. On the role of frame-invariance in structural mechanics models at finite rotations. *Comp. Meth. Appl. Mech. Engng.*, 191:5159–5176, 2002.

[16] G Jelenić and MA Crisfield. Geometrically exact 3D beam theory: implementation of a strain-invariant finite element for statics and dynamics. *Comp. Meth. Appl. Mech. Engng.*, 171:141–171, 1999.

[17] I Romero and F Armero. An objective finite element approximation of the kinematics of geometrically exact rods and its use in the formulation of an energy–momentum conserving scheme in dynamics. *Int. J. Num. Meth. Engng.*, 54:1683–1716, 2002.

[18] O Bauchau and CL Bottasso. Contact Conditions for Cylindrical, Prismatic, and Screw Joints in Flexible Multibody Systems. *Multibody System Dyn.*, 5:251–278, 2001.

[19] A Cardona and MA Géradin. *Flexible Multibody Dynamics. A Finite Element Approach.* John Wiley & Sons, 2001.

[20] A Ibrahimbegović and S Mamouri. On rigid components and joint constraints in nonlinear dynamics of flexible multibody systems employing 3D geometrically exact beam model. *Comp. Meth. Appl. Mech. Engng.*, 188:805–831, 2000.

[21] G Jelenić and MA Crisfield. Dynamic analysis of 3D beams with joints in presence of large rotations. *Comp. Meth. Appl. Mech. Engng.*, 32-33:4195–4230, 2001.

[22] J Muñoz, G Jelenić, and M Crisfield. Master-slave approach for the modelling of joints with dependent degrees of freedom in flexible mechanisms. *Comm. Num. Meth. Engng.*, 2003. In print.

[23] J Muñoz and G Jelenić. Conserving time-integration schemes for flexible mechanisms with sliding joints. In preparation.

[24] NM Newmark. A method of computation for structural dynamics. *J. Engin. Mech. Div.*, 85(EM3):67–94, 1959.

[25] E Graham. PhD thesis, University of London, Imperial College, Department of Aeronautics, 2003. In preparation.

[26] G Jelenić and MA Crisfield. Problems associated with the use of Cayley transform and tangent scaling for conserving energy and momenta in the Reissner-Simo beam theory. *Comm. Num. Meth. Engng.*, 18, 711-720, 2002.

[27] G Jelenić and MA Crisfield. Non-linear master-slave relationships for joints in 3D beams with large rotations. *Comp. Meth. Appl. Mech. Engng.*, 135:211–228, 1996.